ЕЛЕКТРОНИКА
/
ELECTRONICS

(ЕЛ/ELI)

Monitoring system for AC current up to 20A

Milan Savić, Dejan Stevanović and Miona Andrejević Stošović, *Member, IEEE*

***Abstract*—This paper presents a system for AC current monitoring in home appliances up to 20A. It is implemented on a custom made PCB. System also measures voltage, line frequency, power factor, active power and total imported active energy. Measurement results can be obtained by a remote computer or some other device via serial RS-485 interface. Consumer is enabled to have bigger control over real-time current consumption by installing several monitoring devices and connecting them into a network.**

***Index Terms*—Energy efficiency, Modbus RTU, monitoring system.**

I. Introduction

Energy efficiency has a fundamental role to play in the transition towards a more competitive, secure and sustainable energy system. Although energy powers our societies and economies, future growth must be driven with less energy and lower costs. According to the Energy Efficiency Communication of July 2014 [1], the EU is expected to achieve energy savings of 18%-19% by 2020. Reduced power consumption leads to reduced emissions and, consequently, reduced carbon footprint. This is a straightforward benefit. However, this could happen only if EU countries implement all of the existing legislation on energy efficiency. Unfortunately, efficiency of electrical distribution is currently not much managed or planned by utilities. The unfavourable result is that most utilities waste considerable amounts of electricity. For example, the annual value of transmission and distribution losses runs up to 6% of total generated energy [2]. These losses mainly occur in the low and medium voltage lines, and also in primary and secondary substations. One way for reducing losses and increasing efficiency on low level power distribution system is improving the system for registration of electric energy consumption [3-4]. Another way is to implement home energy monitoring system. A lot of similar systems have been already developed [5]. Some of realized systems are described in [6-8].

Economic return is one of the major reasons why households should consider and adopt smart energy management products. A home energy monitoring system allows consumers to have significant role in energy management activities. It can be implemented by using smart sockets. Probably, the simplest and most straightforward way to monitor and control energy usage is by replacing traditional sockets and plugs with the smart ones.

Milan Savić is with the College of Applied Technical Sciences Niš, Aleksandra Medvedeva 20, 18000 Niš, Republic of Serbia (e-mail: milan.savic@ vtsnis.edu.rs).

Dejan Stevanović and Miona Andrejević Stošović are with University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, Niš, Serbia,{dejan.stevanovic, miona.andrejevic}@elfak.ni.ac.rs.

Namely, placing multiple measurement devices in the household enables that consumer has nearly instant information about current consumption of each appliance. Some smart sockets contain relay, so that turning load on and off is supported. In this paper we proposed a system that can be implemented at electric panel. Moreover, our system can be used as smart fuses.

This paper is organized as follows. In the next section, the basic definitions that correlate power parameters with measured current and voltage data will be given. The third chapter will be dedicated to description of realised system, while in the fourth chapter measurement results will be given. The conclusion is in the fifth section.

II. Definitions of Electric Power Quantity

The core of our system is a MCP39F521 [9] which calculates all power quantities that are of interest for utility to control consumption. Usually, these values are defined by appropriate standards. All these circuits relay on digital signal processing of voltage and current samples. The instantaneous value of voltage and current are attenuated through voltage divider, while for current can be used current transformers, Rogowski coil sensors or shunt resistors. The first set of signal conditioning that occurs inside MCP39F521 is shown in Fig.1.

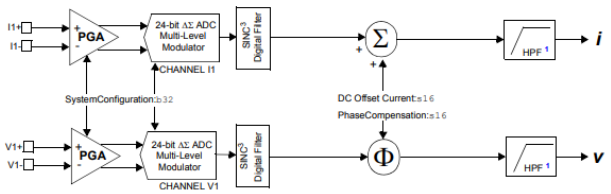


Fig. 1. Channel I1 and V1 signal flow

The obtained signal at output of attenuator is sent to ADC where it is sampled at discrete time points (at least two per period, according to the Nyquist-Shannon theorem) and digitalized. DSP processes digital voltage and current samples and calculates all necessary power quantities. Instantaneous value of signal (current or voltage) in time domain can be expressed as:

$$x(t) = \sqrt{2} X_{RMS} \cdot \cos(2\pi f t + \varphi) \qquad (1)$$

After the discretization in equidistant time intervals, it is transformed to:

$$x(nT) = \sqrt{2} X_{RMS} \cdot \cos(2\pi \frac{f}{f_{sempl}} n + \varphi) \quad , \quad (2)$$

where f and f_{sempl}, are frequency of the signal and the sampling frequency, respectively. The RMS value is calculated using the following equation:

$$X_{RMS} = \sqrt{\frac{\sum_{n=1}^{N} x(nT)^2}{N}} \quad . \quad (3)$$

The signal flow of calculation of RMS current and voltage values is presented in Fig. 2.

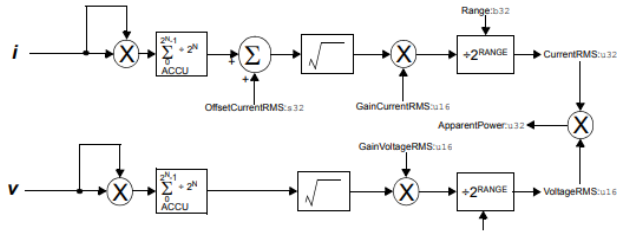


Fig. 2. RMS current and voltage calculation signal flow

The active power is obtained as average value of the multiplied instantaneous values of current and voltage, by using Eq. (4). Signal flow of active power calculation is shown in Fig. 3. The MCP39F521 has two simultaneous sampling A/D converters. For active power calculation, the instantaneous currents and voltages are multiplied in order to create instantaneous power. The instantaneous power is then converted to active power by averaging or calculating DC component.

$$P = \frac{\sum_{n=1}^{N} v(nT) i(nT)}{N} = \frac{\sum_{n=1}^{N} p(nT)}{N} . \quad (4)$$

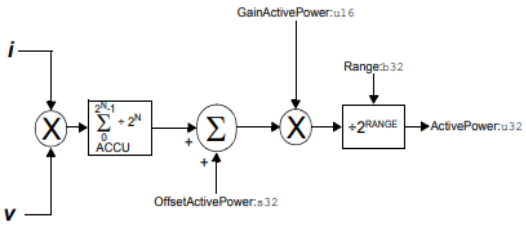
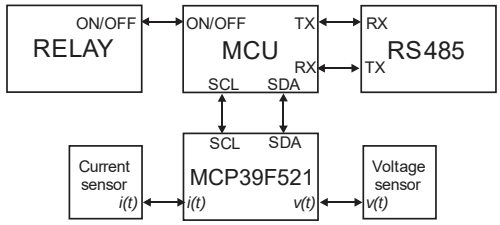


Fig. 3. Active power calculation signal flow

III. REALIZED SYSTEM

The block diagram and photography of our system are shown in Fig. 4. As can be seen from Fig. 4a our system consists relay circuit, RS485 circuit, current/voltage sensor circuit, power measuring circuit and MCU. We used a well-known microcontroller Atmega328P [10] as MCU, which characteristics meet all our demands. MCU communicates with power measuring circuit (MCP39F521) by using I2C protocol and passes the obtained data through RS485 to central monitoring system by using Modbus RTU protocol.



a)

b)

Fig. 4. a) Block diagram of realized system, b) photo of realized system

RS485 interface supports multiple devices on the same bus. This interface is currently widely used in data acquisition and control applications where multiple nodes communicate with each other. Consequently, each board needs to have unique address (unique on the network level). Because of that remote monitoring computer can send request only to specific node in a network. This address can be set by using jumpers in our system. The maximum number of nodes on the same RS-485 network in our case is 213 [11]. The number of supported devices on same networks depends on IC that is used for RS485 interface, bound rate and distance between nodes.

The load is powered via the T9A series relay, which has normal open (NO) and normal closed terminals (NC) [12]. The relay can be set to normal closed state, which is useful in applications where electrical appliance has to be always on, but on demand it can be turned off. Typical example of this usage is refrigerator in hotel rooms which needs to be always powered on. When customer exits the room, all electrical outlets and devices are disabled, except refrigerator which is connected via the relay board.

As we said before, communication between computer and

our device is done by using Modbus RTU protocol and QModMaster application. We use only four functions from Modbus RTU:*write single coil, write multiple coil, read input registers and read holding registers.*

The state of relay can be controlled by using *write single coil* or *write multiple coil* function, while reading measurement results is done by using *read input registers* or *read holding registers* functions. These registers contain value of RMS voltage/current, line frequency, power factor, active power and total accumulated energy, as shown in Fig.5.

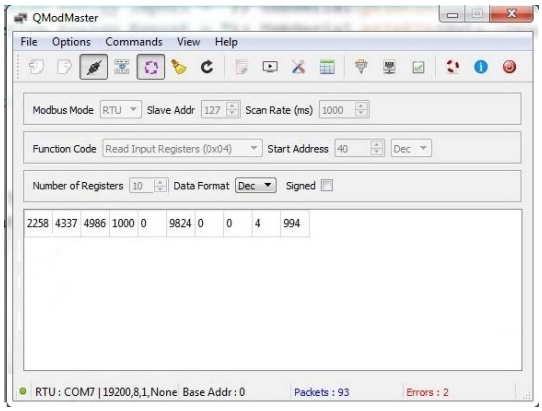


Fig. 5. Modbus application

QModMaster application runs on central computer which is used for reading data from each relay board. For connecting PC to RS-485 network USB-RS485 bridge is used. Before connecting relay board on the network, it is necessary to set slave address of the board, by setting appropriate jumper configuration. In Fig. 5, we showed obtained data when using *read input registers* function. By using this function and setting number of register to 10 we obtain: RMS voltage/current (registers 1 and 2), line frequency (register 3), power factor(register 4), active power(registers 5 and 6) and active energy(registers 7 to 10), respectively. In order to get good accuracy of read values, we introduce correction factors. This factor for voltage is 10, for line frequency and active power is 100, while for current, power factor and active consumed energy is 1000. The negative values of power factor are presented with digit 1 on the 5th position at register 4.

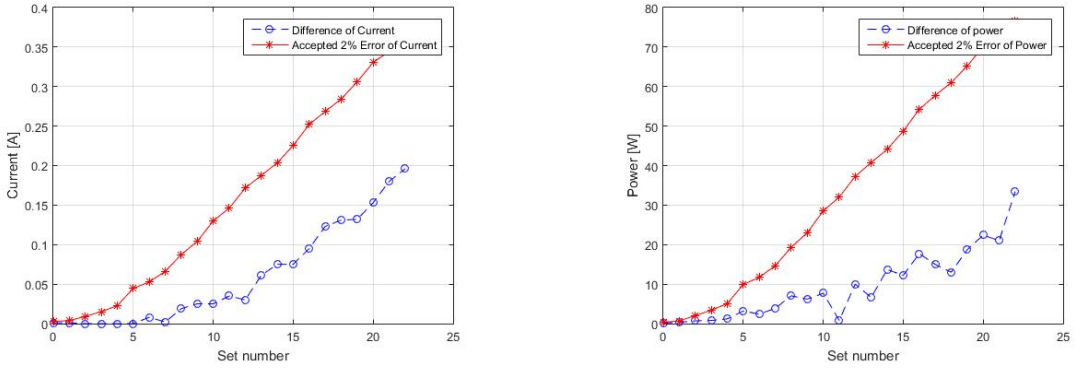
IV. Measurement Results

The accuracy of the realized system is verified by using a set of different linear and nonlinear loads. As nonlinear loads we chose LED and CFL bulbs. These nonlinear loads are chosen as benchmarks because they characterize small nominal power. Namely, the intention is to show that our system measures current in range from 0.1A to 20A with accuracy less than 2%, having different load conditions. For linear load measuring we have used industrial resistor whose resistance can be set in range from 16.66 **Ω** to 500 **Ω**, heater and different types of incandescent light bulbs (ILB). As reference measuring instrument, we have used electronic power meter produced by EWG electronics [13]. It fulfils the IEC 62052-22 standard [14], IEC 62052-23standard [15]. The previously mentioned standards, fulfilled by power meter, guaranty to us that power meter has good accuracy.

TABLE I
Measurement Results

		Power Meter			Our System					
NO.	TYPE OF LOAD	U_{RMS}	I_{RMS}	P (W)	U_{RMS}	I_{RMS}	P (W)	V_{RMS} Error(%)	I_{RMS} Error(%)	P Error(%)
1	CFL20W	225	0.135	17.7	224.8	0.134	17.8	0.09	0.75	0.56
2	LED10W+CFL20W	224.8	0.193	32.72	224.7	0.194	32.4	0.04	0.52	0.99
3	R=500Ω	225	0.455	102	225.00	0.455	102.70	0.00	0.00	0.68
4	R=300Ω	224.7	0.758	169.9	224.70	0.758	170.70	0.00	0.00	0.47
5	R=200Ω	224.6	1.131	253	224.20	1.131	254.20	0.18	0.00	0.47
6	R=100Ω	223.5	2.23	494.6	222.90	2.230	497.80	0.27	0.00	0.64
7	R=500Ω&ILB100W	223.1	2.632	586.2	222.6	2.640	588.6	0.22	0.30	0.41
8	R=70Ω	222.6	3.309	733.33	222.1	3.311	737.2	0.23	0.06	0.52
9	R=50Ω	221.44	4.368	966.37	221.1	4.387	973.4	0.15	0.43	0.72
10	R=50Ω&ILB200W	220.09	5.207	1150.2	220.05	5.232	1156.4	0.02	0.48	0.54
11	R=35Ω	219.8	6.495	1425.6	219.1	6.520	1433.3	0.32	0.38	0.54
12	R=35Ω&ILB200W	218.8	7.311	1600.6	218.2	7.346	1601.4	0.27	0.48	0.05
13	R=25Ω	217.6	8.574	1864	217	8.604	1874	0.28	0.35	0.53
14	R=25Ω&ILB200W	217.44	9.375	2038.9	216.5	9.436	2045.5	0.43	0.65	0.32
15	R=25Ω&ILB400W	216.73	10.171	2206	215.8	10.246	2219.7	0.43	0.73	0.62
16	R=25Ω&ILB650	216.03	11.272	2435.5	215	11.347	2447.7	0.48	0.66	0.50
17	R=16.66Ω	215.38	12.6	2711	214.3	12.695	2728.7	0.50	0.75	0.65
18	R=16.66Ω&ILB200W	214.7	13.424	2888	213.9	13.547	2903	0.37	0.91	0.52
19	R=16.66Ω&ILB400W	214.51	14.215	3050	213.1	14.346	3063	0.66	0.91	0.42

20	R=16.66Ω&ILB650W	212.85	15.3	3260	211.8	15.432	3278.7	0.50	0.86	0.57
21	R=16.66Ω&Heater&ILB200W	212.56	16.533	3506	210.9	16.686	3528.4	0.79	0.92	0.63
22	R=16.66Ω&Heater&ILB400W	211.4	17.22	3641	209.3	17.400	3662	1.00	1.03	0.57
23	R=16.66Ω&Heater&ILB650W	210.7	18.25	3833	209.1	18.446	3866.4	0.77	1.06	0.86



a) b)

Fig. 6. a)Accuracy of current for load shown in Table I , b) Accuracy of power for load shown in Table I

As it is shown in Table I and Fig 6.,after measurements with all the loads, we obtained accuracy less than 2%.

V. Conclusion

This paper presented a current and power monitoring system with accuracy less than 2%. Measuring of voltage, current, line frequency and active power for household appliances gives us sufficient information on which we can perform some action. Built-in relay with NO and NC contacts drastically increases practical usability of our system, so that remote controlling of appliance is also supported. This is very important in terms of creating a power scheme, so that some device can be powered only during night hours, or in case of some unpredicted behaviour we can switch off corresponding device, etc. In the future our goal will be to expand monitoring capabilities so that reactive and apparent power can be measured. Also, beside RS-485 interface, adding WiFi and/or Bluetooth capability will affect places where physically adding cables for RS-485 interface is not an option.

Acknowledgment

This work has been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

References

[1] "Energy Efficiency and its contribution to energy security and the 2030 Framework for climate and energy policy", https://ec.europa.eu/energy/sites/ener/files/documents/2014_eec_communication_adopted_0.pdf

[2] M.Clemence, R.Coccioni, A.Glatigny, "How utility electrical distribution networks can save energy in the smart grid era", Schneider Electric, April 2013

[3] M.AndrejevićStošović, D.Stevanović, P.Petković, "*Application of a Standard Power Meter for Detection Source of Harmonic Pollution and Reducing Economic Losses at Power Grid*, Electric Power Components and Systems, Vol. 48, Issue 1-2, Taylor & Francis, UK, 07.04., 2020, pp. 42-55, ISSN 1532-5008, doi:10.1080/15325008.2020.1731879

[4] D.Stevanović, M. AndrejevićStošović, M. Dimitrijević "*Improving the system for registration of electric energy consumption*, 2020 International Symposium on Industrial Electronics and Applications " INDEL 2020, Banja Luka, Bosnia and Hercegovina, 04.11.-06.11., 2020, pp. 1-5, ISBN 978-1-7281-9863-7, doi:10.1109/INDEL50386.2020.9266207

[5] https://nerdtechy.com/best-smart-home-energy-monitor

[6] J. G. Josué, J. M. Pina, M. V. Neves "Home Electric Energy Monitoring System: Design and Prototyping",Proceedings of Technological Innovation for Sustainability – Second IFIP WG 5.5/SOCOLNET Doctoral Conference on Computing 2011, Costa de Caparica, Portugal, 21 – 23. February, 2011, pp. 437 - 444.

[7] E. Barakat, N. Sinno, C. Keyrouz,"*A Remote Monitoring System for Voltage,, Current, Power and Temperature Measurements*", vol 55, no.1, pp. 421 – 428, 2014.

[8] M. J. Manti,A. V Bossche, R. F. Chisab "A Smart Voltage and Current Monitoring System for Three Phase Inverters Using an Android Smartphone Application", Sensors 2017, DOI:10.3390/s17040872

[9] https://ww1.microchip.com/downloads/en/DeviceDoc/20005442A.pdf

[10] https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061B.pdf

[11] https://www.ti.com/lit/ds/symlink/sn65hvd72.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-wwe&ts=1621931109406

[12] https://www.te.com/commerce/DocumentDelivery/DDEController?Action=srchrtrv&DocNm=1308242_T9A&DocType=DS&DocLang=English

[13] EWG - multi metering solutions, www.ewg.rs

[14] IEC 62053-22 Electricity metering equipment (AC) - Particular requirements - Static meters for active energy (classes 0.2S and 0.5S).

[15] IEC62053-23 (2003) Electricity metering equipment (a.c.) – Particular requirements – Part 23: Static meters for reactive energy (classes 2 and 3)

Matlab/Simulink 1D model of longitudinal wave propagation through piezoceramic rings

Igor Jovanović and Dragan Mančić

***Abstract*— One-dimensional (1D) model of piezoelectric elements enable fast prediction of performance, as well as a good insight into the behaviour of piezoelectric elements and the entire ultrasonic transducer during operation. In this paper, the Matlab/Simulink 1D model of piezoceramic rings that include only thickness oscillation modes is presented, while radial oscillations are neglected. Implementation of equivalent electromechanical circuits in the modelling of piezoelectric elements did not bring a larger amount of information in relation to the number of information obtained by applying constitutive piezoelectric equations. In addition, the presented model that directly relies on the constitutive piezoelectric equations enables better visualization of wave propagation through transducer structure. The input electrical impedances for piezoceramic rings are calculated using the realized model and then compared with experimental results to validate the model.**

***Index Terms*— Piezoceramic ring, Matlab/Simulink model, Input electrical impedance.**

I. Introduction

Models involving computer simulations have become an essential part of the transducer design process. Electromechanical equivalent circuits are frequently used to model and analyse ultrasonic transducers, whose application is based on the idea that wave propagation speed is equivalent to electric current, while mechanical force is equivalent to electric voltage [1]. Today, many transducers are designed with one dominant resonant mode, which can further simplify the models and justify their use. Piezoceramic rings of different thicknesses and inner/outer diameters are widely used as active components in ultrasonic sandwich piezoceramic transducers [2].

In case that the axial dimensions of ultrasonic transducers that oscillate in the thickness mode are larger than the radial dimensions, one-dimensional analysis can be applied in the modelling process of both piezoceramic rings and whole sandwich transducers [3], [4]. Although this dimensional relationship is common with most ultrasonic transducers, in the process of modelling transducers with a complex structure (e.g. composite transducers), it is important to predict the behaviour of transducers in all directions of oscillation propagation [5]. In this case, it is necessary to use three-dimensional models [6]. The components of these power transducers can be modelled by mathematical analysis with the help of appropriate physical laws.

In this paper, the Matlab/Simulink model of piezoceramic rings based on constitutive equations of piezoelectric effect is presented. The proposed Matlab/Simulink model leads to simpler implementation than the mathematical model. This model can also be used to analyse multilayer structures that include both piezoelectric materials and metal endings.

Igor Jovanović and Dragan Mančić are with the University of Niš, Faculty of Electronic Engineering, 14 Aleksandra Medvedeva, 18000 Niš, Serbia (e-mail: igor.jovanovic@elfak.ni.ac.rs and dragan.mancic@elfak.ni.ac.rs).

II. Description of Governing Equations

Constitutive equations for piezoelectric material can be written in the following form when neglecting transverse dimensions [7]:

$$E = -hS + \frac{D}{\varepsilon^S}, \tag{1}$$

$$T = c^D S - hD, \tag{2}$$

where the mechanical stress T can be determined by dividing the total extension or compression force F by the transverse surface P, $T=F/P$. E is the applied electrical field, S is the mechanical deformation, D is the dielectric displacement, c^D is the elastic stiffness coefficient, h and ε^S describe physical characteristics of piezoelectric material.

By applying Newton's II law, which defines force as a product of mass and acceleration, and using mass as a product of volume and density, mechanical stress is expressed as:

$$T = \rho l \frac{\partial^2 u}{\partial t^2}, \tag{3}$$

where ρ is the density, l is the thickness of the piezoelectric material, and u are the mechanical displacements components. By differentiating the last equation along the z-axis, it is obtained [8]:

$$\frac{\partial T}{\partial z} = \rho \frac{\partial^2 u}{\partial t^2}. \tag{4}$$

Expressing Hook's law in the following form [9]:

$$T = c^D \frac{\partial u}{\partial z}, \tag{5}$$

where the elastic stiffness coefficient c^D represents the coefficient of proportionality. The previous expression can be

rewritten using the propagation speed of longitudinal waves in a thin cylinder $v^2=c/\rho$, as:

$$\frac{\partial^2 u}{\partial t^2} = v^2 \frac{\partial^2 u}{\partial z^2}. \tag{6}$$

In 1D models, only mechanical deformation and divergence of the dielectric displacement along the z-axis are of interest:

$$S = \frac{\partial u}{\partial z}, \text{ and } \nabla D = \frac{\partial u}{\partial z} = 0. \tag{7}$$

By differentiating expression (2) along the z-axis, one may obtain:

$$\frac{\partial T}{\partial z} = c^D \frac{\partial^2 u}{\partial z^2} - h\frac{\partial D}{\partial z}. \tag{8}$$

By solving (6) in the general case, the equation of nonstationary wave motion is obtained in the form of a linear differential equation of the second-order with two independent variables (by z-axis and by time t).

If a multilayer structure shown in Fig. 1(a) is observed, the solution of this hyperbolic partial differential equation (determinant is $4v^2 > 0$) can be written in general form for layer n in the multilayer structure:

$$u_n = \alpha_n e^{-j\omega\Delta t_n} + \beta_n e^{j\omega\Delta t_n}. \tag{9}$$

Amplitudes of the mechanical displacements components in the direction of wave propagation are denoted by α_n, and in the opposite direction to the direction of wave propagation by β_n, on the surface of the n-th layer. The ultrasonic wave propagation time through the n-th piezoelectric layer is equal to the ratio of the length of the propagation path and the speed of wave propagation (for propagation through the entire layer $\Delta t_n = l_n/v_n$).

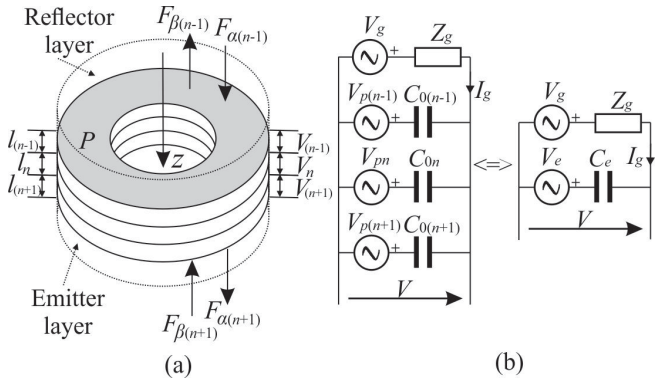


Fig. 1. (a) Multilayer structure of an ultrasonic sandwich piezoceramic transducer, (b) its equivalent circuit in the case of a parallel electrical excitation.

To obtain the analytical expression for the electric voltage V_n on the n-th piezoelectric layer, (1) can be written in the following form:

$$E_n = -h_n \frac{\partial u_n}{\partial z} + \frac{Q_n}{P\varepsilon_n^S}, \tag{10}$$

for $D=Q_n/P$ and static capacitance of the n-th layer $C_{0n}=\varepsilon_n^S P/l_n$. Q_n is the amount of accumulated charge stored on capacitor C_{0n}.

By integrating the last equation along the z-axis the following expression was obtained [9]:

$$\begin{aligned} V_n &= -h_n \int_0^{l_n} \frac{\partial u_n}{\partial z} dz + \frac{Q_n}{P\varepsilon_n^S} \int_0^{l_n} dz = \\ &= -h_n \left[\alpha_n \left(e^{-j\omega\Delta t_n} - 1 \right) + \beta_n \left(e^{-j\omega\Delta t_n} - 1 \right) \right] + \frac{Q_n}{C_{0n}}. \end{aligned} \tag{11}$$

Is useful to introduce mechanical displacements components u_n with the help of forces acting on the surfaces, which are perpendicular to the z-axis ($F_{\alpha n}$ and $F_{\beta n}$) [9]:

$$V_n = -\frac{h_n}{j\omega Z_{cn}} \left(F_{\alpha_n} + F_{\beta_n} \right) \left(1 - e^{-j\omega\Delta t_n} \right) + \frac{Q_n}{C_{0n}}, \tag{12}$$

where the specific acoustic impedance Z_{cn} and the amplitudes of the mechanical displacements components are represented as:

$$Z_{cn} = \frac{c_n^D P}{v_n}, \quad \alpha_n = -\frac{F_{\alpha_n}}{j\omega Z_{cn}} \text{ and } \beta_n = \frac{F_{\beta_n}}{j\omega Z_{cn}} e^{-j\omega\Delta t_n}. \tag{13}$$

The general analysis of the multilayer structure of an ultrasonic sandwich piezoceramic transducer includes several piezoelectric layers that act as sources or as sensors of oscillations. However, in this paper, the analysis is limited to the most commonly used electrical connection of piezoelectric layers in practice, to the analysis of single-layer or multilayer structures that are mechanically connected in series and electrically excited in parallel, Fig. 1(a). Fig. 1(b) shows that the voltage V of the parallel connection of all active layers consists of voltage contributions based on the piezoelectric effect V_{pn} and the amount of accumulated charge stored on capacitor C_{0n}. The equivalent capacitance of the parallel connection of N piezoelectric layers C_e, as well as the total amount of charge Q_e, which depends on the voltage of the electric generator V_g, and its impedance Z_g, can be represented by the expressions:

$$C_e = \sum_{n=1}^{N} C_{0n}, \quad Q_e = \frac{V_g - V}{j\omega Z_g}. \tag{14}$$

When the equivalent circuit of Fig. 1(b) is considered as a capacitive voltage divider, the equivalent voltage generated by the piezoelectric effect can be calculated as follows [10]:

$$V_e = \frac{V_g}{1+j\omega Z_g C_e} - \\ -\frac{j\omega Z_g C_e}{1+j\omega Z_g C_e}\sum_{n=1}^{N}\frac{C_{0n}}{C_e}\frac{h_n}{j\omega Z_{cn}}\left(F_{\alpha_n}+F_{\beta_n}\right)\left(1-e^{-j\omega\Delta t_n}\right). \quad (15)$$

If the structure has only one active layer, then the voltage on it is calculated by the expression:

$$V = \frac{1}{1+j\omega Z_g C_0}\left[V_g - j\omega Z_g C_0 \frac{h_n}{j\omega Z_c}\left(F_\alpha + F_\beta\right)\left(1-e^{-j\omega\Delta t}\right)\right]. \quad (16)$$

It is necessary to present mechanical deformation S in (2) as a derivative of mechanical displacements components u along the z-axis (7), and use expressions (13) for amplitudes of the mechanical displacements components to obtain analytical expressions for forces acting on piezoceramic surfaces [11]:

$$F_n = c_n^D P \frac{j\omega}{v_n}\left(\frac{F_{\alpha_n}}{j\omega Z_{cn}}e^{-j\omega\Delta t_n} + \frac{F_{\beta_n}}{j\omega Z_{cn}}\right) - h_n Q_n. \quad (17)$$

Equation (17) is an expression for the force on the external surfaces (perpendicular to the z-axis) of the n-th layer in the observed piezoceramic structure, in general form. Boundary conditions can be expressed based on continuity of forces and mechanical displacements components on contact surfaces:

$$F_n\big|_{z=l_n} = F_{n+1}\big|_{z=0} \text{ and } u_n\big|_{z=l_n} = u_{n+1}\big|_{z=0}. \quad (18)$$

When the ultrasonic wave passes from one layer to another, the speed of propagation changes. For explicit analysis of forces acting on the boundary surfaces between two layers, it is necessary to define the transmission and reflection coefficients during the passage of an ultrasonic wave from one layer (n) to another (n+1) [12]:

$$T_n^{n+1} = \frac{2Z_{c(n+1)}}{Z_{c(n+1)}+Z_{cn}} \text{ and } R_n^{n+1} = \frac{Z_{c(n+1)}-Z_{cn}}{Z_{c(n+1)}+Z_{cn}}. \quad (19)$$

Components of mechanical forces acting on boundary surfaces, with defined boundary conditions and taking into account the transmission and reflection of the ultrasonic waves, can be presented by the following expressions:

$$F_{\alpha_n} = \frac{1}{1-T_{n-1}^n K_n}\Big[F_{\alpha_{(n-1)}}T_{n-1}^n\left(e^{-j\omega\Delta t_{(n-1)}}-K_{n-1}\right) - F_{\beta_{(n-1)}}T_{n-1}^n K_{n-1} + \\ +F_{\beta_n}\left(R_n^{n-1}e^{-j\omega\Delta t_n}+T_{n-1}^n K_n\right) + \\ +\frac{T_{n-1}^n V}{2}\left(h_n C_{0n} - h_{n-1}C_{0(n-1)}\right)\Big], \quad (20)$$

$$F_{\beta_n} = \frac{1}{1-T_{n+1}^n K_n}\Big[F_{\beta_{(n+1)}}T_{n+1}^n\left(e^{-j\omega\Delta t_{(n+1)}}-K_{n+1}\right) - F_{\alpha_{(n+1)}}T_{n+1}^n K_{n+1} + \\ +F_{\alpha_n}\left(R_n^{n+1}e^{-j\omega\Delta t_n}+T_{n+1}^n K_n\right) + \\ +\frac{T_{n+1}^n V}{2}\left(h_n C_{0n} - h_{n+1}C_{0(n+1)}\right)\Big], \quad (21)$$

wherein $K_n = h_n^2 C_{0n}\frac{1-e^{-j\omega\Delta t_n}}{2j\omega Z_{cn}}$.

Indices in (20) and (21) indicate that for the first layer ($n = 1$) and the last layer ($n = N$), the values with indices $n - 1 = 0$ and $n + 1 = N + 1$, refer to the propagation medium behind the reflector layer and in front of the emitter layer, respectively.

The components of mechanical forces on the surfaces of the unloaded piezoceramic transducer that consists of only one piezoceramic layer (the influence of external forces on the transducer is equal to zero $F_{\alpha 0}$=$F_{\beta 2}$=0) are calculated as:

$$F_{\alpha_1} = \frac{1}{1-T_0^1 K}\left[F_{\beta_1}\left(R_1^0 e^{-j\omega\Delta t_1}+T_0^1 K\right)+\frac{T_0^1 V}{2}hC_0\right], \quad (22)$$

$$F_{\beta_1} = \frac{1}{1-T_2^1 K}\left[F_{\alpha_1}\left(R_1^2 e^{-j\omega\Delta t_1}+T_2^1 K\right)+\frac{T_2^1 V}{2}hC_0\right]. \quad (23)$$

Since this paper shows the modelling of only one active layer, which is unloaded (surrounding medium is air), external acoustic impedances are 400 Rayl, [11]. The value used for external acoustic impedances is much less than the specific acoustic impedance of the active layer, so it can be considered $T_n^{n+1} = 2$ and $R_n^{n+1} = -1$.

III. Simulation and Experimental Results

Expressions (15), (20) and (21) can form a system of equations that describes the electromechanical structure shown in Fig. 1. By solving this system of equations, numerical results are obtained that represent the mechanical forces in each of the layers of the packet transducer. In addition to knowing the characteristics of the electric generator, it is possible to calculate the input electrical impedance of the transducer itself.

Fig. 2 shows the Matlab/Simulink 1D model of one piezoceramic layer (a model of longitudinal wave propagation through piezoceramic rings). The model represents a system of equations formed by the Laplace transform (with related expressions for impulse delays $e^{-\tau s}$, differentiations s, and integrations $1/s$ with respect to time) of terms (16), (22) and (23).

The calculated and experimental results are obtained using a piezoceramic equivalent material. The dimensions of the used piezoceramic rings are given in the Table I, where l is the thickness, b and a are the inner and outer diameters of lead zirconate titanate (PZT) piezoceramic rings.

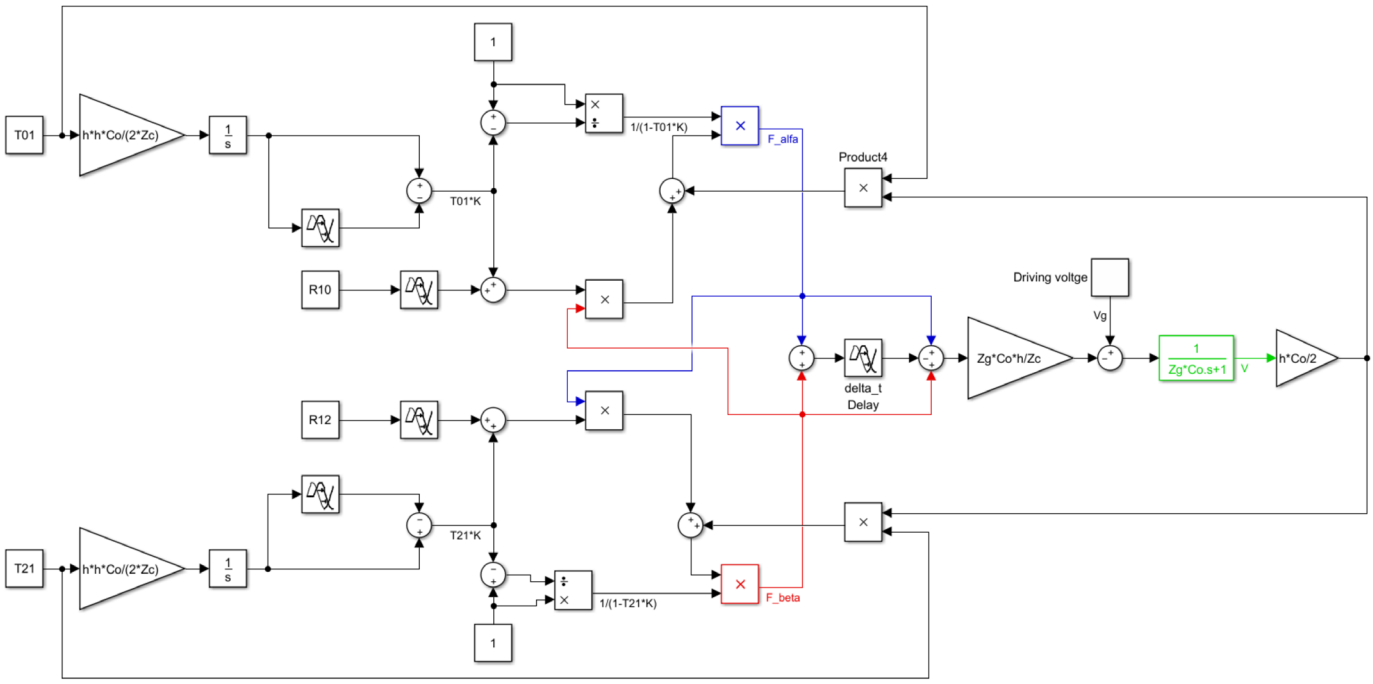


Fig. 2. Matlab/Simulink model of the piezoceramic ring.

Three samples of commercial PZT4 rings and three samples of commercial PZT8 rings have been characterized [13]. The electrical impedance measurements are conducted using an HP 4194A Network Impedance Analyzer.

TABLE I
PIEZOCERAMIC RING DIMENSIONS

Sample	*a* (mm)	*b* (mm)	*l* (mm)	*PZT equivalent material* [11]
I	38	15	5	PZT4
II	38	13	6.35	PZT4
III	50	20	6.35	PZT4
IV	24	15	3	PZT8
V	38	13	6	PZT8
VI	10	4	2	PZT8

As shown in Figs. 3-8, the measured frequency characteristic corresponds with the simulated curves using the proposed model. The proposed 1D model predicts only thickness modes while does not consider other mods. The images show the first thickness resonant mode for all used PZT samples.

The model acts as a three-port network whose ports refer to mechanical forces (F_{alfa} and F_{beta}) acting on the circular-ring surfaces of the piezoceramic ring, and the third port represents the electrical driving voltage (V_g). The charge components are proportional to the difference of mechanical displacements components between circular-ring surfaces and can be obtained including the equation (14) in the model.

These charge components further cause secondary forces to propagate through the transducer and the surrounding medium. The secondary piezoelectric effect was modelled using two positive feedback loops in Fig. 2.

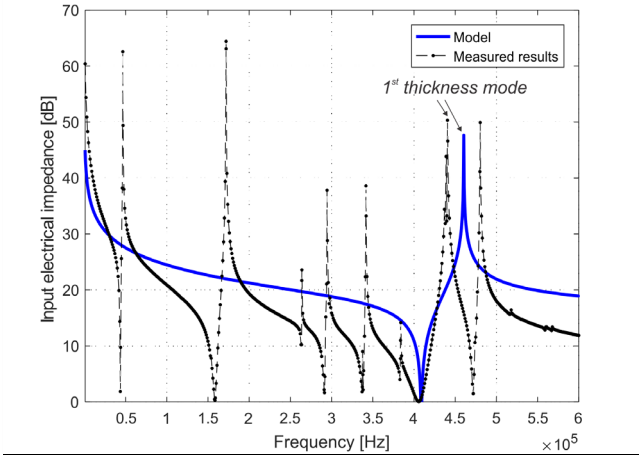


Fig. 3. Simulated and experimental input electrical impedance versus frequency for the I sample.

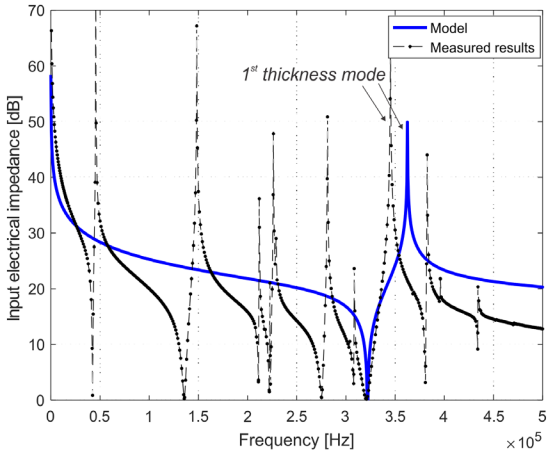


Fig. 4. Simulated and experimental input electrical impedance versus frequency for the II sample.

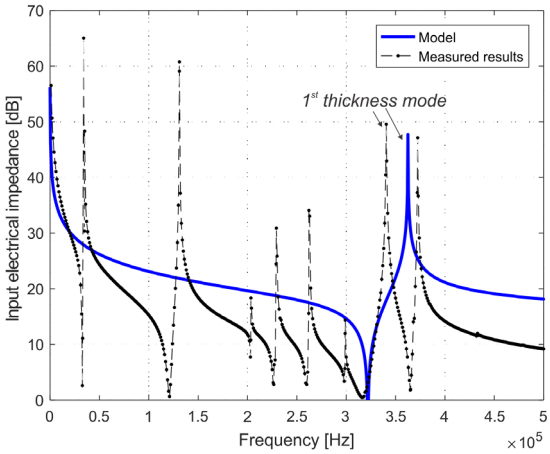


Fig. 5. Simulated and experimental input electrical impedance versus frequency for the III sample.

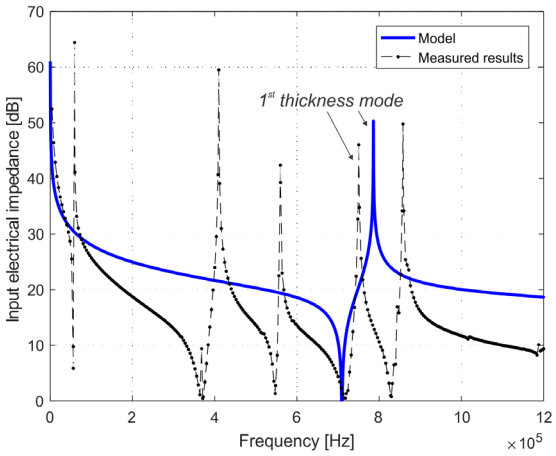


Fig. 6. Simulated and experimental input electrical impedance versus frequency for the IV sample.

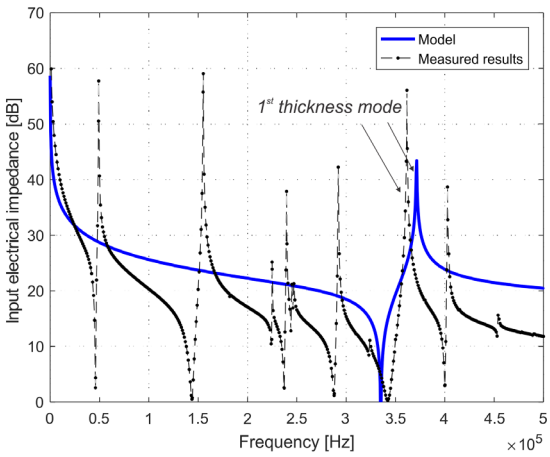


Fig. 7. Simulated and experimental input electrical impedance versus frequency for the V sample.

The driving current I_g from Fig. 1(a) was obtained by differentiating the charge components $I_g=sQ$. The input electrical impedance was obtained by dividing the voltage V from expression (16) and driving current.

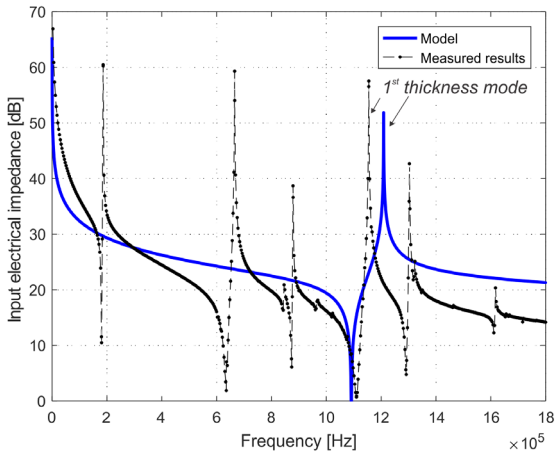


Fig. 8. Simulated and experimental input electrical impedance versus frequency for the VI sample.

IV. Conclusion

The piezoceramic rings with different dimensions are analysed using the developed model. Verification of the proposed model is performed by comparing the modelled dependencies of input electrical impedance vs. frequency with the experimental results. The matching of experimental and theoretical results is quite good and validates the proposed model. The presented model predicts thickness oscillatory modes of PZT piezoceramic rings taking the interaction with the surrounding media into account. The model describes the behaviour of the piezoceramic ring with two mechanical ports (one for each external surface normal to the z-axis) and one electrical port.

This approach is suitable for the analysis of the completely multilayer structure of the transducer. When calculating the forces acting on the passive layers of the transducer (materials that do not have piezoelectric properties), it is necessary to adopt that h=0.

This approach is not only effective in terms of computation time but also in reducing difficulties associated with the calibration of material parameters. Errors in predicting thickness oscillatory modes can be reduced by fitting the parameters of the piezoceramic material.

Acknowledgment

This work has been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

References

[1] D. Mančić, M. Radmanović, “Piezoceramic ring loaded on each face: a three-dimensional approach," Electronic Journal "Technical acoustics", vol. 2, pp. 1.1-1.7, 2002.

[2] I. Jovanović, D. Mančić, V. Paunović, M. Radmanović, Z. Petrušić, “A Matlab/Simulink Model of Piezoceramic Ring for Transducer Design," ICEST, Niš, Serbia, vol. 3, pp. 952-955, 2011.

[3] W. P. Mason, *Electromechanical Transducers and Wave Filters*, Princeton: Van Nostrand, 1948.

[4] A. Abdullah, M. Shahini, A. Pak, “An approach to design a high power piezoelectric ultrasonic transducer," J. Electroceram. vol. 22, pp. 369–382, 2009.
[5] I. Jovanović, D. Mančić, U. Jovanović, M. Prokić, “A 3D model of new composite ultrasonic transducer, "Journal of Computational Electronics, vol. 16, no. 3, pp. 977-986, 2017.
[6] D. Mančić, G. Stančić, “New Three-dimensional Matrix Models of the Ultrasonic Sandwich Transducers, "Journal of Sandwich Structures & Materials, vol. 12, pp. 63-80, 2010.
[7] Z. H. Zhang, B. Y. Sun, L. P. Shi, “Theoretical and Experimental Study on Secondary Piezoelectric Effect Based on PZT-5,” Phys.: Conf. Ser. vol. 48, pp. 86-90, 2016
[8] U. Jovanović, Lj. Perić, I. Jovanović, D. Mančić, “Analisys of longitudinal oscillations of Free prismatic piezoceramic beams", 24th International Conference "Noise and Vibration", Niš, Serbia, pp. 229 - 235, 29-31. October 2014.
[9] S. Cochran, C. E. M. Demore, C. R. P. Courtney, “Modelling ultrasonic transducer performance: one dimensional models," in *Ultrasonic transducers: materials and design for sensors, actuators and medical applications*, Cambridge: Woodhead Publishing Ltd, 2012. pp. 187-219.
[10] J. L. Dion, E. Cornieles, F. Galindo, K. Agbossou, “Exact one-dimensional computation of ultrasonic transducers with several piezoelectric elements and passive layers using the transmission line analogy, "IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control, vol. 44, no. 5, pp. 112031, 1997.
[11] I. Jovanović, “New Power Composite Ultrasonic Transducers,” Ph.D. dissertation (in Serbian), University of Niš, Faculty of Electronic Engineering, Niš, Serbia, 2018
[12] G. Hayward, M. N. Jackson, “A lattice model of the thickness-mode piezoelectric transducer," IEEE Trans. Ultrason. Ferroelec. Freq. Contr. Vol. 33, No. 1, pp. 41-50, 1986.
[13] *Technical Publication TP-226*, Morgan Electro Ceramics.

Arduino-Based Gas and Smoke Detector Realized Using MQ-2 Sensor

Petar Stančić, Aleksandra Stojković and Miljana Milić *Member, IEEE*

***Abstract*— System for detecting carbon monoxide, particles of smoke and combustible gases is described in this paper. Detector is based on application of Arduino platform with MQ-2 sensor as a source of input signal. The gas sensor functioning principles and detector configuration are explained in details. The described detector could be successfully used for various gas leakage detection, alone, or as a part of more complex system. The experimental results are given for burning paper smoke and three types of combustible gases. They confirm good performance of the system.**

***Index Terms*—Arduino, microcontroller, flammable gas detector.**

I. Introduction

We are the witnesses of rapid growth and wide application of electronic devices and systems in everyday life. The new disciplines and branches of technical sciences have emerged, which are now developing at an even faster pace. Electronic, mechanical and computer science engineering offer implementation of components that enable the development of complex systems that have a wide range of applications in industry, medicine and other fields of human life and work. Protection of human lives and health becomes a primary goal in modern conditions. The Arduino platform has an enormous potential as an educational and research tool, and represents an excellent base for designing important real-life systems. By using the Arduino platform, which consists of a physical part with a microcontroller and of a software, it is possible to design a system that is capable of protecting a human life and goods from fire and combustible gases.

II. System components

Key components for gas/smoke detector system are J1 piezo buzzer, a green and a red LED diode, an MQ-2 smoke detector and Arduino platform. Some of them will be explained in more details next.

A. Arduino

Arduino is an open-source electronics platform, based on easy-to-use hardware and software [1]. Over the years, it has been used for thousands of projects, from everyday circuits to complex scientific instruments. The entire Arduino project has started in 2004. when a Colombian student made a "Wiring" platform for his graduate thesis. In this way a new, low cost, and simple electronic device for fast prototyping was created. Arduino programs are written using a simplified version of C++, which makes it easier to learn. Arduino boards are very versatile and can be used for a variety of different applications. Some of them are: Uno, Due, Mega, Leonardo, Micro, Esplora etc. For the purpose of gas detection system, Arduino Uno was considered quite acceptable. Arduino Uno is the most frequently used variant, since it is very beginner friendly. It consists of 14-digital I/O pins, where 6-pins could potentially be used for the Pulse Width Modulation (PWM) outputs, 6-analog inputs, a reset button, a power jack, a USB connection, In-Circuit Serial Programming (ICSP) header etc., and – ATMega328 [2]. ATMega328 is a high performance AVR microcontroller with 8-bit RISC (*Reduced Instruction Set Computer)* architecture. It has low power consumption and can execute 131 instructions per single clock cycle. It has 32KB ISP (*In-System Programming*) flash memory with read-while-write capabilities, 2KB SRAM, 1KB EEPROM and maximum operating frequency of 20MHz.

B. MQ-2 gas and smoke detector

From 2013. to 2017., there were more than 350 thousand fires per year occurring only in homes, and that number is only a quarter of total number of fires. Smoke detectors are very much needed, since they can help in reducing the number of fires or at least decrease the damage done. Having any smoke detector is better than having none.

Best smoke detectors can detect smoke particles, flames and carbon monoxide. Smart smoke detectors represent a cutting-edge technology for fire safety, since they can communicate through the apps and deliver alerts to a phone or some other device or system. Smoke detectors should always have a backup power source for the case of power loss.

There are two basic types of passive smoke detectors: photoelectric and ionization [3]. Combination of these makes a dual sensor smoke alarm, which is recommended for maximum protection from both fast flaming and slower fires. Photoelectric alarms use light to detect smoke. They sense sudden scattering of light when smoke enters into the detectors chamber, which further triggers the alarm. This method of detection can detect fires that begin with long duration of smoldering aptly. Photoelectric smoke detectors respond from 15 to 50 minutes faster than ionization alarms in early stages of fire. Ionization alarms use radiation to detect smoke [4]. They carry a small amount of radioactive material

Petar Stančić, Aleksandra Stojković and Miljana Milić are with the University of Niš, Faculty of Electric Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: petar.stancic@elfak.rs, aleksmdv@elfak.rs, miljana.milic@elfak.ni.ac.rs

between two electrically charged plates, which ionizes the air and causes current to flow between the plates. The smoke disrupts the flow of ions and reduces the flow of current, which triggers the alarm. Ionization alarms are highly sensitive and respond from 30 to 90 seconds faster than photoelectric alarms to fast flaming fires. This type of detectors is more suited to rapid flaming fire outbursts, unlike the photoelectric detectors, which responds better to smoldering stages. Ionization detectors might perform better where there is risk of fast flaming fire, whereas photoelectric detectors react better to cases of slow smoldering, like electrical or furnishing fire. Dual smoke detectors include both photoelectric and ionization sensors, making them safest smoke and fire detection devices.

MQ-2 is a Metal Oxide Semiconductor (MOS) type of gas sensor, also known as chemiresistor. Chemiresistor is a material which changes its electrical resistance as a response to changes in the nearby environment. Sensors made from metal oxides require high temperatures (200 °C or higher) to operate because, in order for the resistivity to change, an activation energy must be overcome, thus leading us to the conclusion of MQ-2 being more efficient where there is risk of fast flaming fire, as the high temperature will be reached in shorter amount of time[7]. In our case, MQ-2 sensor's resistance changes when smoke or flammable gases are present. It requires 5V DC power supply and draws about 800mW of power. Beside the smoke, it can also detect Liquefied Petroleum Gas (LPG), alcohol, butane, propane, hydrogen, methane and carbon monoxide concentrations from 200 to 10 thousand ppm [5]. The sensitivity curve of MQ-2 sensor for different gases is shown in Fig. 1.

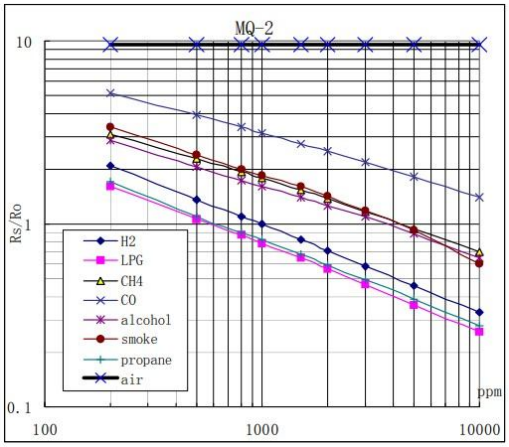


Fig. 1. MQ-2 sensitivity characteristic curve. [8]

Where R_0 is sensor resistance at 1000ppm (*parts per million*) of H_2 in the air, and R_S is sensor resistance at various concentrations of gases.

The sensor is sealed between two layers of stainless steel called anti-explosion network. It is necessary to ensure that the heater inside the sensor does not explode when flammable gases are sensed. The sensor also filters the air particles and allows that only gaseous elements pass inside the chamber. The anti-explosion network is attached to the enclosure with a copper plated clamping ring.

Fig. 2. MQ-2 external structure.

Figures 2 and 3 show the external and internal structure of the MQ-2 sensor.

Internal star-shaped structure is formed from sensing element and six connecting legs. Two of six leads, (H in Fig. 3.) are dedicated for heating the sensing element and connected through the nickel-chromium coil (conductive alloy). The remaining four leads (As and Bs in Fig. 3) generate the output signal and are connected using platinum wires which are linked to the sensing element and deliver small changes in the current that passes through the sensing element.

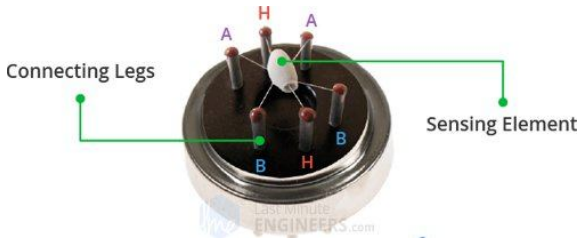


Fig. 2. MQ-2 internal structure. [9]

Sensing element is made of aluminum oxide and a tin dioxide coating. Tin dioxide coating is the most important part of the component since it is sensitive to combustible gases. Aluminum oxide increases the heating efficiency and ensures that the sensor reaches the working temperature.

If the air is clean, donor electrons from tin dioxide are attracted toward oxygen adsorbed on the surface of the sensing material, which prevents electric current flow as shown in Fig. 3. In the presence of smoke or combustible gases, oxygen reacts with gases which causes decrease of the surface oxygen density. Electrons from Fig. 5. are released back into the tin dioxide, allowing the current to flow through the sensor.

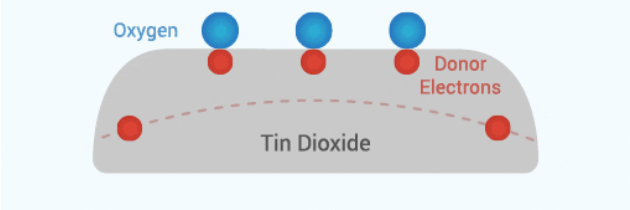


Fig. 3. Absorbed oxygen in clean air prevents the current flow.[9]

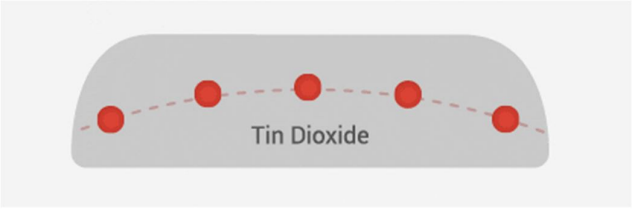


Fig. 4. Electrons are released into the tin oxide in the presence of

gas/smoke.[9]

The generated output voltage of the sensor is proportional to the concentration of the present smoke or gas. Higher gas concentration causes higher output voltage. The analog signal from MQ-2 is digitized as in the case of Arduino board based gas detection system implementation.

III. System Realization

For simulations and realization of this project, software environment "Fritzing" was used [6]. Fritzing is an open-source hardware initiative mainly used for documenting and sharing prototypes, layouts and manufacturing of Printed Circuit Boards (PCBs).

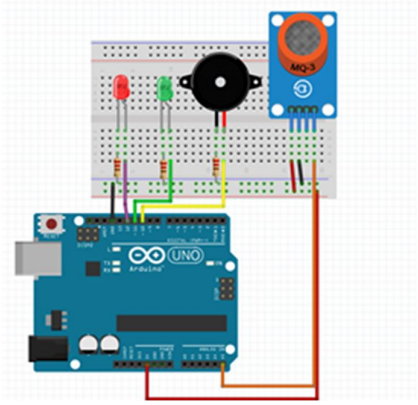


Fig. 5. Breadboard realization of the system using Fritzing.

Breadboard realization of the system using Fritzing is shown in Fig. 5.

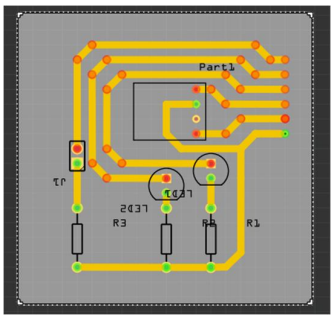


Fig. 6. PCB of our project.

PCB of the smoke/gas detection system is shown in Fig. 6.

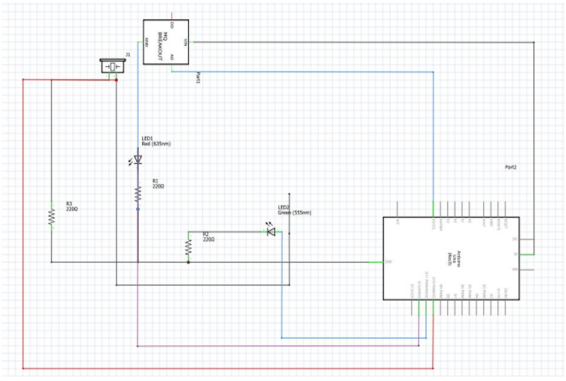


Fig. 7. Schematic of the system.

Schematic diagram of the system is shown in Fig. 7.

IV. The Experimental Results

The verification of the described detector system is performed in modest improvised laboratory environment. The greatest attention is paid to the fact that all measurements are performed in similar conditions as would be expected in real-live scenarios. In order not to endanger household safety, the measurements were performed inside a pan with a lid, as shown in Fig. 8. In this way a sufficient concentration of gases and good sensor response to small amount of gas is achieved.

Fig. 8. Experimental conditions.

Experiments show that smoke sensitivity of MQ-2 is considerably lower than its sensitivity to combustion substances (butane, alcohol and LPG in our case). This can be noticed from Fig. 10. After conducting multiple measurements, we could calculate the detector's sensitivity value (slope of the given characteristics) for different gases using the following equation:

$$u_{A5}(nT) = 5\frac{A5}{1024}\ [V], \quad T = 102\ ms \tag{1}$$

where u_{A5} is voltage on analog pin A_5 with maximal possible value of 5V. A_5 is a digital representative of measured voltage on the sensor with maximal value of 1024.

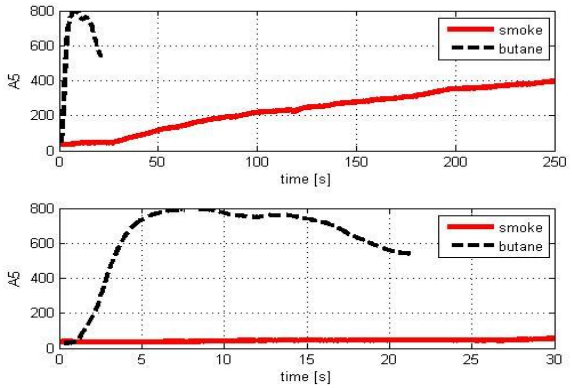


Fig. 9. Measured voltage for smoke and butane.

The relevant measure of sensor sensitivity could be the sensor's output voltage change rate $\frac{du_{A5}}{dt}$.

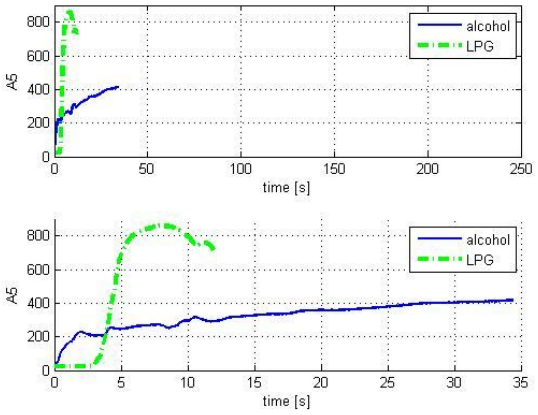


Fig. 10. Measured voltage results for alcohol and LPG.

That value is estimated from collected data, shown in Figs. 9 and 10, taking into account only part of the curve where the slope was constant and maximal.

- Butane: 0.9896 V*s^{-1}
- Alcohol: 0.5127 V*s^{-1}
- LPG: 1.8815 V*s^{-1}
- Smoke: 0.0145 V*s^{-1}

Measured voltage values were shown in *Serial Monitor* view in Fritzing, and it allowed us to track results in real time via serial data transfer (In this case it was set to 9600 bauds). Gathered data was imported in MATLAB after which it was used to generate the above characteristics. Those graphs show cumulative data. The latter graphs are zoomed and show curve parts where MQ-2 shows significant change when detecting larger amounts of gas.

One can conclude that MQ-2 is very sensitive to gases mentioned above. Considering the conditions in which experiment was conducted, obtained sensitivities cannot be considered as very accurate. The source of LPG was a 20l gas bottle. In short time it can deliver larger amounts of gas than a lighter (used as a source for butane). It is also noticed that the sensitivity to alcohol vapors is slightly lower.

Measured results are in compliance with data from Fig. 1, while maximal sensitivity is obtained for the LPG. Home conditions were not very satisfactory because the main priority was not to burn the sensor (small distance from the fire), while maintaining to "feed" the fire by removing the lid regularly. Real time information from Serial Monitor detected the smoke concentration drop with lid removal (to obtain more paper fuel) and this caused slow increase of sensor output voltage.

At the end we can conclude that with the right calibration, the MQ-2 can be used for detecting (and alarming) larger quantities of combustible gases in houses or storages.

V. Conclusion

In this paper one possible solution for realization of detector system for recognizing presence of smoke or combustible gases is presented. System is tested in home conditions with exposing MQ-2 sensor to different types of gases. Output sensor voltage is monitored for smoke generated by burning papers and cigarets and for available combustible gases as butane from lighter, stove LPG and alcohol vapors. Experiments prove high sensitivity of MQ-2 sensor making it a good choice in detector system basic sensor selection. By combining with other types of sensors, it is possible to create more complex detectors.

Acknowledgment

This work has been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

References

[1] M. Verde, "*Pic Mikrokontroleri*", Beograd : Mikroelektronika; 2008
[2] http://www.atmel.com/devices/ATMEGA328.aspx
[3] https://www.ifsecglobal.com/smoke-detectors/
[4] D. Malain, and P. Kanchana. "Evaluation of radiation safety for ionization chamber smoke detectors containing Am-241, "*Journal of Physics: Conference Series*", Vol. 1285. No. 1. IOP Publishing, 2019.
[5] https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/?fbclid=IwAR1gYvlT6VbUM_3bxXKsW8CnFyzmtqPiCwA6myb9DaOV_6P4SWNYBfN3sQM
[6] https://fritzing.org/
[7] O. Asif, Md. B. Hossain, M. Hasan, M. T. Rahman and M. E. H. Chowdhury, "Fire-Detectors Review and Design of an Automated, Quick Responsive Fire-Alarm System Based on SMS", *Int. J. Communications, Network and System Sciences*, Vol. 7, pp. 386-395, 2014
[8] *https://www.pololu.com/file/0J309/MQ2.pdf?fbclid=IwAR3zv-g1QSDJtT4FSUAfIC3BxyQVeRiqy7dfRQjnLVBjb440svr5GfqUwWI*
[9] *https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/*

A Chisel Generator of JTAG to Memory-Mapped Bus Master Bridge for Agile Slave Peripherals Configuration, Testing and Validation

Vukan D. Damnjanović, *Student Member, IEEE*, and Vladimir M. Milovanović, *Senior Member, IEEE*

***Abstract*—This paper presents a design of a JTAG to memory-mapped bus master bridge generator implemented using Chisel hardware design language. This type of digital module can provide convenient and practical means of configuring a wide range of peripheral circuitry with a memory-mapped slave interface attached to a bus interconnection, as well as of their testing and debugging. The peripherals can be configured by driving the input signals of the JTAG interface with the values that represent the previously defined instruction codes, thus initiating write or read data transactions on the interconnect bus through the master interface to their memory-mapped registers. The master interface can be either AXI4 or TileLink, depending on the characteristics of the whole system which the depicted bridge is a part of. The proposed generator offers the ability of creating slightly different modules by using different parameter selections. The implemented design has been extensively tested using various software simulations with a number of different slave peripherals and mapped and tested onto a commercial FPGA platform. These actions experimentally confirmed the previously made assumption of the utility and convenience of the proposed generator.**

***Index Terms*—JTAG to memory-mapped bus master bridge, AXI4 and TileLink protocols, memory-mapped interface, peripherals testing and debugging, Chisel hardware design language, design generator.**

I. Introduction

From the very beginnings of the integrated circuits and the emergence of the first microprocessors, memories and data storing were emphasized as one of the most important and vital parts of its structure because of a vast number of possibilities and numerous functions it provided. Almost all digital applications and devices were able to develop and operate on its basis. As the time passed by, with the improvement of the existing technologies and the emergence of the new ones, along with the development of microprocessors, those applications and devices started to become more and more complex and sophisticated as well. Therefore, not only that the limited capacities of the devices' memories appeared to be the major problem, but the ways of accessing their data were too, usually due to a need for the standardized methods or high performance criteria of the systems. Several ways of a microprocessor data access were developed over the years, with the usage of the port-mapped input/output (PMIO) and the memory-mapped input/output (MMIO) interfaces [1] being the most common ones.

The main characteristic of the port-mapped input/output data access interface is the presence of special address space outside the common system memory for every included peripheral. Usually, that implied the existence of special, dedicated instruction set for data access, such as "IN" and "OUT" instructions in x86 architectures [2]. The PMIO was more extensively utilized in earlier digital systems with less developed microprocessors with small address spaces, since the valuable resources were not consumed by the input/output (IO) devices. However, sometimes it is not convenient to use this kind of data access because of the possible frequent context switching or the need for the manipulation of IO devices using only standardized memory access instructions. Those features are delivered by using the memory-mapped input/output interface.

As mentioned before, systems with the MMIO have a shared virtual address space, along with the program memory or user memory, with the same instruction set for accessing it. All the devices are attached to an interconnect bus and from the perspective of the microprocessor, there is no difference whether it manipulates with the peripheral I/O device, or some internal data. Therefore, a wide range of different peripherals with the memory-mapped registers can be integrated in the system without almost any additional logic, thus allowing it to grow plentifully in terms of its functionality. Having this in mind, it is no wonder that modern-day systems more and more rely on this version of peripheral device's data access.

In the previous couple of paragraphs, the characteristics and the importance of the MMIO interface within the microprocessor-based systems were elaborated. For each one of those systems, manipulating the peripheral devices by accessing their data should be well-explained and straightforward. However, sometimes there is a need to manipulate or test those devices without implicating the microprocessor. In those cases, accessing the interconnect bus and initiating data transactions could be very challenging and complex. For that particular reason, the JTAG to memory-mapped bus master bridge generator from this paper's topic was designed and created. It allows a user to access the peripheral device connected to either AXI4 [3] or TileLink [4] bus without the engagement of any kind of processing unit, but by using the standardized and quite popular JTAG interface [5], [6].

Vukan D. Damnjanović is with NOVELIC d.o.o., Veljka Dugoševića 54/B5, 11060 Belgrade, Serbia (e-mail: vukan.damnjanovic@novelic.com).

Vladimir M. Milovanović is with the Department of Electrical Engineering, Faculty of Engineering, University of Kragujevac, Sestre Janjić 6, 34000 Kragujevac, Serbia (e-mail: vlada@kg.ac.rs).

This paper, along with the quick overview of the used protocols and detailed description of the design of the JTAG to memory-mapped master bus bridge and its implementation with the used Chisel libraries, also serves as the user manual of the module and depicts the obtained results through simulations and hardware implementation.

II. A JTAG to Memory-Mapped Bus Master Bridge, Its Interfaces and Instructions

A JTAG to memory-mapped bus master bridge [7], [8] is a digital component which initiates data transactions with the peripheral devices possessing the memory-mapped input/output interface, attached to an interconnect bus. This module communicates with the outer world through two ends and three possible different interfaces. The user sends the desired instruction by driving the signals on the user-end JTAG interface, present in all the variants of the module. The instruction, if performed correctly, then initiates the appropriate transaction on the bus-end interface, which can be either AXI4 or TileLink, depending on the user's preferences or the system requirements. In the next few paragraphs, a brief overview of these interfaces is provided.

A. JTAG Interface

The Joint Test Action Group (JTAG) is a standardized four-wire serial protocol usually used for testing and debugging integrated circuits through a JTAG port. It consists of three input signals and an output signal: Test Clock (TCK), which is used as a clock signal for a JTAG controller and is independent of system clock signal; Test Mode Select (TMS), an input signal which serves as a control signal for a JTAG finite state machine (FSM), which will be discussed later; Test Data Input (TDI), an input signal that represents the input serial data for JTAG instruction and data registers; Test Data Output (TDO), the serial output data for JTAG instruction and data registers (an additional output signal exists to express the validity of the output data on the TDO pin). The on-chip JTAG Test Access Port (TAP) implements the mentioned FSM, which is used for the realization of the JTAG subpart of the module.

The JTAG FSM is used to correctly accept the JTAG interface signals and to recognize whether the arrived data values are the instruction values or the data values itself. It is driven by the rising edge of the TCK signal and the state changing is controlled by the TMS signal. In certain FSM states, data from the TDI port is captured. From the idle state, by driving the TMS signal in the appropriate way, the user chooses to enter either the select data or the select instruction state and then the data/instruction capture state. From that point on, the procedure is identical for both instruction and data capturing. The only difference is that the captured values are stored in different registers. In the data/instruction shift state, values from the TDI port are stored in a shift register as long as the FSM stays in that state. Afterwards, the FSM enters the update data/instruction state and then returns to the idle state, with the complete data/instruction value stored in the appropriate register. A block diagram of the complete JTAG FSM with all the state changes and the TMS signal values is shown in Fig. 1.

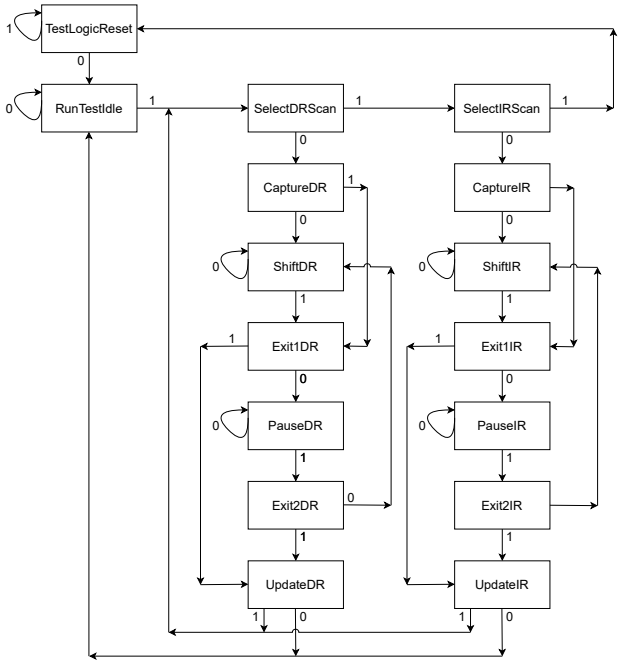


Fig. 1. A block diagram of the complete JTAG finite-state machine with all the state transitions and the TMS signal values.

B. AXI4 and TileLink Interfaces

Advanced eXtensible Interface 4 (AXI4) protocol is a parallel, synchronous, high-frequency multi-master and multi-slave communication interface. It is tailored mainly for the on-chip communication, which makes it suitable for the systems mentioned above. AXI4 interfaces consists of a vast number of different signals, with many of them optional, making it a versatile interface applicable to various different systems and applications. Even though it is described as multi-master and multi-slave interface, in every transaction only a single master and single slave communicate with each other. All the masters and the slaves are mutually connected through an interconnect bus. In the JTAG to memory-mapped bus master bridge module, this interface is used as a master interface and in most applications that include it, because of its sole purpose, the only active master is the module itself, whereas one or more slaves could exist. A simplified block diagram of an example of such a system is shown in Fig. 2. AXI4 interface protocol consists of 5 different channels: write address (AW), read address (AR), write data (W), read data (R) and response (B). Every one of those channels work on the handshaking principles and thus contain the pair of ready/valid signals. All the channels except the response channel have data signal among the various others signal whose function is to additionally describe and secure successful data transactions. Apart from the single read and single write data transactions, AXI4 interface supports the burst read and burst write data

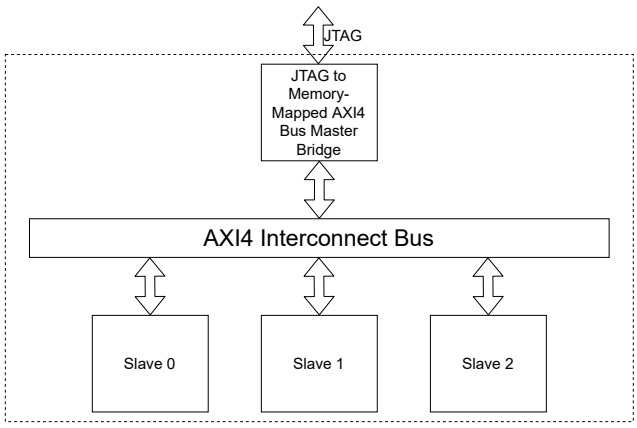


Fig. 2. A simplified block diagram of a system with a JTAG to memory-mapped bus master bridge and three slave peripherals attached to an AXI4 interconnect bus.

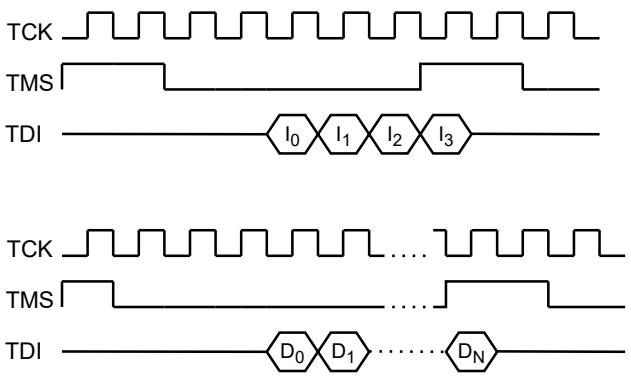


Fig. 3. JTAG input signal diagrams for correctly sending an instruction code (upper diagram) or a data value (bottom diagram) when the JTAG finite-state machine is in the idle state.

transactions.

TileLink is a parallel, synchronous, high-frequency multi-master and multi-slave communication protocol, in some extent similar to the AXI4 protocol. It is also designed mostly for the on-chip communication, with a special emphasis on the cache coherence transactions. The communication between a master and a slave (sometimes called a client and a manager in terms of this interface) is also performed based on the handshaking protocol. Overall, five communication channels can exist, with only two mandatory: channels A and D are mandatory, while channels B, C and E are optional. Each of the channels consists of several signals, including a data signal, a couple of ready/valid signals and some other signals used to describe and control transactions. The mandatory channel A flows from master interface to slave interface, carrying request messages sent to a particular address. Then, the slave responds to the master's request through the mandatory channel D. Other channels B, C and E are optional, and they are utilized for complete TileLink cached protocol, where channels B and C have similar functions as channels A and D respectively, whereas channel E is used as a final acknowledgment channel. In the module from the topic of this paper, however, only mandatory channels are used, and therefore, not much attention is provided to other three channels. TileLink protocol interface can be used as the master interface of the module instead of the AXI4 interface.

C. Defined Instructions

As it can be concluded from the previous paragraphs and sections, the JTAG to memory-mapped bus master bridge is a module that integrates two different interfaces with its dedicated controllers: JTAG and AXI4/TileLink. Those two controllers operate independently, they are even driven by different clock signals (TCK for JTAG controller and system clock signal for the AXI4/TileLink controller), but they are mutually synchronized through the internal signals. The JTAG controller works on the basis of the previously described JTAG FSM, while AXI4/TileLink has its own FSM. The whole module has the following, rather simple, signal flow. User calls an instruction by driving the serial JTAG input signals. The JTAG controller accepts the data, converts it to the parallel form and sends it to the AXI4/TileLink controller who, if it is recognized as a write instructions or a read instruction or any subvariant of them, initiates the transaction between the module and the appropriate slave. JTAG input signal values for correctly sending an instruction code or a data value when the JTAG FSM is in the idle state are shown on the timing diagrams in Fig. 3. In order to send the data value to the serial TDI input correctly, the least significant bit of the data should be sent first. It is strongly recommended that, prior to using the module, the user ensures that the JTAG FSM enters the reset state. It is achieved by driving the TMS signal with the active high value for five straight TCK cycles. By doing so, no matter what state was the JTAG FSM in, it will enter the desired reset state.

Total of four types of data transactions can be initiated by the AXI4/TileLink master interface on the interconnect bus: write, read, burst write and burst read. Also, nine instructions that the user can call through the JTAG interface are defined. The purpose of them is to enable the data transactions to be performed and to pass all the information needed for it. Each instruction can require either both instruction code and data itself to be provided, or just the instruction code. Both of them are captured in their dedicated JTAG FSM state. After the input serial data arrived entirely to the JTAG controller, it sends both the instruction code and the data to the AXI4/TileLink controller. A list of defined instructions, along with their codes and descriptions, is the following:

- 0x01 - Write instruction, initiates the AXI4/TL controller to begin writing the acquired data to the acquired address.
- 0x02 - Address acquire instruction, accepts the serial data as the address for the read/write instruction.
- 0x03 - Data acquire instruction, accepts the serial data as the data for the read/write instruction.
- 0x04 - Read instruction, initiates the AXI4/TL controller

to begin reading data from the acquired address.

- 0x08 - Number of burst transactions acquire instruction, accepts the serial data as the number of the read/write instructions during one burst transfer cycle.
- 0x09 - Burst write instruction, initiates the AXI4/TL controller to begin performing acquired number of the write transactions. Data is written to the consecutive addresses.
- 0x0A - Data index number acquire instruction, accepts the serial data as the index number of data to be acquired using the following instruction for the burst read/write transfer.
- 0x0B - Indexed data acquire instruction, accepts the serial data as the data at the acquired index number for the burst read/write transfer.
- 0x0C - Burst read instruction, initiates the AXI4/TL controller to begin performing acquired number of the read transactions. Data is read from the consecutive addresses.

Before the write instruction, both the address acquire and the data acquire instructions must be performed. Before the read instruction, the address acquire instruction must be performed. For the burst write instruction, the data for every single transaction must be acquired beforehand, as well as the total number of burst transactions for both the burst write and the burst read instructions. Two read/write/burst read/burst write instructions of the same type cannot appear sequentially one right after another, there must be at least one other instruction between the two. After performing the read or burst read instruction, read data appear on the serial output JTAG TDO data port, with the TDO driven signal having the active high value. All the instruction codes that are not mentioned in this paper can be assumed to be the no-operation (NOP) instructions.

III. A Design Generator and Its Implementation

Previously depicted JTAG to memory-mapped bus master bridge have been captured inside Chisel 3 hardware design generator. Both solely and in a combination with numerous slave peripheral modules, the generator has been thoroughly tested using standard Chisel verification and implementation paths for FPGA design flow. The design generator is made available [9] for public use as a free and open-source hardware library.

The generated module itself consists of two main subparts: JTAG controller and AXI4/TileLink controller, with their interfaces and internal communication signals. A block diagram of the JTAG to memory-mapped bus master bridge with all its interfaces and two main submodules is depicted in Fig. 4.

The JTAG controller is the submodule that communicates with the user. Its main purpose is to accept the serial data from the JTAG user interface, pack it in the appropriate format and send it to the AXI4/TileLink controller, as well as to accept the parallel data read from the slave peripherals and to put it on the JTAG serial data output. The JTAG controller is based on the previously depicted, standard JTAG FSM. Besides the

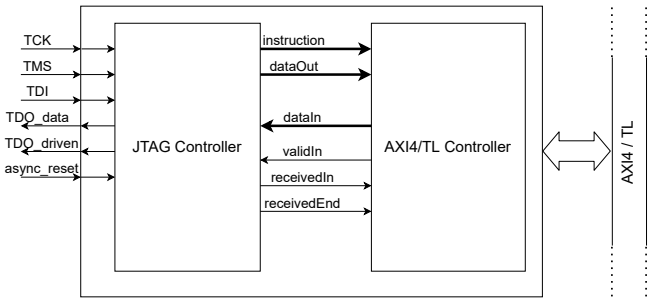


Fig. 4. A block diagram of the JTAG to memory-mapped bus master bridge with all its interfaces and the two main submodules.

common JTAG interface signals TCK, TMS, TDI and TDO, several more I/O signals exist. To begin with, TDO signal is divided into the one-bit-wide output signals: TDO data, which represents the serial output data, and TDO driven, which serves as the data valid signal. Those two signals are active at the falling edge of the TCK clock. There is also an asynchronous reset input signal which transits the JTAG FSM current state to the reset state.

Through the TDI JTAG serial input pin, the user can send either an instruction code or the data value itself. The JTAG can distinguish between those two thanks to the TMS control signals. Arrived data is translated from serial to parallel data format by using two shift registers, one for both instruction and data values. Data from the shift registers are stored into the two data registers, one at a time, when the JTAG FSM enters the appropriate data/instruction capture states. Data from those two registers are sent separately to the AXI4/TileLink controller.

A. The AXI4 and TileLink Controllers

The AXI4 and TileLink controllers are similar to one another, with the obvious difference in the master interface signals. They accept the instruction and data values from the JTAG controller, recognize the instruction code and take the action correspondingly. If the instruction code suits the either read or write instruction code, the controller initiates the communication with the appropriate slave peripheral through the AXI4/TileLink interconnect bus. The instruction code and data values are stored into the two separate registers. The value from the instruction register is constantly checked and compared to the instruction code of each of the four data transfer instructions (write, read, burst write, burst read). When those two values match, the appropriate flag value is set to the active high (a flag exist for every one of those four instructions) and that signifies that the appropriate instruction is set to be executed. Apart from signalizing that the instruction should be performed, the flag signals are used to prevent other instructions to be executed until the end of the current instruction. For the purpose of the burst transfers, a counter that counts the number of performed transfers is implemented inside the controller. Both controllers rely on their own FSMs

which secures the correctness of the communication with the peripherals.

The AXI4 and TileLink controllers also send the data read from the peripherals to the JTAG controller. For that purpose, several more internal signals exist. Apart from the one that carries data values to the JTAG controller, there are signal that marks the validity of the arrived data and two signals that mark that the JTAG controller has received the data and that the all bits of the data were sent to the output TDO pin.

The AXI4 controller FSM has the task to control the communication with the slave peripherals. State transitions are realized thanks to either flag values mentioned above, or the AXI4 signal values from the slave, such as ready signal for the handshaking protocol. Following states exist:

- `sIdle` - The idle state, the FSM stays in this state until the write instruction or the read instruction flag is set.
- `sSetDataAndAddress` - The state in which address is set on the AW channel, data is set on the W channel and valid signals are set on both the AW and W channels. The FSM stays in this state until the ready signals are not set on both the W and AW channels or until a counter which ensures that the FSM isn't stuck in this state counts out.
- `sResetCounterW` - The state in which the mentioned counter is reset. Stays in this state for exactly one clock cycle.
- `sSetReadyB` - The state in which the ready signal is set on the acknowledgement B channel. The FSM stays in this state until the B channel valid signal is not set or until the counter which ensures that the FSM is not stuck in this state counts out. The write instruction flag is reset in this state.
- `sSetReadAddress` - The state in which the address and the valid signals are set on the AR channel. The FSM stays in this state until the AR channel ready signal is not set or until the counter which ensures that the FSM is not stuck in this state counts out.
- `sResetCounterR` - The state in which the mentioned counter is reset. The FSM stays in this state for exactly one clock cycle.
- `sSetReadyR` - The state in which the ready signal is set on the R channel and data is read from the same channel. The FSM stays in this state until the R channel valid signal is not set or until the counter which ensures that the FSM is not stuck in this state counts out.
- `sDataForward` - The state in which the read data is forwarded to the JTAG controller, along with the active valid signal. The FSM stays in this state until the JTAG controller does not confirm that the data is received. The read instruction flag is reset in this state.

A state transition diagram for the AXI4 FSM is depicted in Fig. 5. Note that the states for the write and the read instructions only are shown. The reason for the deficiency of the other two is simply the similarity to the depicted ones. The only novelty for the burst transfers is the fact that after the completed single data transfer, FSM enters `sIdle` state only if the burst

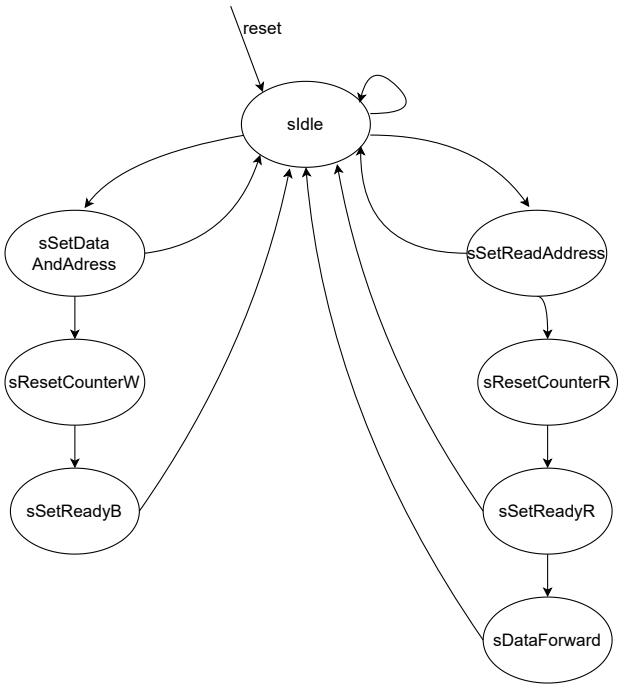


Fig. 5. A state transition diagram for the read and the write instructions of the AXI4 FSM.

transfers counter has counted out. Otherwise, the AXI4 FSM enters `sSetDataAndAddress/sSetReadAddress` state to perform another transfer.

The TileLink controller FSM has the same task as the AXI4 FSM. Its state transitions are also realized thanks to either the flag values or the signal values received from the slave peripheral. Although the states themselves are not identical to the ones from the AXI4 FSM, mostly because of the differences between the interfaces, the overall principles are the same. Therefore, they will not be elaborated in this paper.

B. The Chisel Generator

The Chisel generator of the JTAG to memory-mapped bus master bridge has few parameters that can impact the characteristics of the generated instances. Data and address buses widths for all the AXI4/TileLink channels can differ between 32 and 64. The instruction code width is also changeable. As the current number of instructions is nine, the width of four bits is sufficient for all the instruction codes. Another parameter represents the code for the initial instruction. It is strongly recommended that any code of the NOP instruction is provided as this parameter. Maximum number of transfers in a burst cycle can also take different values, as well as the set of the addresses that the module's master interface can access. Even though the Chisel language is extremely suitable for parameterization of the modules, this capability is not exploited a lot in this case due to the nature of the proposed module itself.

For the implementation of the generator, several exploited open-source Chisel libraries worth mentioning exist. Chip-

salliance's Rocketchip library [10] is extensively used. It provided the extremely valuable classes for the implementation of both AXI4 and TileLink master interfaces, as well as of the interconnect bus and memory-mapped address space. Also, the Ucb-art's Chisel-JTAG library [11] was beneficial for the realization of the module. Its JTAG FSM design with some other modules, such as shift registers and I/O bundles, were utilized. The generator itself was integrated into the Ucb-bar's Dsptools library [12].

IV. Implementation and Verification Results

There are several stages of the JTAG to memory-mapped bus master bridge testing. The first one represents the usage of the software simulations. For the performance of these tests, Chisel testers are utilized to drive the JTAG input signals. Apart from testing the module solely, it was also verified experimentally using various other modules with memory-mapped control and status registers, from the simple ones, such as a streaming multiplexer, to more complicated ones, such as a parameterizable numerically-controlled oscillator or run-time configurable fast Fourier transformation module. The tested module was also verified within the simulation environments with multiple slave modules.

Another stage of the proposed generator's verification is implementing and testing the generated instances on an FPGA-based development board. A Digilent's Arty A7 board with Xilinx Artix-7 FPGA family is used for it. All the generator's instances are synthesized for 100 MHz system clock frequency. The JTAG input signals were driven from the PC using the FTDI's C232HM-DDHSL-0 cable [13]. That cable contains the FT232H integrated circuit [14] and represents the USB 2.0 hi-speed to multi-protocol synchronous serial engine (MPSSE) cable. For the utilization of the FT232H chip and the cable itself, Pyftdi open-source library [15] is used. The JTAG input signals are generated as the general purpose input/output (GPIO) signals. JTAG clock frequency was set to 15 MHz (GPIO pins can work with the frequency up to 30 MHz) which is the convenient speed having in mind that the JTAG clock frequency is obligated to be lower than the system clock frequency in order for the module to work properly. Similar to the verification using software simulations, several additional modules with the memory-mapped registers were used to validate the functional correctness of the proposed generator's module.

The FPGA resource utilization for the JTAG to memory-mapped bus master bridge is not significant due to the lack of the complex arithmetic or logic operations and few used registers. Moreover, it is expected for the peripheral's utilized resources to be drastically more numerous.

V. Conclusion

In this paper, a generator of the JTAG to memory-mapped bus master bridge implemented using Chisel hardware design language is proposed. Mainly, it is used for configuring, testing and debugging the peripheral modules with memory-mapped input/output interface in the systems without the processing core or where the processing core is set to remain inactive in terms of communication with the slave peripherals through the interconnection bus. Even tough it seems to be a complete product right now, theoretically speaking a lot of space was saved for the further upgrade, primarily to the Chisel's parameterizable characteristics.

The generated instances of the JTAG to memory-mapped bus master bridge were tested and verified by both using software simulations and mapping onto a commercial FPGA development board. Numerous additional modules with the memory-mapped slave interface are utilized in the testing process. The module from the topic of this paper proved to be trustworthy regarding its functionality and performance.

Acknowledgements

The authors would like to thank NOVELIC d.o.o. for financially and logistically supporting the work on this project.

References

[1] N. Y. Song, Y. J. Yu, W. Shin, H. Eom, and H. Y. Yeom, "Low-latency memory-mapped I/O for data-intensive applications on fast storage devices," in *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, 2012, pp. 766–770.

[2] *Intel 64 and IA-32 Architectures Software Developer's Manual*, Order number: 325383-060us ed., Intel, September 2016.

[3] *AMBA AXI and ACE Protocol Specification*, Arm ihi 0022e (id033013) ed., ARM, February 2013.

[4] *SiFive TileLink Specification*, Tilelink specification, version 1.8.0 ed., SiFive, Inc, August 2019.

[5] B. V. Ngo, P. Law, and A. Sparks, "Use of JTAG boundary-scan for testing electronic circuit boards and systems," in *2008 IEEE AUTOTEST-CON*, 2008, pp. 17–22.

[6] L. Ungar, H. Bleeker, J. McDermid, and H. Hulvershorn, "IEEE-1149.x standards: achievements vs. expectations," in *2001 IEEE Autotestcon Proceedings. IEEE Systems Readiness Technology Conference. (Cat. No.01CH37237)*, 2001, pp. 188–205.

[7] *JTAG to AXI Master v1.2*, Pg174 ed., Xilinx, February 2021.

[8] *Embedded Peripherals IP User Guide*, UG-01085 ed., Intel, March 2021, pages 56-60.

[9] V. D. Damnjanović and V. M. Milovanović, "JTAG to memory master (JTAG2MM) chisel generator," www.github.com/milovanovic/jtag2mm, accessed: 2021/06/21.

[10] K. Asanović, R. Avizienis, J. Bachrach, S. Beamer, D. Biancolin, C. Celio, H. Cook, D. Dabbelt, J. Hauser, A. Izraelevitz, S. Karandikar, B. Keller, D. Kim, J. Koenig, Y. Lee, E. Love, M. Maas, A. Magyar, H. Mao, M. Moreto, A. Ou, D. A. Patterson, B. Richards, C. Schmidt, S. Twigg, H. Vo, and A. Waterman, "The rocket chip generator," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2016-17, Apr 2016. [Online]. Available: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2016/EECS-2016-17.html

[11] C. M. Richard Lin, Paul Rigge, "chisel-jtag," www.github.com/ucb-art/chisel-jtag, accessed: 2021/06/21.

[12] P. R. Chick Markley, Angie Wang, "dsptools," www.github.com/ucb-bar/dsptools, accessed: 2021/06/21.

[13] *C232HM*, version 1.3 ed., FTDI, document No.: FT000401 Clearance No.: FTDI 214.

[14] *FT232H*, version 2.0 ed., FTDI, document No.: FT000288 Clearance No.: FTDI 199.

[15] E. Blot, "PyFTDI," www.eblot.github.io/pyftdi/, accessed: 2021/06/21.

Allpass Based Double Notch IIR Filters with Constant Phase

Goran Stančić, Ivana Kostić and Stevica Cvetković

***Abstract*—Narrow stopband filters with two notch frequencies and piecewise constant phase are investigated in this paper. The notch filters are determined by allpass subfilter phase approximation. Obtained filters with simple and double poles are compared in conditions when fractional part of the coefficients is represented with limited number of bits.**

***Index Terms*—Notch IIR filters, allpass filters, phase approximation, constant phase, quantization.**

I. Introduction

THE ideal notch filters have exactly zero magnitude at frequency which need to be removed from input signal spectrum and unity gain otherwise. In many devices in practice the power line frequency signal and corresponding harmonics are often treated as noise [1]. Over time, digital electronic components become faster and cheaper allowing designers to oversample input signal. In that case the neighboring harmonics start to go closer to one another at the frequency axis. Notch filters are part of radar systems, control and instrumentation systems, medical applications and communications systems. In order to keep the distortion of desired signal as low as possible, the stopband of the notch filter should be as narrow as possible. In this paper problems associated with close notch frequencies will be observed. All of the presented results are given for double notch filters but it is easy to modify proposed method for arbitrary number of notch frequencies.

II. Realization structure

In addition to standard realization structures filters can be obtained by parallel connection of two allpass filters [2]. The magnitude of resulting filter depends on phases of applied allpass filters. That is a reason why design of the linear/constant phase IIR filters comes down to the allpass phase approximation problem.

In practice linear and constant phase are ultimate goals to avoid phase distortions [3]. To obtain the notch filter with approximately constant phase one allpass filter becomes direct path as shown in Fig. 1 [4]. The notch filter with linear phase will be achieved if one allpass filter is pure delay [5].

Goran Stančić, Ivana Kostić and Stevica Cvetković are with University of Niš, Faculty of Electric Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: ivana.kostic@elfak.ni.ac.rs

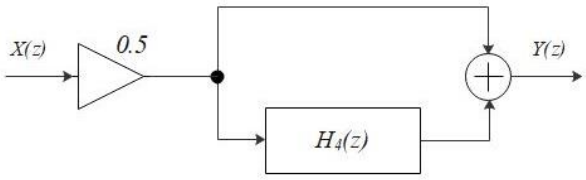


Fig. 1. Double notch filter realised as parallel connection of direct path and fourth order allpass filter.

The transfer function of constant phase notch filter is

$$H(z) = 0.5(1 + H_4(z)) \tag{1}$$

where $H_4(z)$ represents transfer function of allpass filter of the form

$$H_4(z) = \prod_{i=1}^{2} \frac{(\rho_i - e^{-j\theta} z^{-1})(\rho_i - e^{j\theta} z^{-1})}{(1 - \rho_i e^{j\theta} z^{-1})(1 - \rho_i e^{-j\theta} z^{-1})} \tag{2}$$

taking into account the fact that allpass filters have conjugate-reciprocal pole-zero pairs. Magnitude of the notch filter directly depends on the allpass filters phase φ

$$\left|H(e^{j\omega})\right| = \left|cos\,\frac{\varphi(\boldsymbol{\rho}, \boldsymbol{\theta}, \omega)}{2}\right| \tag{3}$$

where $\boldsymbol{\rho}$ and $\boldsymbol{\theta}$ represent moduli and phase angles of the allpass filters poles, respectively. Every pole and zero contribute to the phase with π/2 radians making fourth order filter to reach -4π radians phase at Nyquist frequency as shown in Fig. 2.

The closer a pole is to the unit circle the higher negative slope is at frequencies in vicinity of pole position. The phase is monotonically decreasing function of frequency with emphasized jump around pole position. Fourth order transfer function also could be obtained with two simple poles. This case is marked with d) in Fig. 2. Now poles are not at the same frequency and two separate phase jumps of approximately -2π radians could be observed.

According to (3), filter realized with described parallel structure possess passbands at frequencies ω where $\varphi(\boldsymbol{\rho}, \boldsymbol{\theta}, \omega)$ approximates $2k\pi$ with allowed tolerance ε, for $k \in \mathbb{Z}$. Stopbands would be obtained at frequencies where phase value is approximately $2(k+1)\pi$.

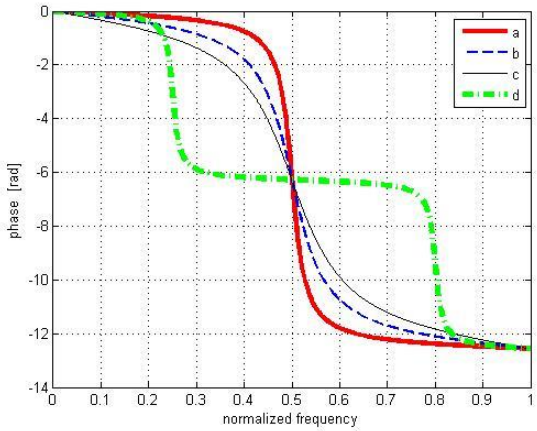


Fig. 2. Phase of allpass filter of fourth order for double pole moduli a) ρ=0.94, b) ρ=0.85, c) ρ=0.77 for θ=0.5π and d) simple poles $\rho_1 = 0.968, \rho_2 = 0.967$ for $\theta_1 = 0.25\pi$ and $\theta_2 = 0.8\pi$.

For predefined attenuation a in decibels, at stopbands or passbands boundary frequencies, allowed phase approximation error has value

$$\varepsilon = 2\arccos\left(10^{-a/20}\right) \tag{4}$$

III. Design procedure

If filter with two notch frequencies has double pole, to determine the transfer function only two unknown values need to be calculated- modulus and angle of the pole. The positions of notch frequencies are of highest importance, so one need to solve next system of equations

$$\begin{aligned} \varphi(\rho, \theta, \omega_{n1}) &= -\pi \\ \varphi(\rho, \theta, \omega_{n2}) &= -3\pi. \end{aligned} \tag{5}$$

Notch filter with two simple poles has four unknown parameters. It allows two boundary frequencies to be controled. System of equations that provide notch filter transfer function has now form

$$\begin{aligned} \varphi(\boldsymbol{\rho}, \boldsymbol{\theta}, \omega_{n1}) &= -\pi \\ \varphi(\boldsymbol{\rho}, \boldsymbol{\theta}, \omega_{n2}) &= -3\pi \\ \varphi(\boldsymbol{\rho}, \boldsymbol{\theta}, \omega_{l1}) &= -\varepsilon \\ \varphi(\boldsymbol{\rho}, \boldsymbol{\theta}, \omega_{r3}) &= -4\pi + \varepsilon \end{aligned} \tag{6}$$

where ω_{l1} and ω_{r3} represents boundary frequencies of the first and the third passband, respectively.

Instead of passband edges, it is possible to use stopband edges in system (6) but in (4) minimal attenuation in stopband need to be applied, with minimal modifications of last two equations in (6) ($-\varepsilon$ will be changed with $-\pi \pm \varepsilon$ and $-4\pi + \varepsilon$ with $-3\pi \pm \varepsilon$).

Systems of equations (5) and (6) could be solved applying some iterative procedure which demands the initial solution. System (6) is given in alternative form

$$\boldsymbol{A\Delta} = \boldsymbol{B} \tag{7}$$

after approximating phase $\varphi(\boldsymbol{\rho}, \boldsymbol{\theta}, \omega)$ by truncated Taylor series, where elements of matrix $\boldsymbol{A}$ are

$$a_{ij} = \begin{cases} \dfrac{d\varphi(\omega_i)}{d\rho_j}, & j = 1,2 \quad i = 1,..,4 \\ \dfrac{d\varphi(\omega_i)}{d\theta_{j-2}}, & j = 3,4 \quad i = 1,..,4 \end{cases} \tag{8}$$

elements of column vector $\boldsymbol{B}$ are

$$\boldsymbol{B} = \begin{bmatrix} -\pi - \varphi(\boldsymbol{\rho}^*, \boldsymbol{\theta}^*, \omega_{n1}) \\ -3\pi - \varphi(\boldsymbol{\rho}^*, \boldsymbol{\theta}^*, \omega_{n2}) \\ -\varepsilon - \varphi(\boldsymbol{\rho}^*, \boldsymbol{\theta}^*, \omega_{l1}) \\ -4\pi + \varepsilon - \varphi(\boldsymbol{\rho}^*, \boldsymbol{\theta}^*, \omega_{r3}) \end{bmatrix} \tag{9}$$

and vector of increments to be found is given with

$$\boldsymbol{\Delta} = [\Delta\rho_1;\ \Delta\rho_2;\ \Delta\theta_1;\ \Delta\theta_2] \tag{10}$$

In every iterative step system (7) is solved, the modulus and phase angle of poles are corrected until maximal absolute value of elements of column vector $\boldsymbol{\Delta}$ becomes less than predefined small value (in all given examples 10^{-10} is applied). As a good initial solution one could choose values

$$\rho^* = 0.9 \quad \text{and} \quad \theta^* = \frac{\omega_{n1} + \omega_{n2}}{2} \tag{11}$$

for double pole and

$$\boldsymbol{\rho}^* = [0.9; 0.9] \quad \text{and} \quad \boldsymbol{\theta}^* = [\omega_{n1};\ \omega_{n2}] \tag{12}$$

for simple poles. Extensive experiments shown that the final solution would be reached in less than ten iterations for arbitrary feasible input parameters.

IV. RESULTS

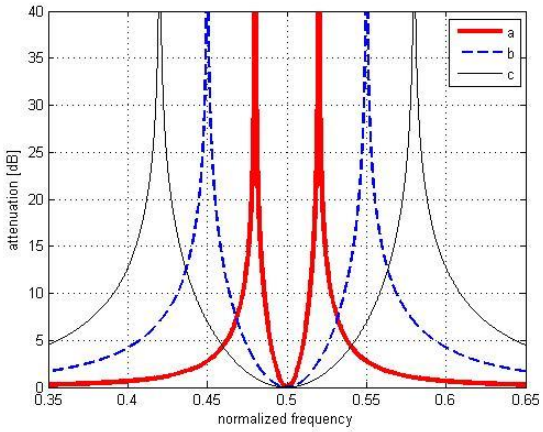


Fig. 3. The filters with two notch frequencies realized with double pole for a) $\omega_{n1} = 0.48\pi, \omega_{n2} = 0.52\pi$, b) $\omega_{n1} = 0.45\pi, \omega_{n2} = 0.55\pi$ and c) $\omega_{n1} = 0.42\pi, \omega_{n2} = 0.58\pi$.

In Fig. 3 are displayed characteristics of double notch filter attenuation for different zero magnitude frequencies. All filters have a double pole. Taking into account the phase characteristics shown in Fig. 2, to achieve more distance between the notches it is inevitable to move the pole closer to the origin. That is good for the stability because the pole moves further from the unit circle. Lower pole modulus values provoke lower phase slope, so the transition zones and stopbands become wider at the expense of passbands. This feature points to fact that double pole notch filter has restricted application. It is not possible to choose higher order allpass filter in attempt to improve notch filters characteristics, as in case of linear phase filters.

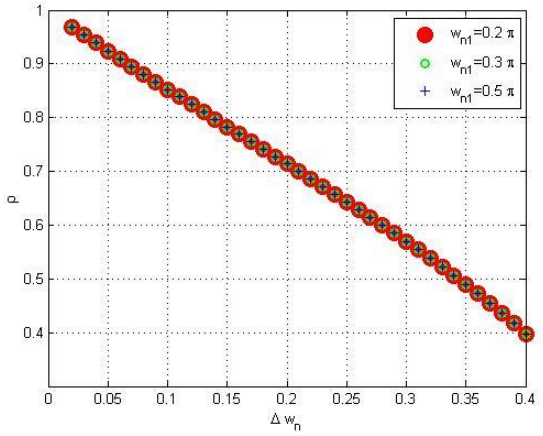


Fig. 4. The dependance of double pole modulus on notch frequencies gap for different ω_{n1} locations ($\Delta\omega_n = \omega_{n2} - \omega_{n1}$).

From Fig. 4 could be observed that most significant influence on double pole modulus has the notch frequencies gap. The very value of notch frequencies location have no visible impact. In practical realization of digital filter, the number of bits for filters coefficients representation need to be defined.

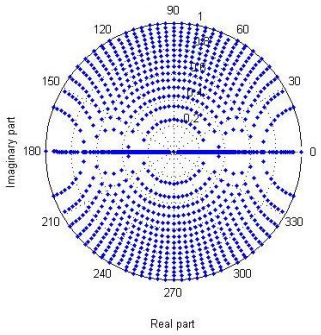


Fig. 5. Possible positions of filters poles for fixed point arithmetics when 4 bits are reserved for fractional part of transfer function coefficients.

Finite number of bits leads to rounded values of coefficients so realized filter characteristics just approximate derived ones. Possible positions of poles of the second order transfer function are displayed in Fig. 5 in case four bits are dedicated to fractional parts. In other words, calculated transfer function will be replaced with approximated one and obtained poles have to move from obtained positions to available locations like in Fig. 5.

The Fig. 5 indicates the fact that one can expect bigger error as consequence of quantization if notches are positioned at low and high frequencies.

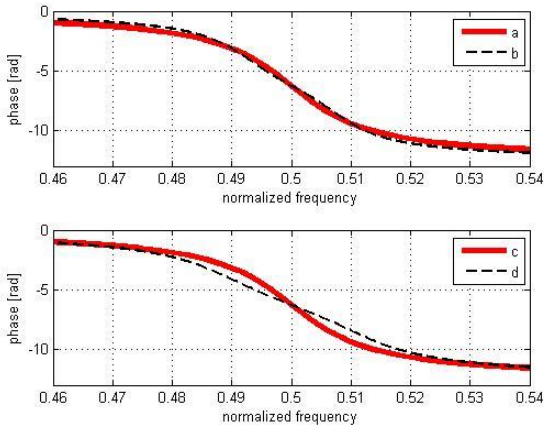


Fig. 6. Phase of notch filters ($\omega_{n1} = 0.49\pi$, $\omega_{n2} = 0.51\pi$) with a),c) double and b), d) simple poles before and after quantization fractional parts with 4 bits, respectively.

As first example filters with $\omega_{n1} = 0.49\pi$ and $\omega_{n2} = 0.51\pi$ are designed with desribed procedure. Attenuation of 1 dB is chossen in passbands. Boundary frequencies are ω_{l1}=0.47 π and ω_{r3}=0.53 π. Corresponding phase, attenuation and poles location are presented in Fig. 6, Fig. 7 and Fig. 8, respectively.

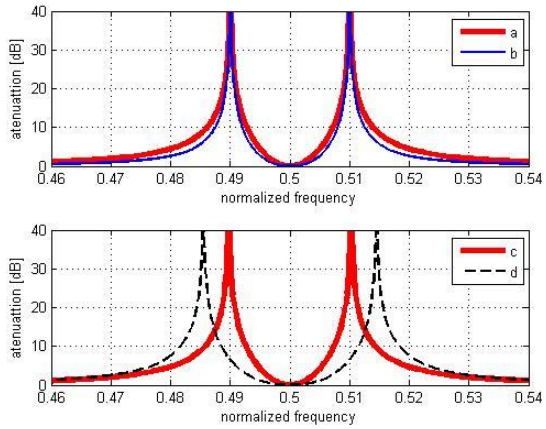


Fig. 7. Attenuation of notch filters ($\omega_{n1} = 0.49\pi$, $\omega_{n2} = 0.51\pi$) with a), c) double and b), d) simple poles before and after quantization fractional parts with 4 bits, respectively.

As it was expected, the phase undergo changes as repercussion of quantization, causing notch frequencies to displace. The notches are misplaced for 0.0142 for simple poles and $9 \cdot 10^{-4}$ for double pole case. Symmetry helps double pole filter less to degrade. The reason can be found in Fig. 5, where one can observe that pole with θ=0.5π will change only pole modulus as given in Fig. 8. Simple poles changed both moduli and phase angles causing significant mismatch between desired and obtained notches.

All obtained filters are realized as serially-cascaded second-order sections. Denominator coefficients of second order sections of the allpass filter with simple poles are given in Table I. All presented results are obtained in Matlab. The coefficients of allpass filter with double pole are presented in Table II.

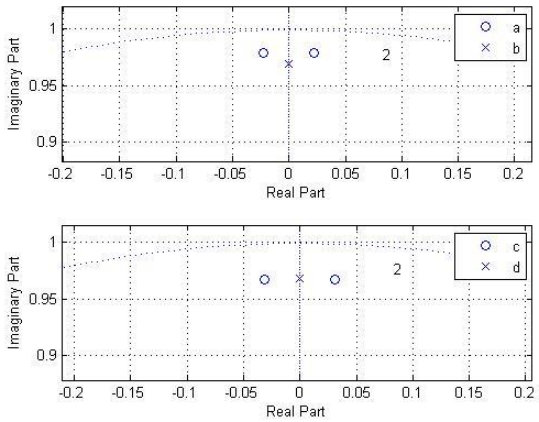


Fig. 8. Location of notch filters poles ($\omega_{n1} = 0.49\pi$, $\omega_{n2} = 0.51\pi$) with a), c) simple and b), d) double poles before and after quantization fractional parts with 4 bits, respectively.

TABLE I
COEFFICIENTS OF THE SECOND ORDER SECTIONS (SIMPLE POLES)

1.	0.0451	0.9581
1.	-0.0451	0.9581

TABLE II
COEFFICIENTS OF THE SECOND ORDER SECTIONS (DOUBLE POLE)

1.	0.	0.9391
1.	0.	0.9391

After quantization, with four bits dedicated to the fractional part, new values for second order sections coefficients are obtained as given in Table III. Table IV contains coefficients of allpass filter with a double pole. Because of existing symmetry second order sections have one coefficient equal to zero demanding less multipliers and adders in hardware realization.

TABLE III
COEFFICIENTS OF THE SECOND ORDER SECTIONS AFTER QUANTIZATION (SIMPLE POLES)

1.	0.0625	0.9375
1.	-0.0625	0.9375

TABLE IV
COEFFICIENTS OF THE SECOND ORDER SECTIONS AFTER QUANTIZATION (DOUBLE POLE)

1.	0.	0.9375
1.	0.	0.9375

For second example filters with $\omega_{n1} = 0.10\pi$ and $\omega_{n2} = 0.11\pi$ are chosen. For design procedure values ω_{l1}=0.09 π, ω_{r3}=0.12 π and a =3 dB are adopted. Obtained phase, attenuation and poles location are presented in Fig. 9, Fig. 10 and Fig. 11, respectively. These filters have poles in area where possible pole locations are scattered. As consequence, quantization of filter coefficients will seriously degrade characteristics. Close notches demand poles to be near the unit circle to provide enough steep slope.

Even 5 bits dedicated to the fractional part was not enough to stop a pole at the end of the unit circle. From (2) it is obvious that in such a case influence of the allpass zero and pole is identical, forcing the transfer function to degrade to second order. As a result, filter possess only one notch, as given in Fig. 10 d). The double pole filter has two notches after quantization but rear possible pole positions considerably influence the notches to displace.

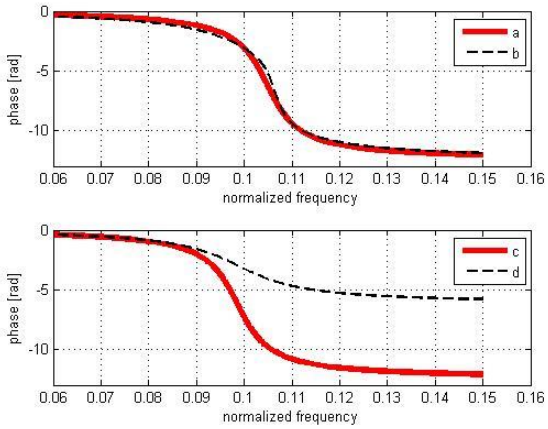


Fig. 9. Phase of notch filters ($\omega_{n1} = 0.10\pi$, $\omega_{n2} = 0.11\pi$) with a),c) double and b), d) simple poles before and after quantization fractional parts with 5 bits,respectively.

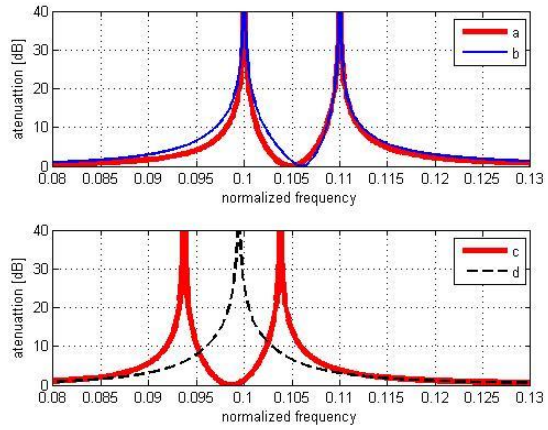


Fig. 10. Attenuation of notch filters ($\omega_{n1} = 0.10\pi$, $\omega_{n2} = 0.11\pi$) with a), c) double and b), d) simple poles before and after quantization of fractional parts with 5 bits, respectively.

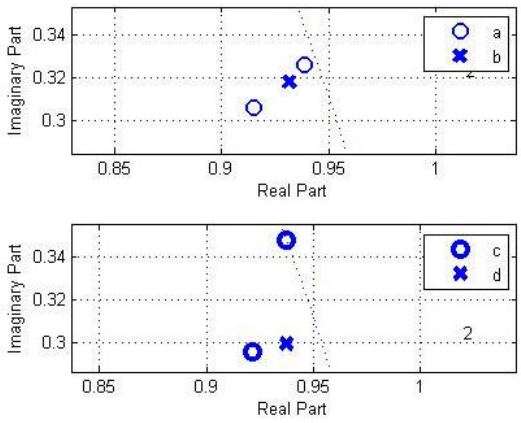


Fig. 11. Location of notch filters poles ($\omega_{n1} = 0.10\pi$, $\omega_{n2} = 0.11\pi$) with a), c) simple and b), d) double poles before and after quantization fractional parts with 5 bits, respectively.

V. Conclusion

Double notch filters with constant phase in passbands are investigated in this paper. Two similar solutions are compared and impact of quantization is analyzed. Parallel allpass structure guarantee low passband sensitivity. Quantization effects primarily affect notch filters stopband, moving away locations of notches from desired positions. The double pole filters are not good choice in case when gap between notches is wider than 0.2π because low phase slope causes transition zones to spread, degrading selectivity. On the other hand, the pole has lower modulus if filter possess double pole, what is guarantee to remain stable after quantization and still to have both notches. The distance of simple pole from the unit circle is always smaller compared to double pole. As consequence, after quantization if notches are close to each other it may occur one or both simple poles to finish at the unit circle and quantized version of filter lose selectivity. Design of numerous notch filters with different notches location have been shown that double pole solution is better option for close notches and small number of bits dedicated to the coefficients fractional part.

Acknowledgment

This work has been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

References

[1] P. Zahradnik and M. Vlček, "Notch filtering suitable for real time removal of power line interference", *Radioengineering*, vol. *22*, no. *1*, pp. *186-193*, 2013.

[2] P. P. Vaiduanathan P. A. Regalia and S. K. Mitra, "A new approach to the realization of low sensitivity IIR digital filters", *IEEE Trans. on Acoust., Speech and Sig. Proc.*, ASSP-34, pp. *350-361*, 1986.

[3] S. C. Dutta Roy, B. Kumar and S. B. Jain, "FIR notch filter design – A review", *Facta Universitatis ser. Electronic and Energetics*, vol. *14*, no. *3*, pp. *295-327*, 2001.

[4] G. Stančić and S. Nikolić, "Design of narrow stopband recursive digital filters", *Facta Universitatis ser. Electronic and Energetics*, vol. *24*, no. *1*, pp. *119-130*, 2011.

[5] S. Nikolić , G. Stančić and S. Cvetković, "Design of nearly linear phase double notch digital filters with close notch frequencies", *IET Signal Processing*, vol. *12*, no. *9*, pp. *1107-1114*, 2018.

Free/Open Source EDA Tools Application in Digital IC Design Curricula

Aleksandar Pajkanovic, *Member*, *IEEE*

***Abstract*—This paper represents a report on how free and open source software EDA tools may be used to organize a digital integrated circuits design course at the university level, without any financial investments in licenses. The course is built around several publicly available processor cores. These are of different complexity so first intrinsic properties are investigated. Then, using more complicated designs we examine how to increase performance through pipeline and cache associativity configurations. In this way we introduce RISC-V ISA and Chisel into the curricula. Finally, we provide a short overview of tools for automated design, from RTL all the way to silicon.**

***Index Terms*—open source, digital design, RISC-V, Chisel**

I. Introduction

Since the first implementation of a transistor was demonstrated at the Bell Labs back in mid-20th century [1], the industry has shifted from traditional, as established by the Industrial Revolution, to an economy based on the information technology (IT). That event, we understand today, represents the onset of the Information Age–the age characterized by rapid growth and development in all areas of life, driven by the semiconductor industry. Its workhorse, the CMOS technology process, is characterized by low power consumption, extreme scalability and ease of mass production. The ability to implement an idea, a solution using this technology, i.e. the ability to design integrated circuits (IC), or chips, has been of greatest importance for decades and it will be even more so in decades to come; namely, even though the Moore's Law [2] has ended–i.e. we do not advance our ICs by scaling anymore, but rather using advanced techniques [3] in design and verification phases, in order to achieve better performance–we still fabricate our solutions in silicon CMOS technology process.

Over the last five years, we have been establishing IC courses at the Faculty of Electrical Engineering in Banja Luka. This effort has been reported in [4], [5] and has recently culminated in fabrication ready circuits [6]. All this has been achieved using exclusively free and open source EDA software tools, with the help of many contributors - both students and tool developers. In those developments, however, main effort was in the analog domain, whereas in this particular paper we focus on digital IC design course, the examples it's built upon and tools used.

Since materials such as combinatorial and sequential circuits are covered in other courses, for this particular course we've decided to learn about more advanced concepts using the free intellectual property (IP) available in the community. Thus we study three processor designs, we learn a new approach to digital design (hardware construction, instead of description) and we drive the RTL code all the way to GDS, i.e. silicon ready file, using the free and open source toolchain.

In the next section we present motivation for writing up this paper, then we provide brief overviews of the RISC-V Instruction Set Architecture (ISA), *Chisel* - the hardware construction language (HCL) and processor cores we learn about and use to demonstrate theoretical concepts. Finally, in the sixth section, we present the free/open source digital IC design toolchain.

Aleksandar Pajkanovic is with the Faculty of Electrical Engineering, University of Banja Luka, Patre 5, 78 000 Banja Luka, The Serbian Republic, Bosnia and Herzegovina, e-mail: aleksandar.pajkanovic@etf.unibl.org

II. Motivation

There is no point in living during the Information Age, unless we are going to use its benefits. While boundaries are important parts of our lives and some should never be crossed, there are those boundaries that are not actually natural - these simply existed due to the fact that we knew not how to overcome them. This is not the matter of destroying those boundaries, but rather outgrowing them to improve the world and general quality of life. Such borders are those related to knowledge. With the Internet and its omnipresence - knowledge is omnipresent as well, for those who seek it. Our idea is to build the IC curricula by standing on the shoulders of the giants–of those who have had the chance to grow and develop for decades in this domain, thanks to another kind of boundaries. Therefore, we bring what's best on this planet right in our own court and thus enable our own students right here in Banja Luka to gain world-class expertise in the semiconductor industry, which, after all, is the most sophisticated commercially available technology process. And we do that without any financial investments - but rather simply: by reaching out.

Main motivation behind this paper is to contribute to the open source hardware community by sharing collected experiences and provide feedback on a subset of freely available tools and IPs, for all those who find themselves struggling to get started, at no cost, in this extremely interesting and exciting engineering and science area.

III. RISC-V

Computer architecture is, as most engineering areas nowadays are, an incredibly vast discipline. However, for the purposes of this short article, we will overly simplify and point out that it can be divided in two subdiciplines: software and

hardware. ISA is probably the most important interface in the universe as it serves to connect these two worlds. This has become clear with the IBM 360 appearance on the market, when the concept of ISA was first introduced. All ISAs with significant usage are proprietary, which does make sense when taking into account that the creators protect their intelectual property. However, *open* standards (the correct term would be open-source - to maintain analogy with software, where it all started, but there's no source code nor source files in this context, so we just call them open standards) like ethernet have proven successful. Successful, meaning that we have seen free-market competition through technical improvements, whence the end-users benefited the most [7]. Therefore, it is a crucial question to raise: why not create an open standard for the most important abstraction layer, the ISA?

In an answer, RISC-V (where V is a roman number five, thus pronounced) ISA has been designed - a completely free and open ISA, built and improved on the original RISC architectures. While it was designed originally for research and teaching, it is on its road to become a standard for industry implementations, as well. The final result is a simple and modular ISA, well suited to both high-performance systems and low-power embedded controllers. This is enabled by dividing the instructions into the obligatory base subset of ISA, present in any implementation, plus the optional extensions (subsets). The base is restricted to contain a minimum number of instructions sufficient for compilers, assemblers and linkers. If an operating system is to be used, an additional subset is required. Therefore, it is possible to look at the RISC-V as if it is actually a family of related ISAs. There are two base integer variants, RV32I and RV64I, each providing 32-bit or 64-bit width, respectively. The subset marked with E is designed for small microcontrollers, whereas the subset M enables multiplication and division, F is single-precision floating-pint, D for double-precision, etc. The most available RISC-V processors implement the RV64IMAFD flavor, which, due to its popularity, is marked with G (for general) [8]. Such is the rocket core, discussed in Section V of this paper.

RISC-V represents a perfect combination of simplicity and industrial application, and is, therefore, selected as the ISA to be studied in VLSI courses.

IV. Chisel

Chisel is a hardware construction language, first introduced in [9], built as a domain specific language (DSL) embedded in Scala, with the idea to support advanced hardware design by providing important concepts established in software engineering. These include object orientation, functional programming, parameterized types and type inference. Such powerful abstraction features enable high level of code reuse, thus improving efficiency in constructing new hardware systems. In this way, while not claiming that Chisel is *better* in general, we do point out that Chisel does provide new paradigms in hardware design, that can increase productivity.

At first glance, looking at simple combinatorial module such as a multiplexer shown in Listing 1, there are no conceptual differences to standard Verilog HDL approach, aside from the syntax [10]:

```
class Mux2 extends Module {
  val io = IO(new Bundle {
    val sel = Input(UInt(1.W))
    val in0 = Input(UInt(1.W))
    val in1 = Input(UInt(1.W))
    val out = Input(UInt(1.W))
  })
  io.out := (io.sel & io.in1) |
              (~io.sel & io.in0)
}
```

Listing 1. Multiplexer 2/1 in Chisel

While some strangeness is present, due to inheritance and `:=` operator, it is quite obvious we have a group of input/output ports packed in a `struct`-like piece of code (a *Bundle*) and some wiring in the last line. Looking at state elements, Chisel is still quite comparable to Verilog, as demonstrated by the 4-bit shift register in Listing 2 [10]:

```
class ShiftRegister extends Module {
  val io = IO(new Bundle {
    val in  = Input(UInt(1.W))
    val out = Output(UInt(1.W))
  })
  val r0 = RegNext(io.in)
  val r1 = RegNext(r0)
  val r2 = RegNext(r1)
  val r3 = RegNext(r2)
  io.out := r3
}
```

Listing 2. 4-bit Shift-register in Chisel

Of course, there are plenty of details to unpack here, such as the casting to `W` (bit width) type, wiring implicated from using `RegNext` specifically and so forth, but these are beyond the scope of this paper. For further instructions on Chisel, the reader is referred to [10], [11]. In [10], Chisel and dependencies are installed to a local machine and a set of examples, problems and solutions is provided, while the basics are explained through an online, github-wiki-based tutorial. In [11] Chisel is presented through an interactive tutorial based on a jupyter notebook with assignments and instructions intervened. There's an online version of the bootcamp, as well.

Chisel, rather, its toolchain, is capable of emitting Verilog for three different backends to generate either: (a) a software simulator that can be fed binaries, (b) a bitstream for an FPGA, or (c) netlist to be further used by automatized toolchain to yield GDS mask patterns ready for ASIC production. The actual advantages of Chisel over today's HDLs are shown in the next section, where processor cores used in our course are presented.

V. Processor Cores

Students get to learn about computer architecture and VLSI design by building one simple core, then applying that knowledge and expanding on it by investigating and modifying two more cores, each of these with several flavors.

A. Hack

Hack is the name of a computer and that computers processor - both developed within the now famous *Nand2Tetris* course, taught by Shimon Shocken and Noam Nisan. The course covers all abstraction levels starting from designing basic combinatorial modules, starting from NOT and XOR gates, collecting them into subsystems such as memories and aritmetic logic units (ALU), all the way through writing assembly code, operating system and even designing a complete java-like language, writing its compiler and then using the whole pyramid to design and play games. All tools required for the course are available free of charge at the course webpage [12], as well as the companion book [13], and the course itself, having been taught at several famous universities, is available at coursera.org.

We use the first half of Nand2Tetris, where students through six project assignments develop the chipset first, then the ALU, registers, RAM and program counter and learn about assembly code. Finally, they put all these together to obtain a fully functional Harvard architecture-based computer. In this way, a very sound foundation is created, making each candidate ready for tackling more complex problems such as memory-mapped input/output (MMIO) peripheries, pipeline, multicore processors, SoC design, etc.

B. Sodor

Sodor [14] is a collection of five simple and open-source cores written in Chisel, developed and published to be used at university level courses, as the basis for practical examples based on the theory presented in [3] and taught at UC Berkeley, *CS152/252A* courses [15].

After having developed Hack completely using the HDL provided at [12], and then taking a week to learn Chisel basics via the bootcamp [11], students learn about a real, RISC-V compatible processor. We focus on 1- and 5-stage implementations, as these provide enough to learn about performance metrics, most important benchmarks, measuring and comparing thus yielded numbers and introduce pipeline.

C. Rocket-chip

All the cores available in Sodor collection are written in ”plain” Chisel, meaning that no advanced principles such as core parameterization via diplomatic design patterns [16] are utilized to fully leverage advantages of Chisel over the standard HDLs.

Rocket-chip [17] is an open-source SoC generator also developed at the UC Berkeley, with the difference to Sodor cores that hardware it generates has actually been fabricated in silicon; therefore, while useful for research and teaching, these designs are applicable in industry, as well. As with other Chisel designs, its output is synthesizable Verilog and in this codebase the main advantages of Chisel are demonstrated.

It it important to differentiate between the *rocket* core as one of the cores that may be used within the SoC generated by the *Rocekt-chip* generator, whereas different cores may be used as well. Furthermore, the actual capabilities and performance may be fine tuned through the configuration parameters. This level of flexibility is yielded by the system's modular design, built upon HCL approach to hardware design.

To learn about specifics of the following example in detail, we refer the reader to Chipyard documentation [18], as such discussion is beyond the scope of this paper. Here, we list a few lines of code in order to demonstrate the agility of the SoC generator, Chisel and the approach in general. The default configuration of a Rocket-chip will yield a single 64-bit core accompanied by a floating point unit (FPU) and with level 1 cache. However, if it is required to generate core with smaller cache and without the FPU (lower the energy consumption, decrease area), by inspecting the `Configs.scala` file an appropriate configuration may be found:

```
class DefaultSmallConfig extends
    Config(new WithNSmallCores(1) ++
    new WithCoherentBusTopology ++
    new BaseConfig)
```

Listing 3. Configuration to gneerate singlecore system without FPU and with smaller cache

Next, just by running:

```
make CONFIG=DefaultSmallConfig
```

we obtain synthesizable Verilog and a software simulator, with characteristics defined by the configuration line above. Similarly, configurations with dual (or more) cores may be used, at 32- or 128-bit widths, etc.

We do not delve into diplomatic patterns that enable such level of modularity, but rather follow examples from UC Berkeley [15], where the students in their first computer architecture/VLSI course develop understanding of the mutual dependency between hardware design and its application, i.e. software that it executes. Hence the projects for this part of the course are related to the features of the pipeline and cache: we ask for performance measurement while a specific piece of code is executed, then we seek ways to optimize hardware design by, say, changing cache associativity, and, finally, the benchmarks obtained after the modifications are compared - number of misses and instruction per cycle (IPC), in this particular example.

VI. Automated Toolchain

Once the design is settled from the computer architecture point of view, i.e. either we've reached the requirements or the time is simply up, it is time to look for ways to materialize the idea in a real circuit. While FPGA is a valid destination for Chisel code, this is not the topic of VLSI courses in general, so we focus on the ASIC digital synthesis toolchain within this paper. That is a set of software tools used to transform the Verilog (emitted by Chisel in this case) netlist into a physical digital circuit. In semiconductor industry today, these tools are vast proprietary suites developed and licensed by large companies such as Cadence or Synopsis. These are expensive to the point that quite a limited number of long running IC manufacturers may obtain the licenses on a regular basis. There are university and start-up programs, but those are also

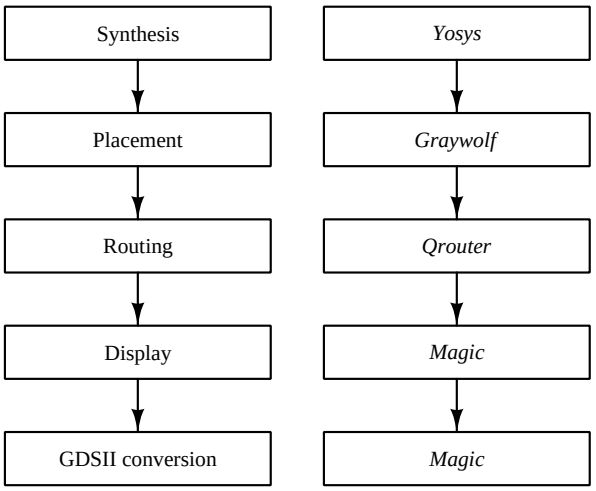


Fig. 1. Digital IC design open source toolchain: a) general approach, and b) tools applied during this course

far from free of charge for small universities or a team of two just starting out.

Qflow [19] is a complete, free of charge and open-source toolchain for synthesizing digital circuits starting from Verilog source and ending in physical layout for a specific target fabrication process. While the process is more detailed, we present a simplification describing only the major steps - rougly shown in Fig. 1. The first step in the automation process is to map the netlist onto a standard cells library, colloquially referred to as *PDK* (stemming from *project design kit*). PDKs are a topic of its own and a complex one, while at that, since these are also proprietary in general. Recently, there have been revolutionary development with SkyWater PDK [20], but not in time to be included in the course edition we are reporting on with this paper. In previous iterations of the course scalable CMOS PDK [21] was used. For now, we keep to the open-source PDK provided by the Oklahoma State University (OSU). This step is done by yosys [22]. Next, the design is to be placed and routed. In shortest terms, this when the standard cells are spread across the available area, while grouped in blocks and interconnected (routed). Graywolf [23] is the member of the qflow toolchain that does the placement, while routing is performed by qrouter [24]. Finally, for layout inspection, DRC and GDS generation Magic [25] is used. Qflow, nor the provided PDK are not capable of creating a microprocessor that may compete with current 3 GHz+ multicore server processors, but these tools will successfully handle simpler designs that may be found in SoC all over the market - such as SPI, for example. Firt live demonstration of a chip fabricated using nothing but qflow is presented in [26].

VII. Conclusion

While ISAs and processor cores are not directly a subset of a VLSI related course, we do live at a revolutionary moment in technology history - free and open source tools and PDKs are a reality, hence ASIC design and fabrication are within reach to individuals, start-ups and small universities with very limited funds. In future iterations of these courses, we plan to improve our automated design flow replacing qflow with openLane and including the open-source RAM compiler, openRAM. To overcome the steep learning curve of the new hardware design paradigm introduced with Chisel and rocket-chip, we are envisioning a Chisel GUI. Finally, we hope to fabricate students' designs through SkyWater 130 nm process, an open-source PDK.

References

[1] W. Shockley, "The theory of p-n junctions in semiconductors and p-n junction transistors," *Bell System Technical Journal. Institute of Electrical and Electronics Engineers (IEEE)*, vol. 28, no. 3, pp. 435–489, 1948.
[2] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, 1965.
[3] J. L. Hennessy and D. A. Patterson, *Computer architecture: a quantitative approach, 6th*. Morgan Kaufmann Publishers Inc., 2019.
[4] A. Pajkanovic, "On the application of free CAD software to electronic circuit curricula," in *Proc. of 3rd International Conference on Electrical, Electronic and Computing Engineering IcETRAN 2016*, Zlatibor, Serbia, 2016, pp. ELI1.3.1–4.
[5] A. Pajkanovic and Z. Ivanovic, "A report on recent development in application of free CAD software to IC curricula," in *Proc. of 5th IcETRAN 2018*, Palic, Serbia, 2018, pp. 847–851.
[6] A. Pajkanovic, "CMOS IC from schematic level to silicon within IC curricula using free CAD software," in *Proc. of INDEL 2020*, Banja Luka, Bosnia and Herzegovina, 2020.
[7] A. Waterman, "Design of the RISC-V instruction set architecture," Ph.D. dissertation, University of California at Berkeley, 2016. [Online]. Available: https://people.eecs.berkeley.edu/ krste/papers/EECS-2016-1.pdf
[8] A. Waterman, Y. Lee, D. A. Patterson, and K. Asanović, "The risc-v instruction set manual, vols. i-ii," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2016-118, May 2016.
[9] J. Bachrach, H. Vo, B. Richards, Y. Lee, A. Waterman, R. Avižienis, J. Wawrzynek, and K. Asanović, "Chisel: Constructing hardware in a scala embedded language," in *DAC2012*, San Francisco, USA, 2012.
[10] chisel-tutorial. [Online]. Available: https://github.com/ucb-bar/chisel-tutorial/wiki
[11] chisel-bootcamp. [Online]. Available: https://github.com/freechipsproject/chisel-bootcamp
[12] nand2tetris. [Online]. Available: nand2tetris.org
[13] N. Nisan and S. Shocken, *The Elemenets of Computing Systems*. MIT Press, 2008.
[14] riscv-sodor. [Online]. Available: https://github.com/ucb-bar/riscv-sodor
[15] Computer Architecture 152/252A, spring 2021. [Online]. Available: https://inst.eecs.berkeley.edu/ cs152/sp21/
[16] H. Cook, "Diplomatic design patterns: A TileLink case study," in *CARRV17*.
[17] K. Asanović, R. Avižienis, J. Bachrach, S. Beamer, D. Biancolin, C. Celio, H. Cook, P. Dabbelt, J. Hauser, A. Izraelevitz, S. Karandikar, B. Keller, D. Kim, J. Koenig, Y. Lee, E. Love, M. Maas, A. Magyar, H. Mao, M. Moreto, A. Ou, D. Patterson, B. Richards, C. Schmidt, S. Twigg, H. Vo, , and A. Waterman, "The rocket chip generator," EECS Department, University of California, Berkeley, MA, Tech. Rep. Technical Report UCB/EECS-2016-17, 2016.
[18] Chipyard. [Online]. Available: https://chipyard.readthedocs.io/
[19] Qflow. [Online]. Available: http://opencircuitdesign.com/qflow/
[20] SkyWater. [Online]. Available: https://www.skywatertechnology.com/
[21] MOSIS. [Online]. Available: https://www.mosis.com/
[22] C. Wolf, "Yosys open synthesis suite," http://www.clifford.at/yosys/.
[23] "Graywolf," https://github.com/rubund/graywolf.
[24] "Qrouter," http://opencircuitdesign.com/qrouter.
[25] Magic. [Online]. Available: http://opencircuitdesign.com/magic
[26] T. Edwards and M. Kassem, "The Raven Chip: First-time silicon success with qflow and efabless," in *Free Silicon Conference 2019, FSiC 2019*, Paris, France, 2019. [Online]. Available: https://wiki.f-si.org/index.php/FSiC2019

Two approaches to automatic configuration of RS-485 network

Nikola Cvetković, Pavle Milenković, Nenad Jovičić and Vladimir Rajović

***Abstract*—The purpose of this paper is to provide an overview of common approaches to automatic configuration of half-duplex RS-485 network, as well as to introduce two alternative methods of automatic slave address configuration in a network. Described mechanisms will be analyzed and compared in terms of hardware and software complexity, while taking into consideration system robustness and implementation feasibility.**

***Index Terms*—RS-485, automatic configuration, differential bus, communication network**

I. Introduction

The interface commonly known as RS-485 represents an electrical standard used in serial communication systems. It only defines electrical characteristics of drivers and receivers connected into a network [1], thus leaving the opportunity of using various standardized or user-defined data communication protocols. Some of the examples of frequently used protocols include Modbus, used in industrial settings, and BACnet, often applied in automation of buildings and other monitoring applications [2]. RS-485 is a balanced differential electrical bus, either full-duplex or half-duplex, supporting up to 32 unit loads, where each unit load represents an impedance of approximately 12 kΩ. Fig. 1 shows an example of a common balanced system. It consists of a driver D and a receiver R. A termination resistor R_T is used on the input of the receiver, to match the input impedance of the lines.

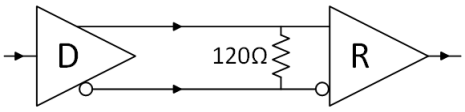


Fig. 1. A common balanced system

Since the lines of the bus are balanced, meaning they have equal impedances along their length and equal impedances with respect to ground, noise induced in the conductors and their electromagnetic radiation is minimized. Drivers' minimal differential output voltage has to be greater than 1.5 V across a load resistance of 54 Ω, while the minimal detectable differential input voltage of receivers is 200 mV. Described characteristics allow for a conventional transmission speed of up to 10 Mb/s and the maximum range of 1200 meters. When designing a system, an empirical relation for transmission speed and bus length ratio has to be taken into consideration [3]. Although the standard recommends 10 Mb/s, today's fast interface circuits are optimized for data rates of up to 50 Mb/s [4].

All the approaches for the automatic network configuration that will be analysed in this paper use a master/slave model of communication, in which only the master can initiate communication with the slave. The master can either send a broadcast address in order to address all the nodes in the network, or it can address individual slave by sending a specific address.

RS-485 standard supports various network topologies, the simplest of which is point-to-point connection (Fig. 2).

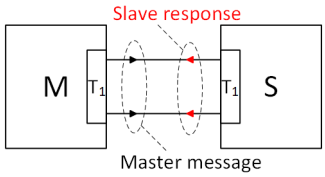


Fig. 2. Point-to-point topology

Adding additional network nodes to the topology mentioned above requires insertion of junction boxes. This modification, referred to as Backbone with Stubs, consists of a main differential bus called a backbone, branching off to the slaves with stubs (Fig. 3). The described topology is one of the two most commonly used methods for connection of the nodes.

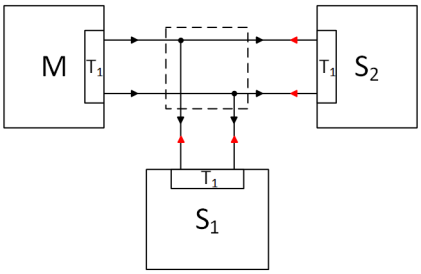


Fig. 3. Backbone with stubs topology

The second most frequently used connection method is daisy-chain topology (Fig. 4), which requires each two consec-

Nikola Cvetković is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: cn203094m@student.etf.bg.ac.rs).

Pavle Mileknović is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: mp203016mm@student.etf.bg.ac.rs).

Nenad Jovičić is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: nenad@etf.bg.ac.rs).

Vladimir Rajović is with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: rajo@etf.bg.ac.rs).

utive nodes to be directly connected to each other via dedicated connection ports.

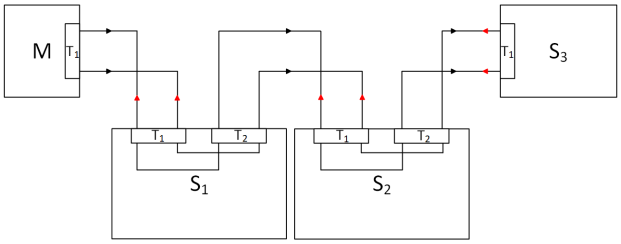


Fig. 4. Daisy-chain topology

When designing a network, special care has to be taken in order to provide a unique address for each node. In conventional address assigning methods, problems occur in manual work cost, hardware and software complexity, and inconvenience in network modification, including faulty node removal and network expansion. Without loss of generality, the main focus of the analysis will be on half-duplex RS-485 bus. Additionally, communication is performed in a traditional master-slave manner and information about relative physical position of each node has to be available at all times.

II. Common Slave Addressing Approaches

In practice, there are three widely used methods for node address configuration, which will be covered in more detail in the following subsections. These include different hardware address assignments, firmware hard coding of an address and software address assignment upon addition of a new node, one by one. The common characteristic of all these methods is that they are all in need of node configuration prior to address assignment.

A. Hardware Address Assignment

The simplest way to assign an address to a slave from a software perspective is to provide the hardware for address configuration. This can be achieved by using dual in-line package (DIP) switches, rotary switches, on-board jumpers, or by directly connecting every slave's address pin to either high or low logic levels. A typical configuration using a DIP switch is represented by Fig. 5.

In spite of being easy to implement, relatively cheap to design and in some applications easily reconfigurable, this method has considerable setbacks. This system is prone to human error, since all switches/jumpers rely on manual configuration, increasing the possibility of multiple devices having the same address. Additionally, in an environment where network nodes are not easily accessible, modification of the addresses might pose an issue. In some cases, major flaw of described approach can be the price of manual work required for network initialization and modification. Despite having the mentioned limitations, this approach is frequently found in commercial applications [5], [6].

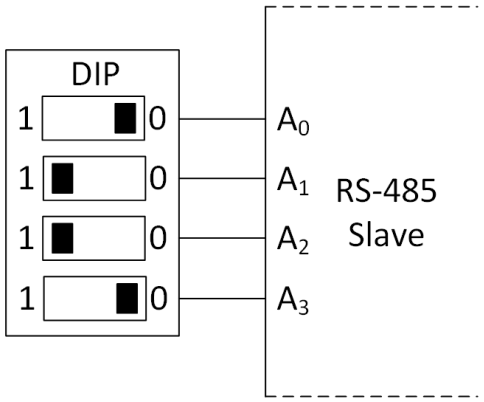


Fig. 5. Address configuration using a DIP switch

B. Firmware Hard Coding of an Address

Another method is the hard coding of an address inside the slave firmware. This approach results in a simpler and cheaper design, as it does not require any additional hardware. Since it does not require a direct human contact during deployment in order to configure the slave, the system is less prone to human error than the previous approach. However, the main weakness of this method lies in the effort to write and maintain different versions of firmware for each node in the network. In terms of network extension and modification, nothing is improved compared to the hardware assignment method.

C. One by One Network Expansion

If hardware modification of the nodes is not possible, different firmware versioning can be avoided by gradual expansion of the network, one node at the time. A common algorithm of RS-485 bus initialization is represented by Fig. 6.

At the beginning of the initialization of the network, every slave is configured to have the same default address, e.g. address 0x00, while the initial topology resembles point-to-point communication (Fig. 2). Master sends a broadcast message, addressing all slaves with the default address. Since the nodes are being added to the network one by one, at any given moment, there will be no more than one device with the default address. The node with the default address sends an echo message, requesting the new address from the master. Finally, the master sends a message containing the new address of the slave and waits for the confirmation of the message, checking if the slave is configured properly. Master can be configured to send a broadcast message periodically, or by user command upon connection of a new node. Some implementation of a waiting/timeout loop should be considered, since various errors in a network can result in improper signal reception, either on master or slave side of the connection. Another advantage of this method is the simplicity of adding new nodes into a network - no additional work to adjust the software or configure the switches is required, apart from physically connecting the nodes one after another. Although relatively simple in terms of hardware and software complexity from the

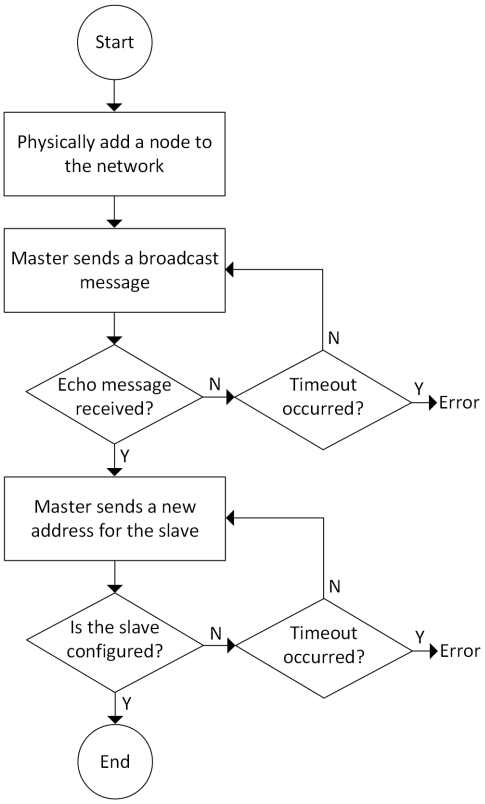


Fig. 6. One by one initialization algorithm

slave side, this approach requires a long initialization process, which results in a costly manual work.

III. Proposed Solutions

In the previous sections, the internal structure of network nodes was implemented using a single RS-485 transceiver. A typical example of an RS-485 transceiver can be observed in the Fig. 7.

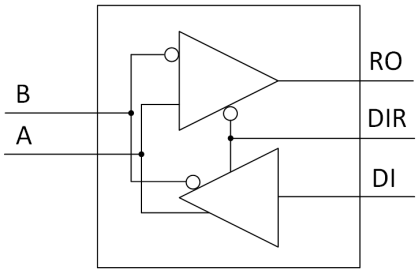


Fig. 7. Typical RS-485 transceiver

By designing the network nodes using additional RS-485 transceiver, new methods for automatic address assignment can be formulated.

A. Utilizing Network Topology for Slave Addressing (Domino Method)

Proposed structure of a network node is given in the Fig. 8. This solution offers a flow of information through the slave, by

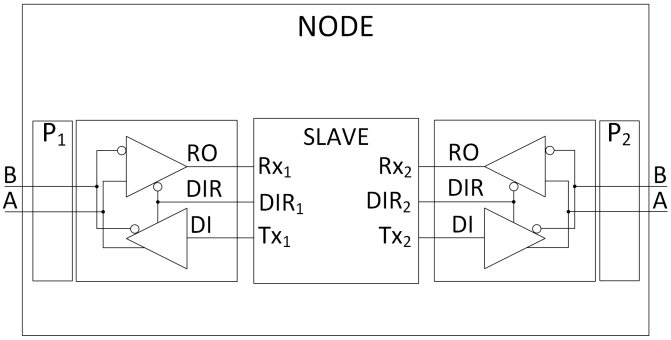


Fig. 8. Node structure for the Domino method

making use of two serial ports. The line topology obtained by connecting multiple slaves with previously presented structure is shown in Fig. 9. Nodes are denoted by N_i , $i = \overline{0, n-1}$, while each node has two ports, P_1 and P_2.

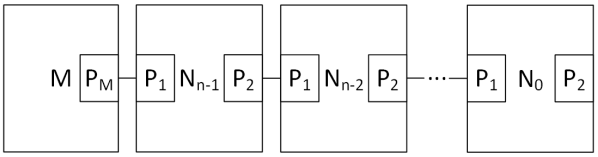


Fig. 9. A proposed network topology for the Domino method

The basic principle of the proposed solution is that there is a predetermined address that triggers the responses from every slave, similarly to the broadcast address in any multiple-access communication network. In a given example, the value of predetermined address is 0x00. The system initialization phase starts with determining the number of nodes in the network. Depending on the expected number of nodes, either incremental or decremental approach can be used. Perhaps more intuitive is the incremental approach, used for small-scale networks. The algorithm starts by master sending a test message over P_M port, containing the predetermined address, which stores the information about nodes counted thus far, in this case 0x00. Since the predetermined address of the slave matches the address field in the message, the first slave in the line responds by echoing the message back to the master, over port P_1. The master detects an echo, and calculates the total number of detected devices by incrementing an address field by one. Now the master repeats sending a test message, this time with the address field 0x01. The first node in the line will not respond to the test message, since its predetermined address is 0x00. Because there is no match, the node will decrement only the address field, and pass the message to the next node in the line through the P_2 port. After passing the message, the node will change the direction of information flow, by toggling the values of DIR_1 and DIR_2 pins, thus configuring the P_1 port as a driver and P_2 port as receiver. The second node receives a message on P_1 port, now with the address 0x00, and responds to it by sending the echo, using the same port and address 0x00. The preceding node

receives the echo on its P_2 port, increments the address field to 0x01, and passes the message to the master using P_1 port. Upon receiving an echo, the master increments the address field, updating the number of counted nodes to 0x02. If the total number of nodes in the line is given by n, the described process repeats for a total of n times. Successful addressing of the n-th node is presented in Fig. 10.

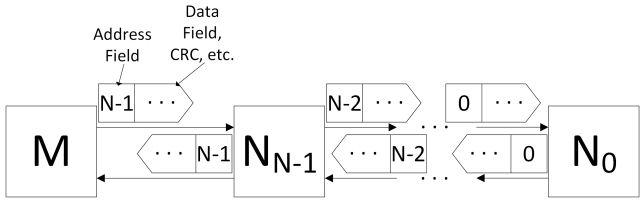


Fig. 10. Domino method addressing

Lastly, the master sends a test message once again, this time reaching the last node in the line, with the address of 0x01. The last node receives the message with this address field, decrements it and attempts to send it using its P_2 port to the next node. Since there are no nodes left, there will also be no echo message back to the master. The master waits for the echo for a certain amount of time, after which a timeout event occurs, signaling the master that there are no nodes left.

Once the total number of devices is determined, it can be used for communicating with each device. This communication protocol is based on master sending a message with the address field corresponding to the relative position of the node in the line.

One of the main issues with the described approach lies in addition of new nodes into a network. When the node is added, the initialization routine has to be repeated. This may be performed by the request from the user, or by periodical repetition of the algorithm. The suggested approach would be to manually start the routine, as it is less time-consuming.

The other downside of the proposed solution is the handling of a malfunctioned node or electrically corrupted bus - the line will be interrupted as the signal can not propagate through subsequent nodes.

Domino method occupies two serial ports of the node's micro-controller, which can be a limiting factor in system design, potentially increasing the overall cost of the system. Additionally, this solution introduces a significant computational delay, as the information is processed by each micro-controller it passes through. The total delay added into a single communication cycle, consisting of a single master request and slave response when addressing i-th slave in the line, is given by:

$$T_{DELAY_I} = 2 \cdot i \cdot T_{MCU}, \tag{1}$$

where T_{MCU} is the time required for a message to be processed and/or modified. This value has to be multiplied by the number of nodes that information passes through, and, since the message propagates back to the master, the whole value is multiplied by two. When evaluating T_{MCU}, it is important to include not only the time required for increment/decrement operations, but also time required for receiving and sending a modified message using a serial port. Assuming there are n nodes in the line, the maximum delay introduced by the Domino method is given by the expression:

$$T_{MAX} = 2 \cdot n \cdot T_{MCU}. \tag{2}$$

B. Staged Address Assignment with Bus Bridging (Pontoon Method)

As it can be observed from the previously described method, the core of the node constantly processes received information, introducing delay. This potential overload of the slave's micro-controller can be avoided by bridging the RS-485 bus inside a node using a discrete multiplexer (Fig. 11).

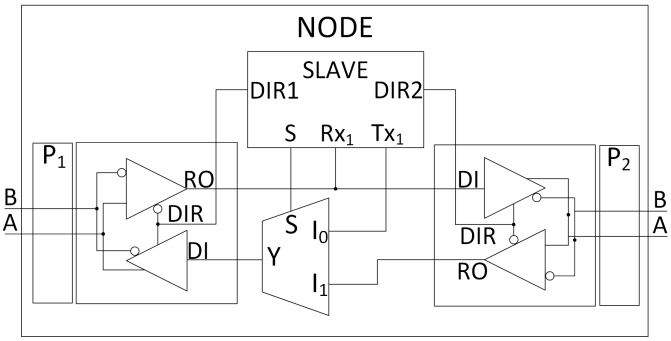


Fig. 11. Node structure for the Pontoon method

On a system level, the topology is identical to the one described in Fig. 9. Contrary to the previous approach, addresses of the nodes can be assigned by the master and each address has to be stored inside of the corresponding node. Additionally, only one serial port is used, which allows for more flexibility when designing a system. Initially, every node of the network is configured in a way that both of its ports receive the information, and each node's address is set to a predetermined value, e.g. 0x00. The master starts the process of address assignment by sending a message, addressing a device with the predetermined address. The first node receives the message, recognizing its address. Since its P_2 port is in receive mode, the propagation of the message stops. After recognizing its address, the first node sends an echo to the master by selecting I_0 multiplexer input and changing the direction of P_1 port, asking for a new address. The master sends another message for slave address configuration, where the address field can be arbitrarily determined by the user. Upon receiving a new address, the first node sends back a conformation message to the master and changes the direction of P_2 port in order to allow for the next address configuration message to pass through it. The master acknowledges that the first node is successfully configured and sends the message with the address 0x00 once again. The first node receives this message on P_1 port, and since its P_2 port is configured as a driver, the message can pass through the node, without unnecessary computing delay. As the information is passed,

both ports of the first node have to change directions and I_1 multiplexer input has to be selected, to allow for opposite data flow. The second node in the line receives the message on its P_1 port. Because its address is 0x00 and its P_2 port is blocking the information flow, it echoes the message back to the master, requesting a new address. The first node receives an echo on its P_2 port, passes it through P_1 port and changes the direction of the ports once again. The process continues until the last node is reached, and once again, the master has to implement a waiting mechanism, when the last node tries to pass the configuration message. If a certain slave recognizes its address after the initialization is done, it has to configure its P_1 port as a driver, selecting the I_0 multiplexer input.

This time, adding new nodes to the network does not require repetition of the whole initialization process, but only a single passing of the configuration message. Again, the line will be interrupted in a situation where the node is malfunctioning or the bus is somehow corrupted, either by open or short circuit.

Contrary to the Domino method, this approach introduces additional delay during network configuration. For any node in a network, configurational delay is:

$$T_{CNFG_{SINGLE}} = 2 \cdot T_{MCU}, \tag{3}$$

where, again, T_{MCU} is the time required for a message to be received, processed and transmitted. It takes single T_{MCU} to process a message containing the predetermined address, and another T_{MCU} to process a message containing a new node address. For a network containing n nodes, the total configurational delay is given by:

$$T_{CNFG_{TOTAL}} = 2 \cdot n \cdot T_{MCU}. \tag{4}$$

After the network is successfully configured, the duration of a communication cycle, consisting of a message transmission from the master and a response from any slave is:

$$T_{COMM} = T_{MCU}, \tag{5}$$

since the message passes through preceding nodes directly, without being processed by their micro-controllers.

IV. Conclusion

From the hardware perspective, both automatic configuration methods presented require an additional RS-485 transceiver, which increases design complexity and overall price of each node. However, these approaches completely eliminate the need for manual work, except when adding or removing a node. By using the described algorithms for node communication, the whole network can be effortlessly reconfigured through master. This means that the nodes do not have to be easily reachable, which avoids implementing galvanic isolation of RS-485 transceivers, except when common mode voltage reduction is required. Further steps can be taken in order to completely relinquish the use of galvanic isolation [7].

The Domino approach that utilizes the network topology for slave addressing does not require individual storing of a slave's address. Therefore, it can be used in a system where there are multiple slaves of the same type, often performing a similar task. The example of a system that can benefit from this solution can be temperature monitoring networks, or street light management systems, where each node is functionally the same.

Additionally, this solution is less complex hardware-wise than the Pontoon method, but occupies an additional serial port and introduces computational delay, as the micro-controller processes address fields of the message.

Pontoon approach allows each node to have a unique address, independently of its position in a communication line. This can be useful in a system where certain types of sensors or devices have to share a specific address range, for example in various industrial or home automation systems.

Since there is no need for master to periodically poll the network in order to detect new devices, the whole method of network reconfiguration, adding and removing the nodes, is simplified compared to the first proposed method. This approach introduces virtually no additional communication delay, apart from the propagation of a signal through multiplexer. Pontoon approach does introduce a multiplexer in the node's design, but offers a possibility of compromise price-wise, since it occupies a single serial port, allowing for a simpler micro-controller. The usage of a discreet multiplexer can be avoided by directly connecting RO line from P_2 port to the DI line from P_1 port and Tx_1 line, and by keeping Tx_1 line in a high-impedance state whenever the message is expected to be received from P_2 port.

The Domino method does not require node address configuration, meaning that there is no time delay introduced during network initialization, compared to the Pontoon method that introduces a delay of $2nT_{MCU}$ during this phase, for a network consisting of n nodes. On the other hand, during a communication cycle, the Domino method requires $2iT_{MCU}$, for i-th node in the line, while the Pontoon approach requires only T_{MCU} for the same operation, regardless of the node position.

Both of the proposed solutions can be used in systems that are inherently linearly connected. In addition to the examples mentioned before, those systems include metallurgical furnaces, speed and traffic detection, environmental, and other kinds of roadway sensors, as well as various sewer sensors. Furthermore, the described methods can relatively easily be generalized in order to be applied to other communication systems, using different electrical standards and protocols. Without change in principal node structure, approaches can be easily adjusted to fit other serial standards. These methods of address assignments can be adapted for different communication mediums, including wireless and optical communication.

The future work should cover defining methodology for generalization of the presented address assignment approaches, as well as finding ways to improve system robustness and failure recovery capabilities.

ACKNOWLEDGMENTS

This work was supported by the Ministry of Education, Science and Technological Development, Republic of Serbia, under Projects TR32043 and TR32039.

REFERENCES

[1] *Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems*, ANSI/TIA/EIA-485-A-1998, TIA/EIA, 1998.

[2] J. Axelson, *Serial Port Complete: COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems*, 2nd ed. Madison, WI, USA: Lakeview Research LLC, 2007.

[3] T. Kugelstadt, "Application Report - The RS-485 Design Guide," TI, Dallas, TX, USA, 2021.

[4] Texas Instruments, "SN65HVD7x 3.3-V Full-Duplex RS-485 Transceivers with ±12-kV IEC ESD," TI, Dallas, TX, USA, 2019.

[5] REFLEX Thinking Solutions, "Busmodul - Profibus DP, Operating Manual," Reflex Winkelmann GmbH, Ahlen, Germany, 2014.

[6] C. Boiano, P. Guazzoni, L. Zetta, C. Guazzoni, and A. Pagano, "A 16-channel programmable antialiasing amplifier", IEEE Nuclear Science Symposuim & Medical Imaging Conference, Knoxville, TN, USA, 30 Oct. - 6 Nov. 2010.

[7] N. S. Jovičić and V. M. Rajović, "A Floating Linear Voltage Regulator for Powering Large-Scale Differential Communication Networks," *IEEE Access*, vol. 6, pp. 24669-24679, May, 2018.