Supervision of the operation of digital circuits by Embedded Microcontroller

György Györök, András Dávid, Nikoletta Tolner, Bertalan Beszédes, Dániel Cseh
Alba Regia Technical Faculty, Óbuda University
H-8000 Hungary Székesfehérvár, Budai str. 45, Hungary
{gyorok.gyorgy, david.andras, tolner.nikoletta, beszedes.bertalan}@amk.uni-obuda.hu; cseh.daniel@gmail.com

Abstract: There are many electronic circuitry applications that can be used to determine circuit behaviour and operation only with the probability of time. For certain applications (medical electronics, safety engineering, operational-critical industrial applications, transport ... etc.) it is important that the quality of the circuit can be determined.
In our present article, we describe a method in which we compared the digitally produced model of a circuit, and the circuit function can be determined in relation to that-time invariant model. For this process, an embedded microcontroller environment is formed which has one of the embodied test states for combinational and sequential digital circuits.

I. PRELIMINARIES

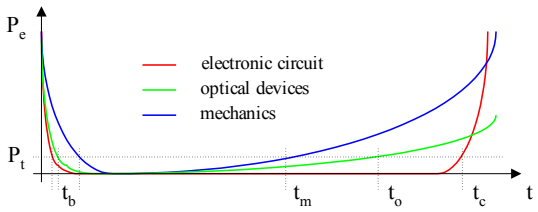

Fig. 1. Failure probability-time functions of three different technology devices.

For technical applications is often used, the bath-curve functions which is characterized failure-probability and lifetime correlation (Fig. 1.), where P_e is the probability of failure, while the horizontal axis is time (t) dimensional. We can observe three probability at three different technologies; t_m, t_o, t_c are the life expectancy typical of mechanic, optical and electrical devices, and t_b is the threshold time of burning-in failures. Above these, the probability of failure reaches the critical (P_t) value [13]. For high-reliability circuits and equipment, it is not necessarily enough to appeal to statistical probabilities.

II. STATECONTROLL OF ELECTRONIC CIRCUIT BY EMBEDDED MICROCONTROLLER

For this, we recommend the layout of Fig. 2. Here, an embedded microcontroller interrogates the voltage of the relevant points in the supervised circuit [11].

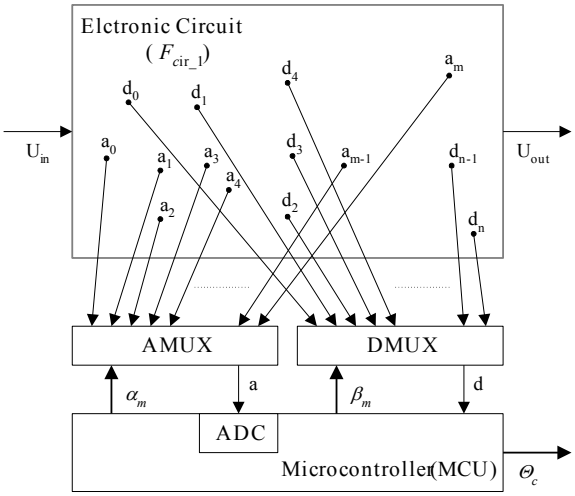

Fig. 2. Embedded microcontroller at the nodal voltage-pooling methods.

Traditional hybrid electronic circuits (Fcir-1) consist of between input (U_{in}) and output (U_{out}) a lot of electronic parts in well designed connection, a circuit network. This electronic circuit has got any significant analog (a_0, a_1, $a_{2...}a_m$) and digital (d_0, d_1, $a_{2...}d_n$) nodes [9] [10]. Microcontroller on α_m surface addressing a multiplexer (*AMUX*), and a selected voltage of a node appears on output (*a*) connected to input of analog-digital converter (*ADC*). The last one is an inner periphery of the microcontroller. Case of digital signals is used a digital multiplexer (*DMUX*), and its select surface is β_m, and output of DMUX is *d*. Important to note that multiplexing of digital signals can solve by pooling of an digital PORT of the microcontroller [14] [15].
If the voltage of the polled nodes deviates significantly (eralier gived) from the desired value or the digital signal levels are inadequate, the microcontroller generates an error signal (Θ_c). The Θ_c can either contain the serial number of the defective node and the nature of the error [1].
The current values of the nodes can be determined in three ways; or by electronic circuit simulation or manual-, or automatic measurement of a reference circuit. The values obtained, together with the tolerance value, can be compressed in a table [9].

III. STATECONTROLL BY INPUT MANIPULATION

The procedure of Fig. 2 can also be extended to input manipulation [12].
For this, it is very important to determine the moment when the test can be performed without disturbing the operation of the circuit (the whole system), now with excited control. Figure 3 shows the proposed arrangement. Here, the input signal of the circuit (U'_{in}) is taken from the normal environment (U_{in}) or is produced by a microcontroller (U_c) for testing purposes. To do this, we use an analog multiplexer (*AMUX*$_i$) that is addressed through the σ surface [3] [4].
Analog signals can be generated by a PWM method to obtain a microcontroller while digital signals can be generated with a bit-type output [2].

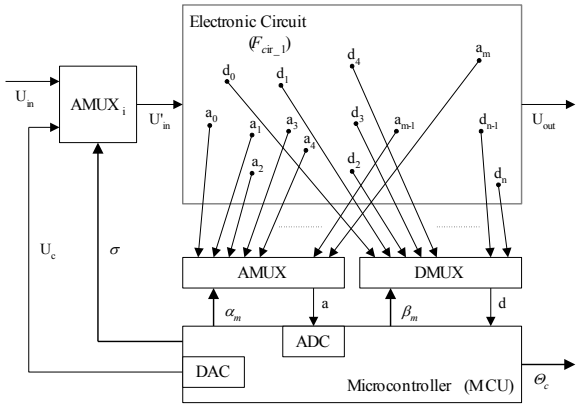

Fig. 3. The input manipulation by a hybrid circuit.

IV. VERIFICATION OF THE OPERATION OF THE PROCESS USING THE SYSTEM IDENTIFICATION METHOD

Fig. 4 shows the test procedure for most errors in the use of possible electronic circuits. Thus, we can test the system's functioning with the stimulated errors.
To test the process itself, we make a reference circuit whose operation is trivial. Thus, we use a digital circuitry where commonly used circuit solutions occur, including combinatorial and sequential networks, and tristate circuits [6][7][8].

For combination networks, it is appropriate to produce the correct total input excitation. For sequential networks, after a well-defined reset, the clock is generated by the testing.ndi networks, after a well-defined reset, the clock signals are generated and the testing can be done [5].

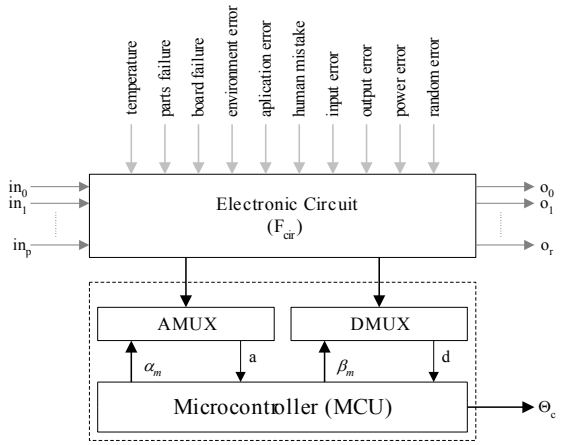

Fig. 4. The proposed circuit supervisor arrangenment with system identification approach.

V. THE TESTED CIRCUITS

The digital network contains the following combinational and sequential circuits:

Combinational networks:

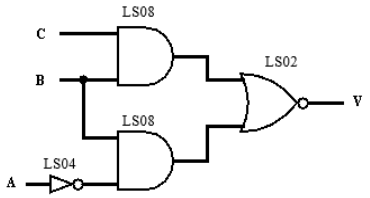

Fig. 5. Simple combinational network.

In the case of the simple combinational logic network the output is 0 if both the inputs B and C are 1's or B is 1 and A is 0.

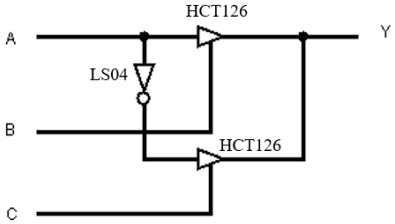

Fig. 6. Tri-state output network

In the case of the tri-state network the output is logic level 1 if the inputs A and B are 1's and the C is 0 or if the input C is 1 while A and B are 0. Applying 1's at the same time to B and C inputs must be avoided, because it can ruin the circuit.

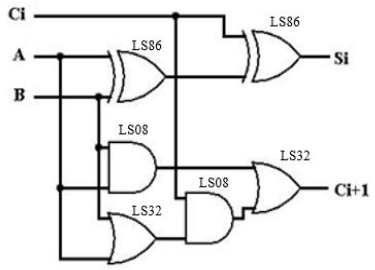

Fig. 7. Full adder

In the case of the full adder the output Si is logic level 1 if one of the inputs A, B, Ci is 1 or all of them are 1's. The output Ci+1 is 1 if at least two of the inputs are 1's.

Sequential circuits:

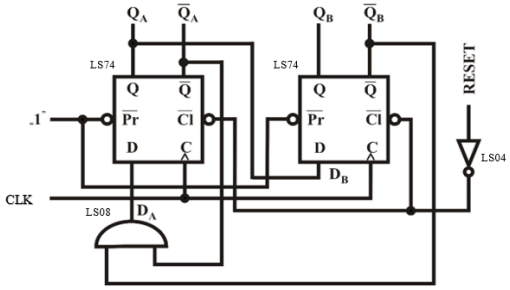

Fig. 8. Sequential circuit using flip-flops

With Reset flip-flop circuit the counter will step into 0. The count sequence is 1, 2 and then 0 again.

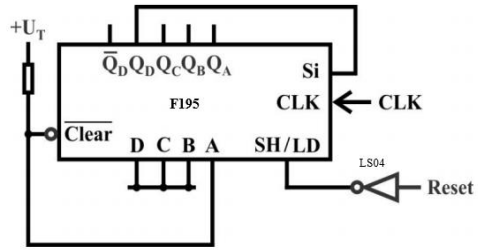

Fig. 9. Ring counter using shift register

With Reset the ring counter will step into 1. The count sequence is 1,2,4,8 and then 1 again.

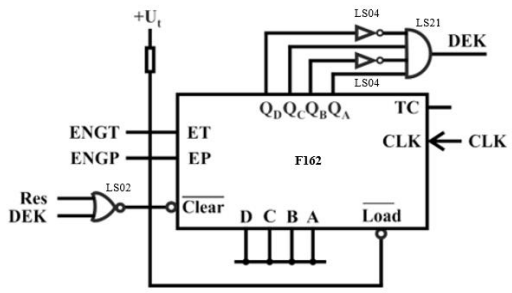

Fig. 10. Network with synchronous counter

With Reset the network will step into 0. The count sequence is 0, 1, 2, 3, 4, 5 and then 0 again. By applying logic level low to the input ET the network won't step into the next state, it will hold its present state.

VI. TESTING THE CIRCUITS

We checked the operation of the combinational and sequential networks discussed in part II by simulation and measurement. We will describe our experiences We checked the operation of the combinational and sequential networks discussed in part V by simulation about the simulation and measurement in another article [15].

VII. THE TESTED CIRCUITS ON BREADBOARD

To test the measurement control program we built the above described circuits on breadboard. We used the same type of IC's which can be seen on the schematics (Fig. 5, 6, 7, 8, 9, 10).

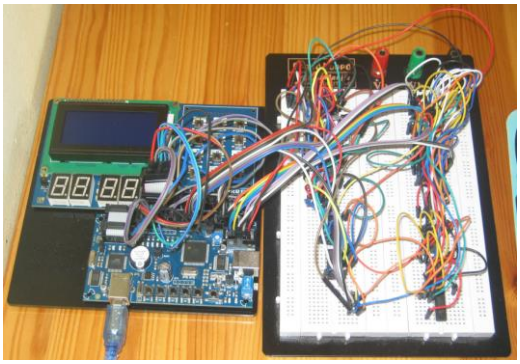

Fig. 11. The test circuit on breadboard and the development environment

VIII. DEVELOPING THE MICRO-CONTROLLER BASED TEST ENVIRONMENT

To prepare the test environment with microcontroller we used the AVR development board "T-Bird 2" with T-Bird expansion board made by BioDigit Kft. This development tool contains ATMEL AVR – Atmega128 microcontroller. On the development board we can find 5 push-buttons, which can be used to navigate in the menu. The results are shown on the LCD screen which is placed on the expansion board. The control program was written in C language.

The microcontroller board operates at 5V DC, which is connected from PC (via USB) or from external power supply. All of the IC's on the board operate at 5V DC,

which comes from the microcontroller board. The microcontroller has 128KB flash program memory, 4KB EEPROM and 4 KB SRAM data memory. On the expansion board we can find a HD44780 compatible LCD screen with 4x16 display format. In the measurement control program the clock and the reset signals which are needed to test sequential networks are generated by software. The clock has a time period of 1ms; the reset is logic high for 1 clock cycle [16][17].

Fig. 12. AVR development board „T-Bird 2”

IX. THE MEASUREMENT CONTROL PROGRAM

The already written measurement control program implements the following functions:

- quick test
- show the results
- step by step test
- manual measurement

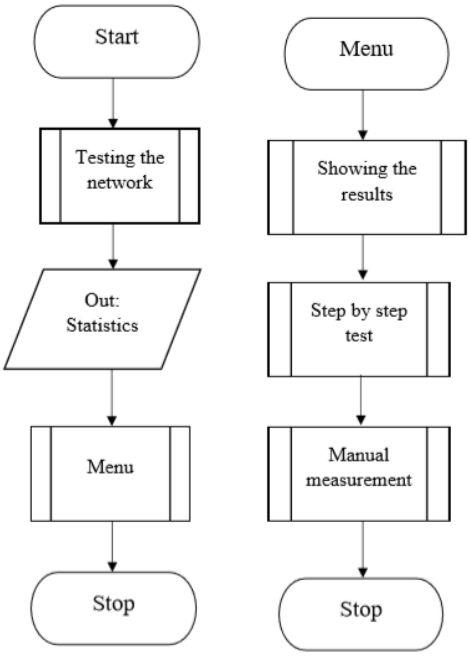

Fig. 13. The global flowchart of the measurement control program

a) Quick test

In the quick test the program measures the logic level of the output nodes. After that it compares the results with the values in the data (reference) table and prints out the quantity of errors and the number of the faulty nodes.

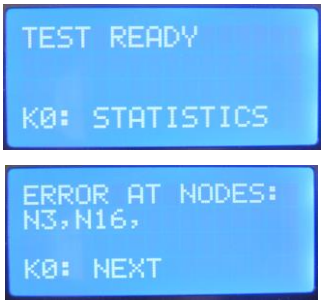

Fig. 14. An example for a running quick test

b) Showing the results

In this mode we can print out particular results of the measurement in details. In the case of a combinational network the input combinations (Fig. 15 column CBA), in the case of a sequential circuit the number of clock cycle and the reference value (Fig. 15 column R) and the actually measured value (Fig. 15 column M) appear on the LCD screen. In the picture below we can see the results of the test of a combinational network.

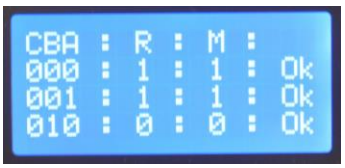

Fig. 15. An example for printing out the results

c) Step by step test

In this mode we can test the network step by step. The results are shown on the LCD screen, the signals appear on the nodes and pins, so we can measure them by multimeter or oscilloscope. By pressing the K0 button we can step further to the next combination and clock cycle. With this mode we can detect the cause of failure more accurately. On Fig. 16 we can see a detail of the test results of the synchronous counter (Fig. 10). “O” means the number of clock cycles after RESET, R is the reference value, M is the measured value. The output order is: $Q_DQ_CQ_BQ_ADEK$.

Fig. 16. Example for step by step test (tested with the network on Fig. 10)

d) Manual measurement

In this mode the combinations and the clock cycles appear on the appropriate inputs of the digital network with the timing set by the program. We can measure output signals on the appropriate nodes and pins with oscilloscope. In our previous article [15] we performed the measurement in this mode.

X. THE MEASUREMENT CONTROL PROGRAM IN DETAILS

a) Assigning the ports to the signals in case of combinational network

In case of the combinational network the inputs A and B of the schematics 5, 6, 7 are connected, so they are controlled together. The inputs C and Ci of the schematics 5 and 7 are also connected, in the case of schematic 6 the input C is an individual input, it is not connected to the input C of the schematics 5 and 7. The reason for this is that there are two input combinations of the tri-state output network of Fig. 6 in which case both tri-state gates would be enabled. Of course this combination must be avoided.

b) Assigning the ports to the signals in case of sequential network

In case of sequential network the inputs CLK and RES are common by all three schematics (Fig. 8, 9, 10), input ET is individual input.

c) Quick test

The essence of the quick test is that the program measures the logic level of the given nodes without human intervention and compares the measured value to the values of a reference data table. The program handles the combinational and the sequential networks separately. The data tables are 2x8 int type matrixes. The order of the bits of the matrixes is not the same as the pinout of the connectors. Row 0 of the matrixes contains the reference values (in case of correct operation these equal with the logic levels to be measured), row 1 contains the actually measured logic levels. In case of combinational network the indexes of the columns are the combination itself (C is the most significant bit, A is the least significant bit), in case of sequential network it is the order of the clock cycles.
We tested the tri-state output network (Fig. 6.) when connected to digital input ports, and we also tested it when connected to analogue input ports. We used different reference data table when we connected the tri-state output network to analogue input. This data table is a 2x6 char type matrix.

d) The algorithm of the quick test:

We the algorithm of the quick test into two main parts. In the first part we test the combinational network, in the second part the sequential network.

Testing the combinational network:
The testing process of the combinational network was taken further apart into two sub pieces, because to test the tri-state output network an analogue measurement is necessary as well.
The essence of the test is that we put out the 3-bit input combination to the appropriate pins cyclically. After the right timing we measure and read the logic level of the output pins, and then we save it into the first row of the reference data table.

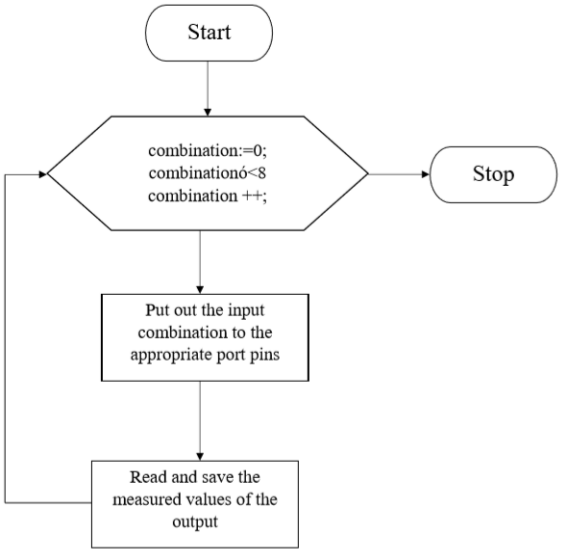

Fig. 17. The algorithm for the test of the combinational network

We tested the tri-state output network also in the way that we simply connected the output to a port that was defined as a digital input. When the output should have been in high impedance state according to the truth table, we measured 4.5V on the output, which is logic level high. The reason for this is that if a port is set as input, then we can attach an inner pull up resistor to the given port. In our case the inner pull up resistor was enabled, which pulled up the output to logic high. When we disabled the pull up resistor on the given port, we measured 1.2V on the output in high impedance state. When we tested the tri-state output network (Fig. 6), we connected the output (Y) through a voltage divider to an analogue input, and we measured the voltage of the output Y. Based on the specified voltage intervals we were able to decide if the output is logic 0, 1, or Z (high impedance state).

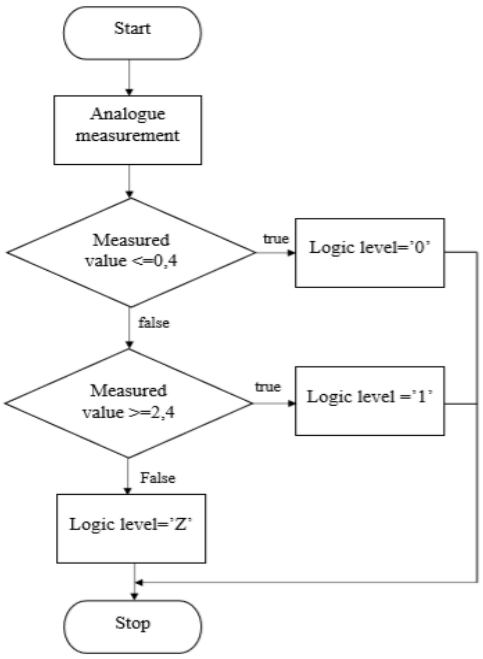

Fig. 18. The principle of the testing of the tri-state output network

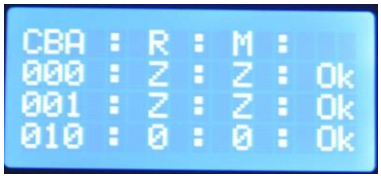

Fig. 19. Result of the testing of the tri-state output network (with the network on Fig. 6)

Testing the sequential network:

The essence of the test is that after the RESET we cyclically query and read the logic level of the outputs after the right timing during 8 clock cycles and we save them into the first row of the data table.

e) Making statistics:

When making statistics we compare the bits in the rows 0 and 1 of the reference data table one by one (row 0: reference value, row 1: actually measured value), so we can decide if the operation is correct or faulty. Every output has got a number (beginning from 0). If the measured and reference values of the given output are different from each other, we set the given bit (=1) of the statistics vector, otherwise we reset it (=0). 1 means error, 0 is the sign of the correct operation. After the program is finished the statistics is printed out. The program prints out the quantity of errors and the number of the faulty nodes (Fig. 14).

f) Printing out the results and the algorithm of the step by step test

In these two menus we can choose combinational or sequential network and after that we can choose which particular network we want to test.

When testing combinational and sequential networks this two menu offer separate algorithms for each type. In both cases we solved the printing out of the results and the step by step test with merged, parameterized functions. One parameter shows if only printing out or step by step test is needed (the difference between the two is that in case of printing out we simply read out the data from the first row of the data table, in case of step by step test we need to perform a new measurement to get the measured value). The other parameter shows which network we talk about (for example the network on Fig. 10).

```
void sorrendi_halozat_teszt(void)
{
    szinkron_reset();
    PORTC |= 0b00000100; //ET = 1

    for(int i=0; i<8; i++)
    {
        //scanning output signals
        //flip-flop
        int port_be_flip_flop = 0x00;
        port_be_flip_flop = PINC;
        port_be_flip_flop >>= 4;
        //register
        int port_be_regiszter = 0x00;
        port_be_regiszter = PINB;
        port_be_regiszter &= 0x0F;
        //counter
        int port_be_szamlalo = 0x00;
        port_be_szamlalo = PIND;
        port_be_szamlalo &= 0x0F;
        int dek_be = 0x00;
        dek_be = PINC;   //DEK
        dek_be &= 0b00001000;
        port_be_szamlalo <<= 1; dek_be >>= 3;
        port_be_szamlalo |= dek_be;

        //generating data
        int mert_adat = 0x0000;
        mert_adat |= port_be_flip_flop;
        port_be_regiszter <<= 4;
        mert_adat |= port_be_regiszter;
        port_be_szamlalo <<= 8;
        mert_adat |= port_be_szamlalo;
        adattabla_sorrendi_halozat[1][i] = mert_adat;

        orajel();
    }
}
```

Fig. 20. Detail of the program – testing of the sequential network

```
void orajel(void)
{
    PORTC |= 0b00000010;  //CLK = 1
    _delay_ms(1);
    PORTC &= 0b11111101;  //CLK = 0
    _delay_ms(1);
}

void szinkron_reset(void)
{
    PORTC |= 0b00000001;   //RES = 1
    orajel();
    PORTC &= 0b11111110;   //RES = 0
}
```

Fig. 21. Detail of the program – the code of the software generated RESET and CLK signals

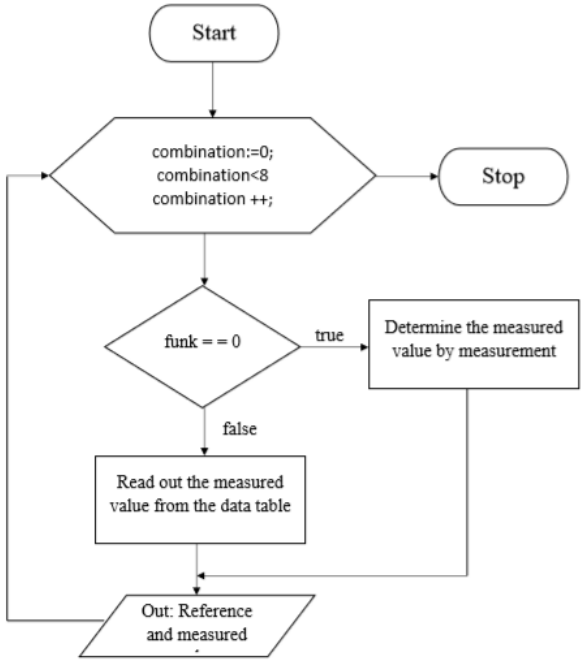

Fig. 22. The algorithm for printing out the data and for the step by step test

g) Manual measurement

In this menu we can choose if we want to measure combinational or sequential network. The program sends out the appropriate timed combinations with the RESET and CLK signals until we push the STOP button (K0), so the signals can be measured on the appropriate measurement pins. The RESET and the clock (CLK) signals are generated by software.

XI. SUMMARY

The description in this article has demonstrated the applicability of the method of digital circuits, for some testing functions.
Further important research, implementation of time-dependent measurement, introduction of the input select function and presentation of the monitoring of analog circuits.
Another interesting application is to create real-time simulation.

XII. REFERENCES

[1] N. Ádám. Single input operators of the df kpi system. Acta Polytechnica Hungarica, 7(1):73–86, 2010.

[2] J. Hurtuk B. Mados, N. Adam and M. Copjak. Brain-computer interface and Arduino microcontroller family software interconnection solution. 2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMI), ISBN 978-1-4673-8739-2, pp. 217-221, Herlany, 2016.

[3] Gy. Györök A. E. Baklanov and B. Beszdes. Extension of Nodal Voltage Method with the Thermosensing. AIS 2017 - 12th International Symposium on Applied Informatics and Related Areas organized in the frame of Hungarian Science Festival 2017 by buda University, ISBN:978-963-449-032-6 p.p. 201-204, Hungary, 2017.

[4] Leon O. Chua and Pen-Min Lin. Computer-Aided Analysis of Electronic Circuit. 1975 Prentice-Hall, INC Englewood Cliffs, New Jersey, USA, 1975.

[5] Gy. Györök, M. Mak´o. Configuration of EEG input-unit by electric circuit evolution. Proc. 9th International Conference on Intelligent Engineering Systems (INES2005), pages 1–7, September 2005.

[6] Gy. Györök. Self configuration analog circuit by FPAA. Proc. 4th Slovakien – Hungarien Joint Symposium on Applied Machine Intelligence (SAMI2006), pages 34–37, January 2006.

[7] Gy. Györök. Self organizing analogue circuit by monte carlo method. Proc. IEEE International Symposium on Logistics and Industrial Informatics (LINDI2007), pages 34–37, September 2007.

[8] Gy. Györök. A-class amplifier with FPAA as a predictive supply voltage control. Proc. 9th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics (CINTI2008), pages 361– 368, November 2008.

[9] Gy. Györök. The FPAA realization of analog robust electronic circuit. Proc. IEEE Internacional Conference on Computational Cybernetics (ICCC2009), pages 1–5, November 2009.

[10] Gy. Györök. Crossbar network for automatic analog circuit synthesis. Proceedings (Liberios Vokorokos, Ladislav Hluch, Jnos Fodor szerk.) of the IEEE 12th International Symposium on Applied Machine Intelligence and Informatics (SAMI 2014). IEEE Computational Intelligence Society,

Budapest: IEEE Hungary Section, ISBN:978-1-4799-3441-6, pages 263– 267, January 2014.

[11] Gy. Györök and B. Beszédes. Duplicated Control Unit Based Embedded Fault-masking Systems. IEEE 15th International Symposium on Intelligent Systems and Informatics : SISY 2017, Szabadka, Szerbia, 2017.09.14-2017.09.16. New York: IEEE, 2017. pp. 283-288. (ISBN:978-1-5386-3855-2, DOI, IEEE Xplore), 2017.

[12] Gy. Györök and B. Beszédes. Highly reliable data logging in embedded systems. SAMI 2018: IEEE 16th World Symposium on Applied Machine

Intelligence and Informatics, proceedings. 237 p., Koice; Herlny, Szlovkia, 2018.02.07-2018.02.10. Seattle (WA): IEEE, 2018. pp. 49-54. (ISBN:978-1-5386-4771-4, REAL), 2018.

[13] Gy. Györök, T. Trifonov, A. E. Baklanov, B. Beszédes, S. V. Grigoryeva, and A. Grigoryeva. A Special Robust Solution for Battery Based Power Supply. 11th International Symposium on Applied Informatics and Related Areas (AIS 2016), Székesfehérvár, Magyarország, 2016.11.17 Budapest: Óbudai Egyetem, 2016. pp. 32-35, Hungary, 2016.

[14] R. Kovacs L. Fozo, R. Andoga. Thermo-dynamic cycle computation of a micro turbojet engine. IEEE, Óbudai Egyetem, ISBN 978-1-5090-3909-8, Budapest, 2016.

[15] Nikoletta Tolner and András Dávid: Examination of digital circuits with simulation and measurement.

[16] Tamás Sándor, Sándor Bedő, Szabolcs Román: Use of 32-bit ARM-based microcontrollers in project tasks at the University of Óbuda. 29th Kandó Conference. Budapest, Magyarország, 2013.11.21 Budapest: Óbudai Egyetem, 2013.

[17] Tamás Sándor, Zolltán Topor, Sándor Bedő: Experiences of 8- and 32-bit microcontroller education at Atmel's Atmel laboratory at the University of Óbuda 28th Kandó Conference. Budapest, Magyarország, 2012.11.08 Budapest: Óbudai Egyetem, 2012.