„POLITEHNICA" UNIVERSITY OF BUCHAREST

ETTI-B DOCTORAL SCHOOL

Decision Nr. 619 from 23.11.2020

DOCTORAL THESIS

THE EXPERIMENTAL DETERMINATION AND EVALUATION OF 8-BIT ARCHITECTURE MICROCONTROLLERS' PERFORMANCE

PhD Student: **Ing. Alexandru Vlădescu (Buturugă)**

DOCTORAL COMMITTEE

President	**Prof. Dr. Ing. Gheorghe Brezeanu**	from	**Politehnica Univ. Bucharest**
PhD Supervisor	**Prof. Dr. Ing. Dan Alexandru Stoichescu**	from	**Politehnica Univ. Bucharest**
Reviewer	Prof. Dr. Ing. Bogdan Ionescu	from	Politehnica Univ. Bucharest
Reviewer	Prof. Dr. Ing. Nicu Bizon	from	Univ. of Pitești
Reviewer	Prof. Dr. Ing. Alexandru Șerbănescu	from	Technical Military Academy of Bucharest

BUCHAREST 2021

Thanks

I express my thanks and gratitude towards prof dr. ing Dan Alexandru Stoichescu, the scientific coordinator of this thesis, for all his support and guidance provided throughout the research activity.

Also, I want to thank ș.l. dr. ing. Rodica Claudia Constantinescu for her support during the research activity and for her collaboration and coordination during the "*uC evaluation system*" project from the UPB-GEX2017 project, as well as the other members of the committee: prof. dr. ing. Adriana Florescu and ș.l. dr. ing. Bogdan Cristian Florea.

I am grateful to ș.l. dr. ing. Valentin Pupezescu for reading this thesis, encouraging me, and providing valuable feedback.

I express my gratitude toward professors dr. ing. Gheorghe Brezeanu, dr. ing. Bogdan Ionescu, dr. ing. Nicu Bizon and dr. ing. Alexandru Șerbănescu for accepting to be part of my doctoral committee.

Finally, I want to thank my family for all the support they have provided during my doctoral studies.

Table of contents

Chapter 1

Introduction

This paper targets the area of evaluating 8-bit architecture microcontrollers. Embedded systems are intended to solve certain operations automatically, in the shortest amount of time possible, using as little electric energy as possible.

One element that influences the performance of an embedded system is the programmable element that controls the entire system. Most often this element is a microcontroller. This paper focuses on evaluating the performance of 8-bit microcontrollers.

8-bit microcontrollers are very widespread in most industry applications; in 2019 the global microcontroller market was evaluated at approximately 8 billion dollars and rising due to the high automotive and IoT market demands. Even so, the 8-bit microcontroller area is less explored in specialty papers, especially from the time performance, current and energy consumption standpoints.

The purpose of this thesis is to describe a new methodology for testing microcontrollers' execution times and drawn current and to perform a study between multiple microcontrollers. The comparative study aims to analyze the microcontrollers' performance and not the performance of the used development software, such as integrated development environment, compilers, or any other computer software.

Throughout this paper I present all the stages needed to obtain reproductible experimental results, starting with the required theoretical knowledge needed to understand and interpret the experimental results, continuing with the test methodology description, both from a hardware and software perspective, ending with the analysis and interpretation of the experimental results. Multiple 8-bit microcontrollers are used and integrated into a comparative study.

The thesis is made up of ten chapters, of which one represents the introduction and another the conclusion. At the end of the thesis there are several annexes and a bibliography section. The introduction contains: the area of study for the doctoral thesis, its purpose and structure plus each chapter's description.

The thesis continues with chapter 2, which includes the theoretical background needed to understand the thesis domain and the experimental results obtained during research. Chapter 3 contains the study and the analysis of the existing research on microcontroller performance.

Chapter 4 details the selection, adaptation and implementation of the test algorithms destined to generate a maximum utilization of the central processing unit. The algorithms are evaluated and their impact on the microcontrollers' operation is estimated. The analysis continues with the exploration of different methods to evaluate the

microcontrollers' time domain performance. The time measurement methods are then validated through experiments.

Chapter 5 presents the test methodology used to evaluate the microcontrollers' performance from an execution time perspective. The characteristics of the chosen microcontroller architectures are described, namely a series of microcontrollers from different architectural families (PIC, AVR, 8051) manufactured by Microchip Technology (including Atmel). Also, at the end of this chapter, the potential errors that can intervene in the microcontrollers' performance analysis are described.

Chapter 6 presents the experimental results obtained from a series of 8-bit microcontrollers from the architectural families described in chapter 5. For each test algorithm described in chapter 4 two types of comparisons are made: the comparison for the same microcontroller oscillator operating frequency and the comparison for the same central processing unit processing speed. In this chapter the execution times for all the algorithms are presented. The results are grouped based on the test algorithm, to allow for easier comparison between microcontrollers.

Chapter 7 reflects the research and design of the test scenarios whose purpose is to measure the current consumption of the studied microcontrollers. These scenarios are experimentally evaluated on a set of test cases and their impact on the microcontrollers is then estimated. The analysis continues with the exploration of methods to evaluate the microcontrollers' current consumption performance. These current measurement methods are validated experimentally on a series of test scenarios.

Chapter 8 presents the results obtained from the current measurements experiments when the microcontrollers execute three test scenarios presented in chapter 7. More details on the test methodology, the experiment conditions and the errors that can intervene in the current measurement process are also provided. For each test sequence described in chapter 7 two types of comparisons are made: for the same microcontroller oscillator operating frequency and the same microcontroller central processing unit speed. The results are grouped based on the test scenario to allow for an easy comparison between microcontrollers.

Chapter 9 presents the result for the microcontrollers' energy consumption evaluation. The analysis is done using microcontrollers executing a series of test scenarios. Also, more details on the potential errors that can intervene in the energy consumption analysis process are provided. The comparison between microcontrollers that operate at the same clock frequency is performed for each scenario and test algorithm described in chapter 7.

The thesis ends with chapter 10, which presents the research conclusions. This chapter contains the achievements, original contributions, a list of original papers on which this thesis is based and potential future research options.

Chapter 2

Preliminary theoretical notions

A microcontroller (μC) can be viewed as a small processing system on a single integrated circuit consisting of a central processing unit (CPU), peripheral devices such as memories, input/output devices (I/O), buses, counters (timers), analog to digital converters and digital to analog converters etc. The concept of System on Chip (SoC) is used. From an architectural point of view microcontrollers can be grouped in several categories; these are briefly detailed below and fully detailed in chapter 5.

Unlike microprocessors, microcontrollers are specialized in detecting and handling interrupts from the outside world, such as the change of the state of a pin or receiving a data word through the serial interface, or internal interrupts, such as the from the timer module.

Although their processing power is much lower than that of x86 based microprocessors, microcontrollers are preferred due to their considerably lower production costs and development time.

There are multiple microcontroller families, such as: 8-bit, 16-bit, and 32-bit. They can consume very low amounts of energy and, most times, include the sleep function where the energy consumption can be reduced to a few nanowatts. This function has made microcontrollers extremely popular in applications such as wireless network sensor nodes or surveillance/monitoring devices because they are usually battery powered and need to last as long as possible.

Chapter 3

Evaluating microcontrollers' performance

To validate an embedded system's parameters tests can be created and used. The tests for functional parameters validate if a parameter has been implemented and works as expected. Nonfunctional parameter tests validate if a parameter is within expected

limits. For example, execution speed and energy efficiency are two nonfunctional parameters that can be measured and validated using nonfunctional performance tests.

A microcontroller's performance analysis is called *benchmarking.* Taking as an example a personal computer, according to [31], benchmarking consists of running a program or set of programs that execute certain tasks with which the performance of the computer's microprocessor can be assessed.

From another perspective, we can consider benchmarking equivalent to nonfunctional performance testing followed by a comparative study of the results gathered from multiple microcontrollers. The most used parameters for determining a microcontroller's performance are the time needed to execute a certain program sequence and the amount of energy required for the execution.

Figure 3.1 graphically represents a generic test sequence used for performance analysis. The OX axis represents the time needed to execute a specially chosen test sequence to generate a high microcontroller load, ideally 100%. The OY axis shows the load of the central processing unit, which can be measure between the two stated: "Idle" (0% load) and "Full load" (100% load).

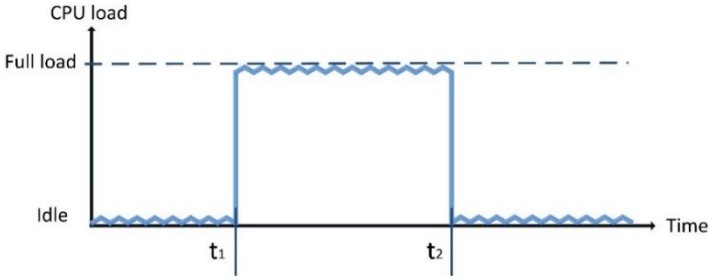

Figure 3.1 Execution time and central processing unit load [5]

$$t_{executie} = t_2 - t_1 \tag{3.1}$$

In other words, the performance parameters can be measured using the execution time for the program sequence $t_{executie}$; $t_{executie}$ is a performance parameter and is calculated as the time difference between the end and start times for the execution of the test sequence. Applying this concept, the performance parameter representing the amount of consumed energy can also be determined.

In the specialty literature there are a substantial number of books and articles that handle the performance evaluation for microcontrollers and personal computers [31], [32], [33]. There are also articles related to the subject of this thesis, but they are not representative.

As a result of my research, I have identified a series of articles that consider microcontrollers' performance in various situations and scenarios [38], [39] and [40]. They do not handle the particular case of 8-bit microcontrollers and focus on 16-bit or 32-bit microcontroller architectures.

To measure the execution times and consumed current a series of measurement instruments are needed. These parameters can be measured and determined with the help of an automated test system. Such a system does not exist and, even if it did, it

would not be flexible enough to be used with different architectures of 8-bit microcontrollers.

To showcase the need for microcontroller performance testing instruments and systems I have published a series of articles that handle evaluating microcontroller performance from both an execution time perspective, as well as the energy consumption perspective: [1], [2], [3], [4], [5], [6], [7].

Chapter 4

Time measurement system: test algorithms and measurement

In this chapter a series of algorithms that can be used for studying microcontroller performance are analyzed and presented. For each algorithm, the following information is provided: description, purpose, and implementation in the C programming language.

As a result of my research, several software algorithms for the microcontroller evaluation system have been identified. They are implemented as a test sequence withing a test program and are used to extract performance parameters for the microcontrollers; they are designed and implemented so as not to be dependent on a specific microcontroller, meaning that they are implemented in a generic manner to allow their execution on any microcontroller that needs testing.

The test program represents the entirety of the C code written for a microcontroller, and the test sequence represents the instruction set that form an algorithm whose purpose is to load, up to 100%, the microcontroller's central processing unit. The performance parameter to be measured is the time needed to execute the test sequence.

The identified algorithms are:

4.1.12 AES128_ENCRIPT algorithm

4.1.13 AES128_DECRIPT algorithm

Next, two of the algorithms are described in detail: the MUL_INT algorithm and the FIND_MIN_ARRAY_RAM_8bit algorithm.

The *MUL_INT* algorithm repeatedly executes a set of integer multiplication operations. The algorithm performs successive multiplications to calculate the expression:

$$x = 3^{20} = 3 * 3 * 3 * \ldots * 3 \quad (4.4)$$

$$x = 3.486.784.401 \quad (4.5)$$

This number (*x*) is deliberately chosen to be representable using an unsigned 32-bit data type. The maximum unsigned value that can be stored using 32 bits is $2^{32} - 1 = 4.294.967.295$. When using compilers from Mikroelektronika the "unsigned long" data type is used.

The purpose of the algorithm is to provide a high load and intensive utilization for the microcontroller's central processing unit. By executing this algorithm we can obtain information about a microcontroller's ability to process intensive arithmetic operations, using integers.

The C language software implementation is presented below:

```
void RunTestMulInt(){
    int i;
    UINT32 x = 3;
    TP_1 = 1; //signals the start of the test
    for(i = 0; i < 19; i++){
            x = x * 3;
        }
    TP_1 = 0; //signals the end of the test
}
```

The FIND_MIN_ARRAY_RAM_8bit algorithm consists of searching for the minimum value from a randomly generated 64 element 8-bit array, stored in RAM. The data array is always generated to have the same elements to be able to obtain consistent and repeatable results. The minimum value is determined by going through the array and comparing the elements with the previously found minimum value. For this algorithm, the data array is stored in the data memory.

This algorithm requires a 64 byte amount of RAM memory (*MemDate*) to store the array in which the minimum value search is performed. A potential problem for older generation microcontrollers is the impossibility of running the algorithm due to insufficient RAM memory.

$$MemDate = dimensiune_{element} * nr_{elemente} \quad (4.14)$$

$$MemDate = 8\ bit * 64 = 512\ bit = 64\ octeti \quad (4.15)$$

The purpose of the algorithm is to provide a high load and utilize the microcontroller's central processing unit as much as possible. By executing the search for the minimum value, we can obtain information about a microcontroller's ability to work with intensive execution loops and accessing a data array stored in its RAM memory. Also, this algorithm helps evaluate the RAM memory read access times when working with 8-bit wide data types. The C language software implementation is shown below:

```
    UINT8 arrayFindMin(const UINT8 *vect, int size){
        int i = 0;
        UINT8 min = vect[0];
        for (i=1; i<size; i++){
                if (vect[i] < min){
                        min = vect[i];
                }
        }
        return min;
    }
    void RunTestFindMinArrayRAM(){
        UINT8 dummyVar;
        static UINT8 array8bit[64] = {41, 107, 214, 235, 44, 169, 3, 33, 187, 239, 95, 95,
76, 252, 16, 236, 190, 212, 237, 81, 6, 69, 77, 153, 37, 142, 81, 101, 83, 5, 92, 51, 236,
63, 84, 22, 167, 34, 205, 204, 143, 96, 212, 243, 78, 74, 96, 61, 203, 238, 47, 104, 22,
117, 147, 109, 53, 51, 244, 13, 76, 230, 5, 57};
        TP_1 = 1; //signals the start of the test
        dummyVar = arrayFindMin(array8bit,64);
        TP_1 = 0; //signals the end of the test
    }
```

The software algorithms for the microcontroller performance evaluation system presented so far are the result of my own research. Several algorithms have been identified, each with their own specific properties.

For each algorithm I presented: its purpose, usage, implementation, and the performance parameter that can be extracted after running the algorithm as part of the test sequence. Following, the algorithms are presented based on the performance parameters they can highlight:

- By executing the *MUL_INT* algorithm we can obtain information about a microcontroller's ability to process intensive arithmetic operations. Depending on the arithmetical logic unit's ability to perform these operations a performance difference between the tested microcontrollers will be observed.
- By executing the *FIND_MIN_ARRAY_RAM_8bit* algorithm we can obtain information about a microcontroller's ability to work with execution loops which utilize data stored at consecutive addresses in the RAM memory. Depending on the data bus and read access times for the data memory, performance differences will be observed between the tested microcontrollers.

According to [5], I have identified two possible methods to measure the execution times for microcontrollers: the internal and external methods. In this chapter the two execution time measurement methods are explored and applied to a series of test cases to determine their advantages and disadvantages.

Throughout this thesis I have preferred the external method because it requires a lower test program complexity, the external measurement instruments have a high accuracy, and the result can be verified using a computer.

Chapter 5

Execution time measurement system: microcontroller architectures and measurement errors

This chapter present the methodology used to measure the microcontrollers' execution time. To be able to better understand and interpret the experimental results, the features of the chosen microcontroller architectures are described. The microcontrollers used are manufactured by Microchip Technology (including Atmel) and represent different architectural families.

The tested microcontrollers have been chosen to cover as wide of a range as possible of the 8-bit microcontroller portfolio.

These microcontrollers can be classified in three distinct architecture categories: PIC architecture, AVR architecture and 8051 architecture. Before presenting the experimental results, the three architecture types have been described in order to identify the architectural elements that can impact performance and to have a better understating of the experimental results.

The architectural traits that can influence a microcontroller's time performance are:

- the complexity of the instruction set,
- the time needed to execute an instruction,
- the width of the data bus,
- the width of the program bus,
- the addressing modes for data and program access,
- the time needed to access the data memory,
- the time needed to access the program memory,
- the complexity of the arithmetical logic unit,
- the clock frequency / operating speed.

The experimental results can be influenced by one of these parameters or, most often, by a combination of them. This happens because a microcontroller is a mini processing system which integrates multiple components and closely interconnected modules.

Chapter 6

Comparative analysis of the execution time measurement results

This chapter presents the results of the microcontrollers' execution time measurements for the algorithms presented previously. For each test algorithm two comparisons are made: for the same microcontroller clock frequency and for the same central processing unit speed. The results are grouped based on the test algorithm. The experimental results grouped by the microcontroller under test are presented in annex 3.

The clock frequency is listed in datasheets as a maximum value at which the microcontroller can operate. The experiments have been performed at five distinct clock frequencies: 1 MHz, 2 MHz, 4 MHz, 8 MHz, and 16 MHz.

The experimental results obtained for each algorithm are presented in tables 6.1 – 6.12 and in multiple figures. The figures represent the experimental result for all the microcontrollers and for all operating frequencies when compiler optimizations are set to level zero or to level four.

For the charts generated for each test algorithm the same OY axis scale was kept, to allow an easier comparison between the tested microcontrollers.

Table 6.1 MUL_INT – execution time values for different clock frequencies

MUL_INT	Time for optimization level zero [ms]					Time for optimization level four [ms]				
Fosc / Microcontroller	1 MHz	2 MHz	4 MHz	8 MHz	16 MHz	1 MHz	2 MHz	4 MHz	8 MHz	16 MHz
PIC16F1509	51.11	25.57	12.79	6.388	3.196	50.96	25.48	12.74	6.368	3.186
PIC16F1787	51.47	25.73	12.87	6.429	3.206	51.248	25.624	12.812	6.406	3.203
PIC16F688	51.27	25.63	12.81	6.405	3.202	51.26	25.63	12.81	6.384	3.202
PIC16F18326	51.55	25.57	12.77	6.394	3.191	51.43	25.57	12.73	6.373	3.183
PIC16F19156	51.184	25.592	12.796	6.398	3.199	51.008	25.504	12.752	6.376	3.188
PIC16F84A	51.2	25.6	12.8	6.4	3.2	51.2	25.6	12.8	6.4	3.2
PIC12F675	51.72	25.86	12.93	6.465	3.2325	51.72	25.86	12.93	6.465	3.2325
PIC18F4525	46.73	22.36	11.68	5.835	2.889	46.6	23.3	11.65	5.825	2.913
PIC18F2550	47.57	23.78	11.89	5.939	2.972	47.57	23.78	11.89	5.939	2.972
PIC18F13K50	46.89	23.45	11.73	5.859	2.926	46.89	23.47	11.7	5.866	2.931
PIC18F47K42	46.848	23.424	11.712	5.856	2.928	46.792	23.396	11.698	5.849	2.925
PIC18F24J10	46.832	23.416	11.708	5.854	2.927	46.832	23.416	11.708	5.854	2.927
ATMEGA328P	1.671	0.835	0.4117	0.208	0.104	1.392	0.696	0.348	0.174	0.087
ATMEGA8	1.659	0.83	0.41	0.209	0.104	1.662	0.831	0.412	0.209	0.104
ATMEGA16	1.688	0.844	0.422	0.211	0.106	1.664	0.832	0.416	0.208	0.104
ATMEGA32	1.688	0.844	0.422	0.211	0.106	1.672	0.836	0.418	0.209	0.105
ATMEGA1284	1.706	0.845	0.431	0.218	0.104	1.698	0.835	0.432	0.216	0.104
AT89S8253	33.936	16.968	8.484	4.242	2.121	33.936	16.968	8.484	4.242	2.121
AT89S52	33.936	16.968	8.484	4.242	2.121	33.936	16.968	8.484	4.242	2.121
AT89S2051	33.936	16.968	8.484	4.242	2.121	33.936	16.968	8.484	4.242	2.121
AT89LP828	3.008	1.504	0.752	0.376	0.188	3.008	1.504	0.752	0.376	0.188
AT89LP214	3.008	1.504	0.752	0.376	0.188	3.008	1.504	0.752	0.376	0.188

After interpreting the experimental results, the following can be stated:

- the time needed to execute the MUL_INT algorithm decreases as the clock frequency increases, for all tested microcontrollers;
- there are groups of microcontrollers for which the MUL_INT algorithm is executed in a similar amount of time;
- the best results were obtained by the megaAVR architecture microcontrollers and the worst results were obtained by PIC architecture microcontrollers; between the two groups there is a performance ratio of approximately 1/30 in favor of the megaAVR microcontrollers;
- the results obtained for level zero optimizations are similar to the results obtained for level four optimizations, for all studied microcontrollers;

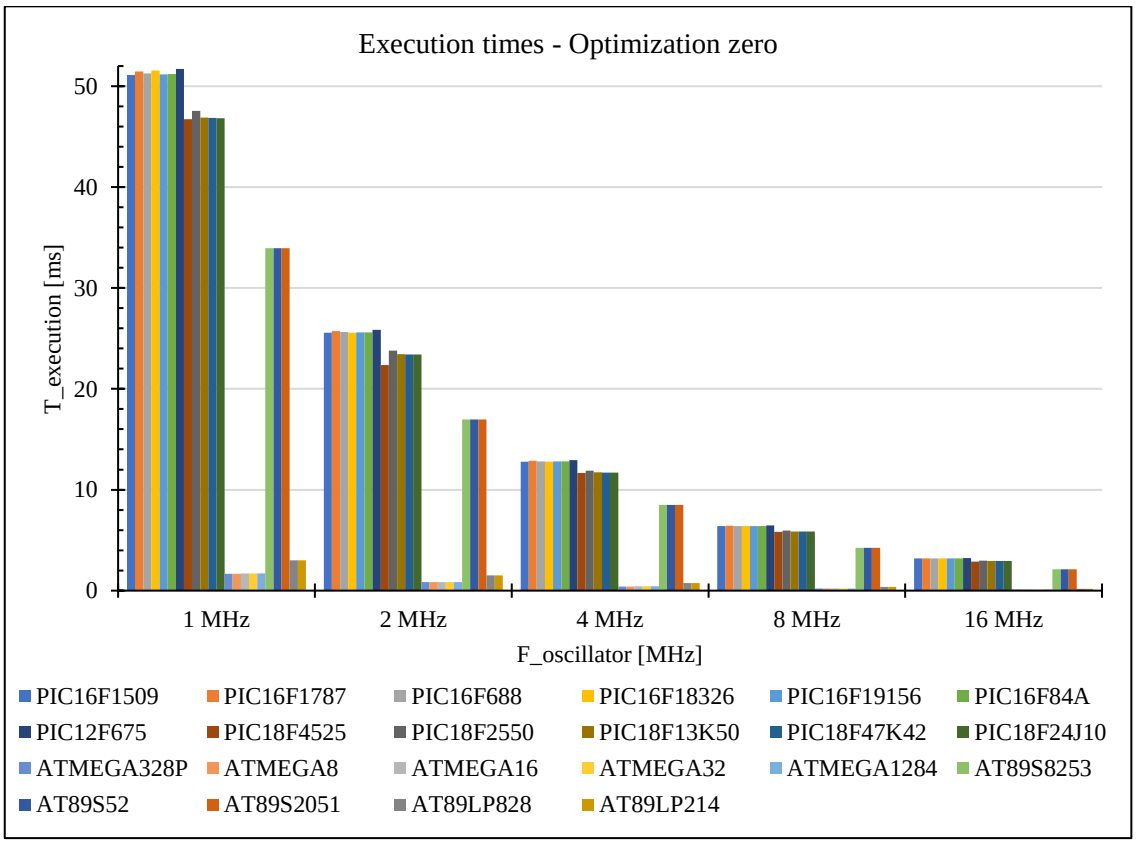

Figure 6.1 MUL_INT – dependency of the execution time on the clock frequency for optimization level zero

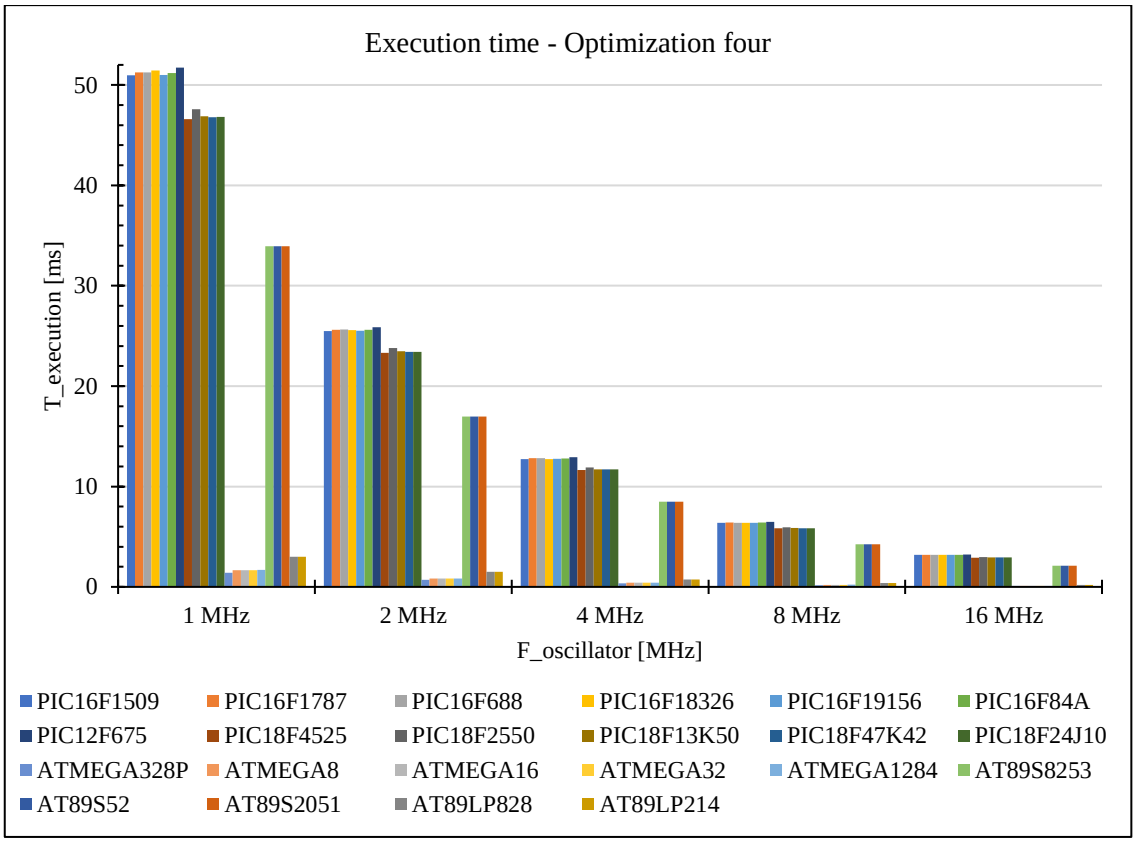

Figure 6.2 MUL_INT – dependency of the execution time on the clock frequency for optimization level four

Table 6.2 FIND_MIN_ARRAY_RAM_8bit – execution time values for different clock frequencies

FIND_MIN_ARRAY RAM_8bit	Time for optimization level zero [ms]					Time for optimization level four [ms]				
Fosc / Microcontroller	1 MHz	2 MHz	4 MHz	8 MHz	16 MHz	1 MHz	2 MHz	4 MHz	8 MHz	16 MHz
PIC16F1509	7.747	3.875	1.938	0.968	0.485	7.392	3.696	1.848	0.924	0.462
PIC16F1787	7.801	3.901	1.951	0.974	0.486	7.44	3.72	1.86	0.93	0.465
PIC16F688	6.646	3.322	1.66	0.83	0.415	6.637	3.318	1.658	0.829	0.415
PIC16F18326	7.466	3.728	1.848	0.925	0.462	7.459	3.728	1.847	0.924	0.461
PIC16F19156	7.4096	3.7048	1.8524	0.9262	0.4631	7.392	3.696	1.848	0.924	0.462
PIC16F84A	-	-	-	-	-	-	-	-	-	-
PIC12F675	-	-	-	-	-	-	-	-	-	-
PIC18F4525	6.39	3.195	1.597	0.798	0.4	6.368	3.184	1.592	0.796	0.398
PIC18F2550	6.509	3.254	1.627	0.812	0.407	6.504	3.234	1.626	0.811	0.406
PI8F13K50	6.415	3.207	1.604	0.802	0.4	6.411	3.206	1.599	0.802	0.4
PIC18F47K42	6.664	3.332	1.666	0.833	0.417	6.648	3.324	1.662	0.831	0.416
PIC18F24J10	6.4	3.2	1.6	0.8	0.4	6.4	3.2	1.6	0.8	0.4
ATMEGA328P	2.366	1.183	0.591	0.295	0.148	1.344	0.672	0.336	0.168	0.084
ATMEGA8	2.35	1.175	0.582	0.296	0.147	1.403	0.702	0.348	0.176	0.088
ATMEGA16	2.36	1.18	0.59	0.295	0.148	1.408	0.704	0.352	0.176	0.088
ATMEGA32	2.368	1.184	0.592	0.296	0.148	1.408	0.704	0.352	0.176	0.088
ATMEGA1284	2.404	1.183	0.61	0.305	0.148	4.6	0.704	0.364	0.182	0.88
AT89S8253	22.224	11.112	5.556	2.778	1.389	22.224	11.112	5.556	2.778	1.389
AT89S52	22.224	11.112	5.556	2.778	1.389	22.224	11.112	5.556	2.778	1.389
AT89S2051	22.224	11.112	5.556	2.778	1.389	22.224	11.112	5.556	2.778	1.389
AT89LP828	3	1.5	0.75	0.375	0.188	3	1.5	0.75	0.375	0.188
AT89LP214	3	1.5	0.75	0.375	0.188	3	1.5	0.75	0.375	0.188

After interpreting the experimental results, the following can be stated:

- the time needed to execute the FIND_MIN_ARRAY_RAM_8bit algorithm decreases as the clock frequency increases, for all the tested microcontrollers;
- there are groups of microcontrollers for which the FIND_MIN_ARRAY_RAM_8bit algorithm is executed in a similar amount of time;
- the best results were obtained by the AVR architecture microcontrollers and the worst results were obtained the by the 8051 architecture microcontrollers (AT89S family); between the two groups there is a performance ratio of approximately 1/10 in favor of the AVR microcontrollers;
- the results obtained for level zero optimization are similar with the level four optimization results for PIC and 8051 based microcontrollers. For the AVR architecture microcontrollers an increase in performance is observed when using level four optimization;
- the algorithm couldn't be tested on the PIC16F84A and PIC12F675 microcontrollers due to insufficient RAM memory;

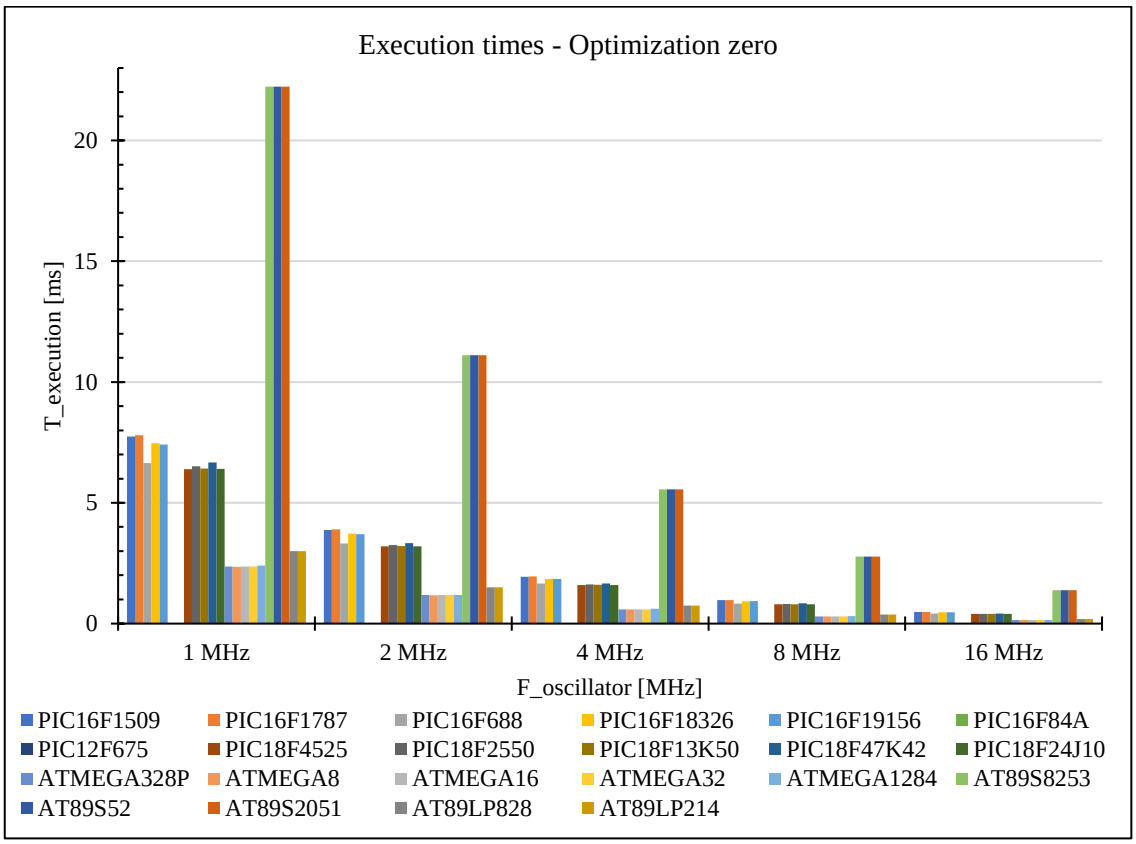

Figure 6.19 FIND_MIN_ARRAY_RAM_8bit – – dependency of the execution time on the clock frequency for optimization level zero

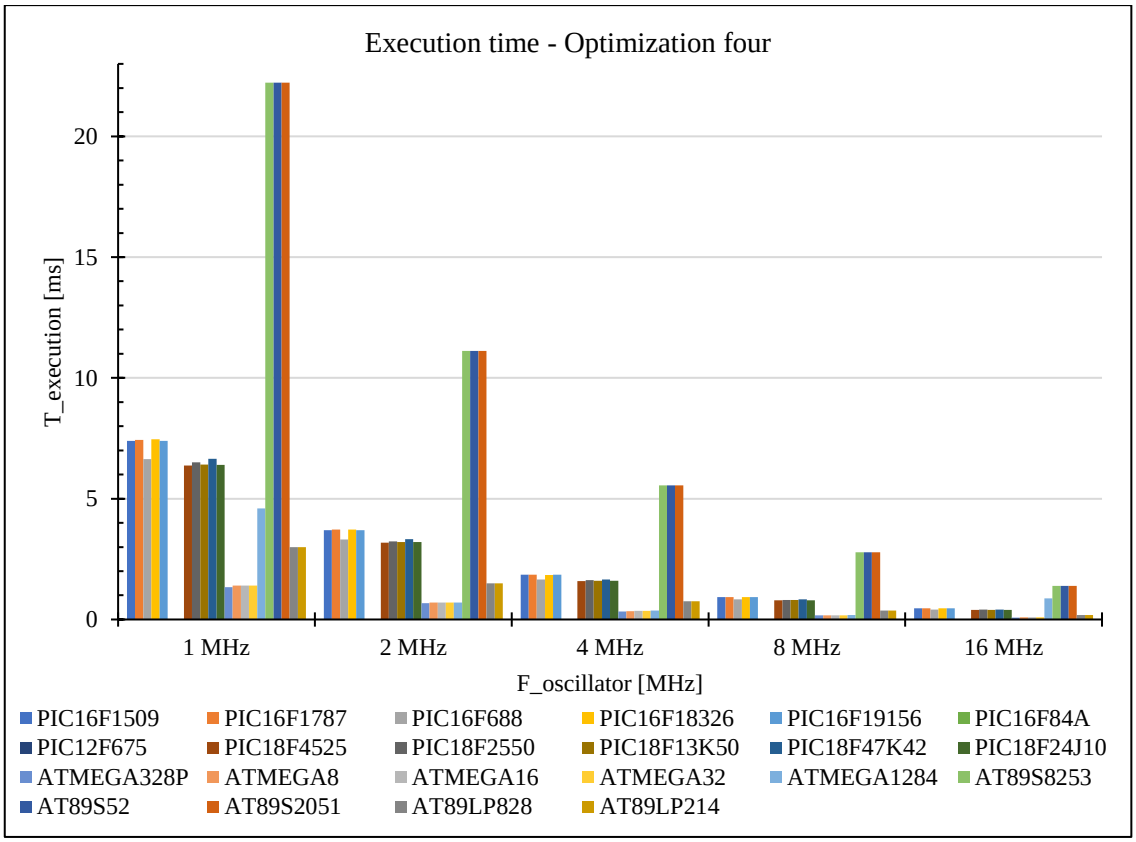

Figure 6.20 FIND_MIN_ARRAY_RAM_8bit – – dependency of the execution time on the clock frequency for optimization level four

Chapter 7

System for determining current consumption

The performance parameter to be measured in this chapter is the current drawn by the microcontroller. A microcontroller can draw more or less current, depending on its configuration and the instructions it executes in a certain amount of time.

According to [1] and [3] a microcontroller's current consumption can be influenced by several parameters and can be experimentally studied using multiple algorithm-based scenarios.

Figures 7.2 and 7.3 present theoretical scenarios for a microcontroller's current consumption when it executes a series of different test sequences (ST0, ST1, ST2, etc.). The expectation is that the execution of each sequence will produce a variation of the

microcontroller's current consumption in time. For different test sequences a different current consumption is expected. Of course, when executing the same test sequence an approximately constant current consumption is expected.

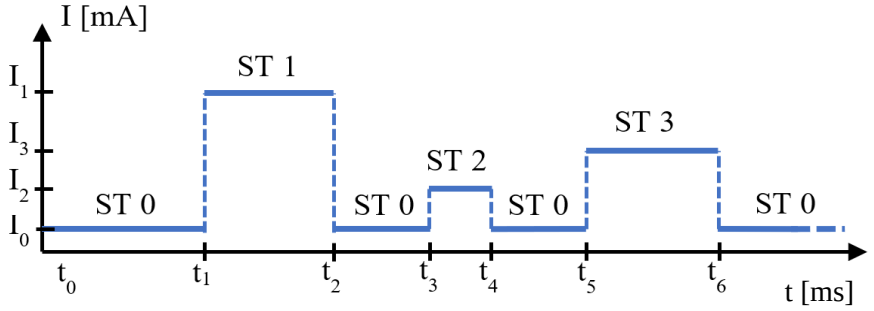

Figure 7.1 Microcontroller current consumption – multiple test sequences

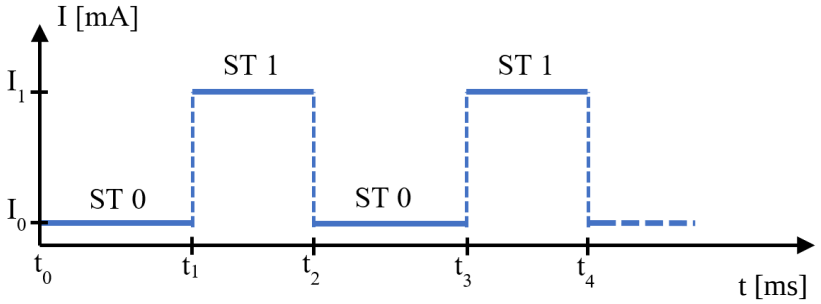

Figure 7.2 Microcontroller current consumption – two test sequences

To experimentally determine the microcontroller's current consumption throughout a test sequence we need to measure the I_0 and I_1 values. Formulas (7.1) and (7.2) provide the microcontroller's current consumption during the execution of a test sequence (I_{ST1}) and the time needed to execute a test sequence (t_{ST1}). The formulas are valid if the current consumed by the microcontroller during the test sequences is relatively constant.

$$I_{ST1} = I_1 - I_0 \quad (7.1)$$

$$t_{ST1} = t_2 - t_1 \quad (7.2)$$

As a result of my research, the identified scenarios for the experimental evaluation of a microcontroller's current consumption are:

7.1.1 IDLE scenario

7.1.2 INTERNAL_MODULE scenario

7.1.3 CORE_LOAD scenario

In this chapter the test scenarios are described in detail. For each test scenario the algorithms are presented, along with their purpose, utilization, implementation, and the performance parameter that can be extracted.

The *CORE_LOAD* scenario uses algorithms that create a high load on the microcontroller's central processing unit. These algorithms were presented in chapter 4.

This scenario can be used to obtain information on the microcontroller's current consumption when it executes a series of instructions. These instructions are part of the algorithms intended to evaluate the microcontroller's time performance. In this scenario we can obtain experimental results that show the amount of current needed by a microcontroller to execute an algorithm.

The MUL_INT and DIV algorithms were chosen because, by executing them, we can extract information about the microcontroller's current consumption when it executes intensive arithmetic operations. The SORT_ARRAY_8bit algorithm is important for extracting information about a microcontroller's current consumption while it executes intensive loops and read/write data transfers using the data memory. By executing the FIND_MIN_ARRAY_FLASH_8bit we can extract information about a microcontroller's current consumption when it performs intensive processing loops and data transfers for reading a constant data set from the program memory.

The CORE_LOAD scenario can be used for analyzing both the instantaneous and average current consumption of a microcontroller under test. Considering that the test algorithms' implementation does not differ from the versions presented in chapter 4, next I present two generic test sequences that apply to all of the four algorithms. By measuring the current consumption in this scenario, we can gather information about the impact the algorithm execution has on the current drawn by the microcontroller under test.

Below are two C language software implementations of the algorithm. The first version is used to measure the instantaneous current consumption and the second is used to measure the average current consumption.

```
void PWR_RunTestCoreLoad(unsigned int time){
    ExecuteAlgorithm();
    PWR_Delay(time);
    ExecuteAlgorithm();
    PWR_Delay(time);
}
```

```
void PWR_RunTestCoreLoadAverage(){
    while(1){
        ExecuteAlgorithm();
    }
}
```

For the average current measurement test sequence implementation, the algorithm is run continuously in an infinite loop. This is necessary to keep the microcontroller's central processing unit busy with the execution of the test algorithm, without having to execute other parts of the test program. Thus, the microcontroller's average current consumption can be measured while it executes only the desired test algorithm.

Next, we will explore the methods used for measuring the current consumption for the microcontroller under test. As described in chapter 3, evaluating a microcontroller's current consumption requires determining the amount of drawn current during the execution of a certain program section, as illustrated in figure 3.1. This study can be extended to determine the energy consumption if we take into account both the execution time and the power consumed during this execution.

From the current measurement perspective, we have two distinct values that are measured: the instantaneous and the average current.

Throughout the analysis of the three scenarios, two measurement methods for the microcontroller's instantaneous current consumption were used:

- measurement using a multimeter configured as an ammeter.
- measurement using an oscilloscope and a current probe.

Both methods yielded similar results for the instantaneous current measurements. Throughout the study the instantaneous current consumption for the microcontrollers under test was observed to be constant in time, for each used scenario; as a result, the main advantage provided by the oscilloscope measurement method, a high sampling frequency, brings no major advantage in extracting the experimental results. On the other hand, the digital multimeter used in the experiments, Aim TTi 1908, has a good 0.05% ± 5 digit accuracy for measuring DC currents. Based on these findings, to determine the current consumption, the digital multimeter configured as an ammeter measurement method will be used.

Chapter 8

Experimental results obtained for the current consumption measurements

This chapter presents the results obtained for the current consumption measurements for microcontrollers running the three test scenarios presented in chapter 7. Details about the test methodology are provided, as well as the conditions in which the experiments took place and the errors that can intervene in the current measurement process.

For each test sequence described in the previous chapter two type of comparisons are made for the tested microcontrollers: for the same microcontroller oscillator frequency and the same microcontroller processing speed. The results are grouped based on the test algorithm to allow for an easier comparison between the tested microcontrollers.

The current consumption experiments are realized using a series of 8-bit microcontrollers. They are tested in isolation, to minimize the errors that can appear. To extract the experimental results, the studied microcontrollers' current consumption is analyzed throughout the execution of the three test sequences. The digital multimeter configured as an ammeter measurement method is used. During the experiments, the instantaneous current is measured at multiple consecutive moments in time and their average value is then calculated.

The performance analysis is performed on a series of microcontrollers manufactured by Microchip Technology. These microcontrollers are part of three separate architectural categories: the PIC architecture, AVR architecture and 8051 architecture. The characteristics of these architectures were presented in chapter 5. The conditions

under which the current measurement experiments are performed are similar to the ones described in chapter 5.

Next, I analyze the impact of the clock frequency on the microcontrollers' current consumption. The experiments are conducted at five distinct clock frequencies: 2 MHz, 4 MHz, 8 MHz, 16 MHz and 20 MHz, with the exception of the megaAVR microcontrollers whose maximum operating frequency cannot exceed 16 MHz.

The tested microcontrollers' current consumption is analyzed during the execution of the test scenarios that were presented in detail and experimentally validated in chapter 7: IDLE, INTERNAL_MODULE, CORE_LOAD (MUL_INT, DIV, SORT_ARRAY_8bit, FIND_MIN_ARRAY_FLASH_8bit algorithms). The experimental results for each microcontroller under test are presented in tables and figures.

Table 8.1 CORE_LOAD – MUL_INT scenario – different microcontrollers' average current consumption depending on the clock frequency

CORE_LOAD - MUL_INT – I_average_total [mA]					
Fosc / Microcontroller	2 MHz	4 MHz	8 MHz	16 MHz	20 MHz
PIC16F1509	0.706	0.817	1.221	2.003	2.314
PIC16F1787	0.748	0.957	1.415	2.2845	2.639
PIC16F18326	0.955	1.04	1.433	2.225	2.558
PIC16F688	1.005	1.235	1.812	2.89	3.371
PIC18F13K50	0.993	1.397	2.262	3.974	4.762
PIC18F2550	1.998	3.086	5.291	9.688	11.77
PIC18F4525	1.828	2.749	4.67	8.462	10.254
AT89S8253	8.02	8.332	8.948	10.106	10.629
AT89S52	7.839	8.206	9.585	12.16	13.37
AT89LP828	4.402	6.095	9.439	13.555	15.725
ATMEGA328P	8.527	10.036	12.575	16.975	-
ATMEGA16	8.386	10.916	15.88	25.445	-
ATMEGA32	8.742	12.015	17.225	27.19	-
ATMEGA1284	6.831	7.184	8.671	11.91	12.78

Table 8.2 CORE_LOAD scenario – the MUL_INT algorithm impact on the microcontrollers' average current consumption depending on the clock frequency

CORE_LOAD - MUL_INT – I_average_impact [mA]					
Fosc / Microcontroller	2 MHz	4 MHz	8 MHz	16 MHz	20 MHz
PIC16F1509	0.031	0.054	0.1095	0.217	0.27
PIC16F1787	0.07	0.108	0.167	0.2825	0.363
PIC16F18326	0.027	0.047	0.037	0.163	0.238
PIC16F688	0.064	0.131	0.239	0.372	0.459
PIC18F13K50	0.109	0.17	0.294	0.471	0.657
PIC18F2550	0.299	0.555	1.036	2.011	2.474
PIC18F4525	0.2	0.367	0.686	1.314	1.605
AT89S8253	0.326	0.35	0.386	0.474	0.478
AT89S52	0.027	0.049	0.122	0.2	0.26
AT89LP828	0.256	0.385	0.68	1.58	1.965
ATMEGA328P	0.468	0.705	1.935	2.59	-
ATMEGA16	0.254	0.729	1.6	3.335	-
ATMEGA32	0.437	1.003	1.84	3.735	-
ATMEGA1284	0.086	0.136	0.493	0.975	1.155

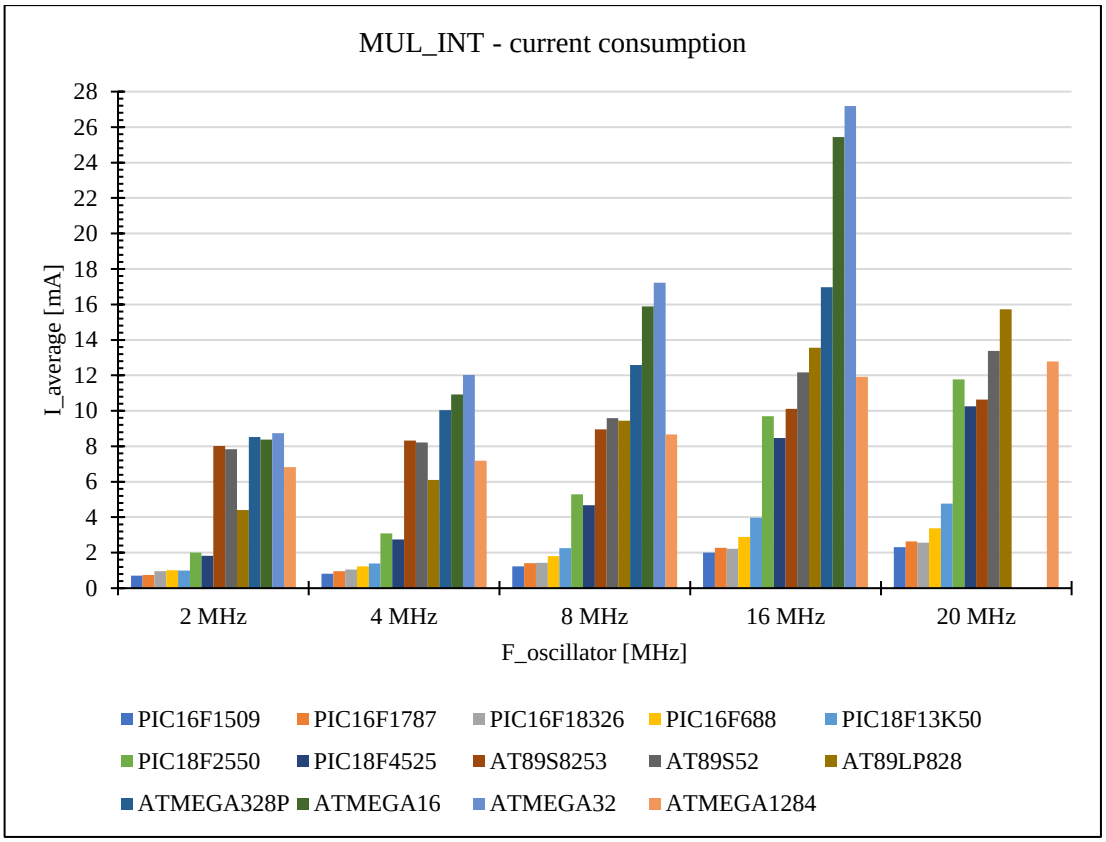

Figure 8.1 CORE_LOAD – MUL_INT scenario– microcontroller average current consumption

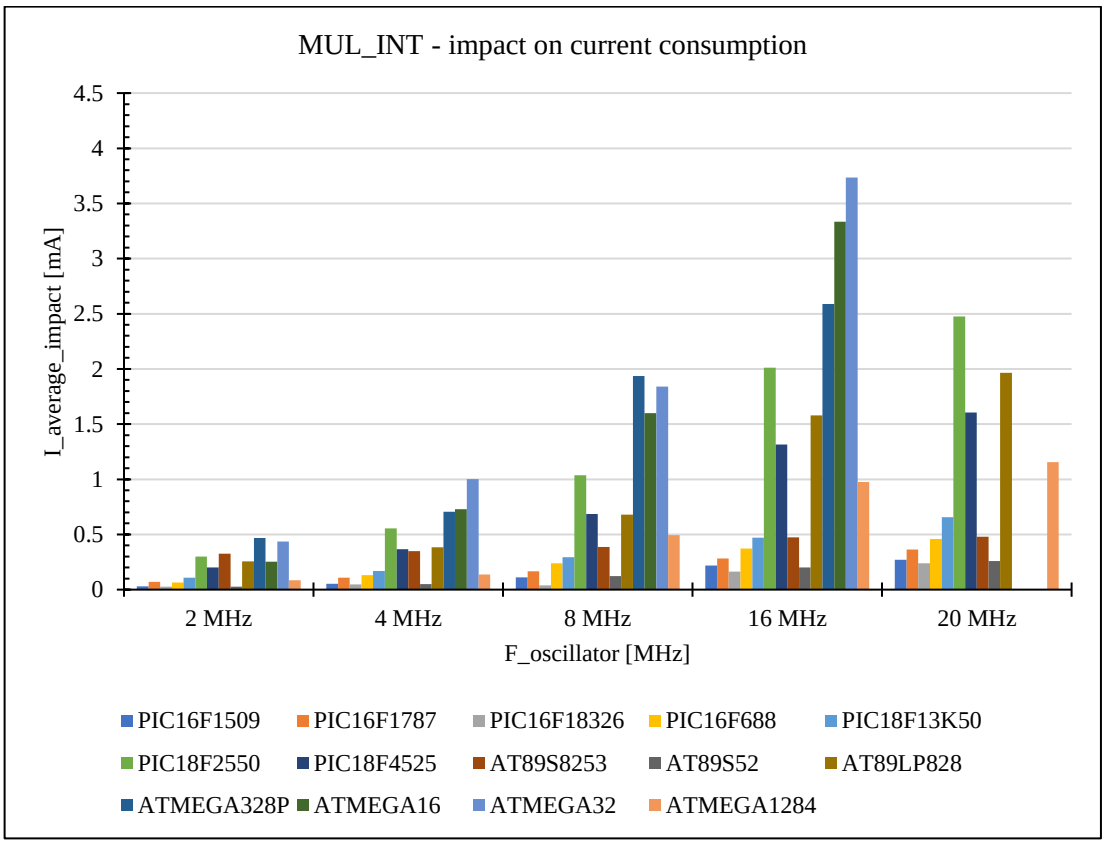

Figure 8.2 CORE_LOAD scenario –MUL_INT algorithm impact on the average current consumption

Chapter 9

Microcontrollers' energy consumption

This chapter examines the dependency between the test scenarios and the energy consumption for several types of microcontrollers. The comparison is made between multiple microcontrollers that operate at the same clock frequency ($F_{oscilator}$) and execute the test scenarios described in chapter 7.

The results are grouped based on the test scenario and algorithm to allow for an easier comparison between the tested microcontrollers. The results from this chapter are obtained by combining the experimental results for the execution time of the test algorithms and the microcontrollers' current consumption during the execution of these algorithms.

Next, the impact of the clock frequency on the microcontrollers' energy performance is being analyzed. The energy consumption is evaluated at four distinct clock frequencies: 2MHz, 4MHz, 8MHz and 16MHz. The microcontroller's energy consumption is evaluated during the execution of the following test scenarios, presented in detail and validated in chapter 7: IDLE, CORE_LOAD (MUL_INT, DIV, SORT_ARRAY_8bit, FIND_MIN_ARRAY_FLASH_8bit algorithms).

The energy values for each operating frequency and microcontroller under test are presented in table 9.1 for the CORE_LOAD scenario.

Table 9.1 Microcontrollers' energy consumption for the CORE_LOAD – MUL_INT scenario

CORE_LOAD - MUL_INT - E [mJ]				
Fosc / Microcontroller	2 MHz	4 MHz	8 MHz	16 MHz
PIC16F1509	0.090262	0.052247	0.038999	0.032008
PIC16F1787	0.09623	0.061583	0.045485	0.036621
PIC16F18326	0.122097	0.066404	0.045813	0.0355
PIC16F688	0.128791	0.079102	0.058029	0.046269
PIC18F13K50	0.116429	0.081934	0.066265	0.05814
PIC18F2550	0.237562	0.183463	0.157116	0.143964
PIC18F4525	0.20437	0.160542	0.136247	0.122234
AT89S8253	0.680417	0.353443	0.189787	0.107174
AT89S52	0.665061	0.348099	0.203298	0.128957
AT89LP828	0.033103	0.022917	0.017745	0.012742
ATMEGA328P	0.0356	0.020659	0.013078	0.008827
ATMEGA16	0.035389	0.023033	0.016753	0.013486
ATMEGA32	0.036891	0.025352	0.018172	0.014411
ATMEGA1284	0.028861	0.015482	0.009451	0.006193

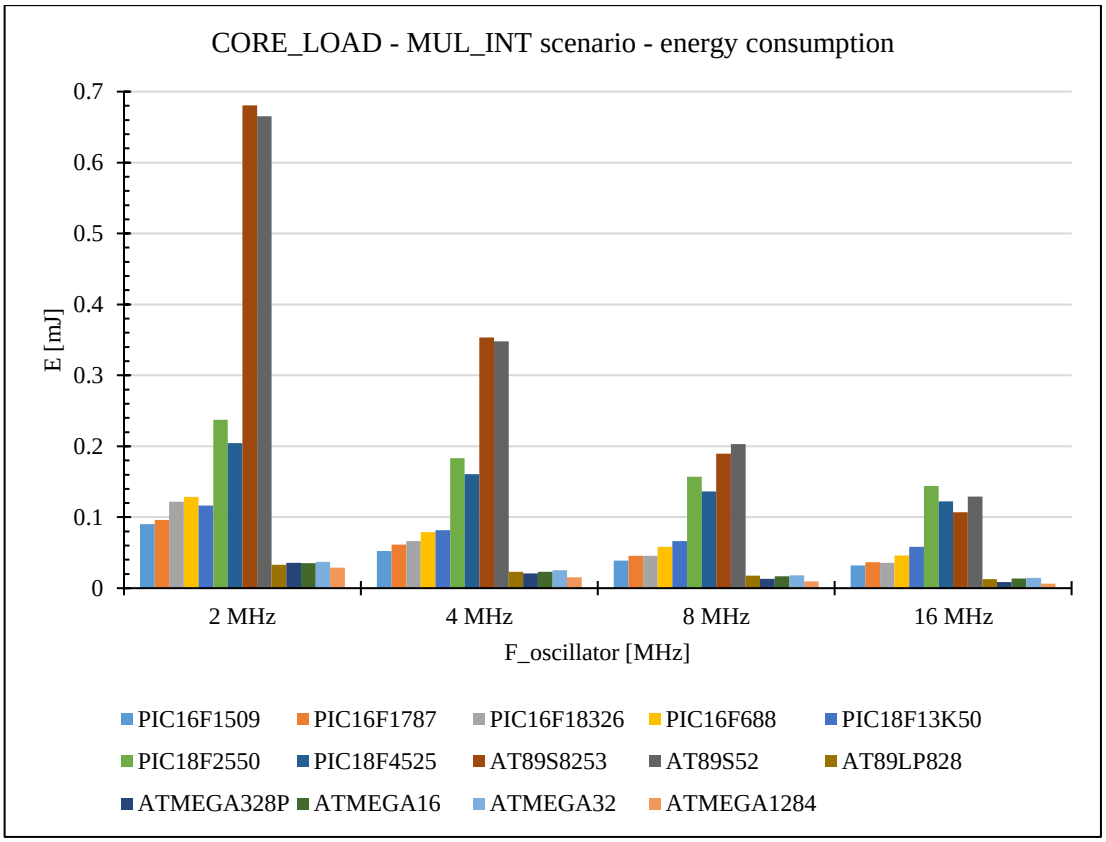

Figure 9.1 Microcontrollers' energy consumption for the CORE_LOAD – MUL_INT scenario

Chapter 10

Conclusions

In chapters 1 and 2 I presented the thesis domain, the fundamental notions needed for the study and I justified the decision to analyze the performance of 8-bit microcontrollers. In chapter 3 a series of specialty articles were reviewed and the existing research for the thesis research field has been presented, from the perspective of evaluating a microcontroller's performance. As a result of my research, I have not identified any papers that explicitly evaluate the performance for 8-bit microcontrollers from the execution time, current consumption, and energy consumption perspectives.

The thesis continues with chapters 4-9 in which the experiments for determining the execution time performance and current consumption performance are being described. At the end of each chapter, I presented a series of conclusions based on the obtained experimental results.

In chapter 4 I designed the test sequences used to evaluate the execution time performance. I also analyzed two different execution time measurement techniques and applied them to extract the first set of partial experimental results. At the end of the chapter, I concluded that all thirteen test algorithms can be used to evaluate the execution time and that the external time measurement method is preferred, due to the lower complexity needed for the test program, the accuracy of the external measurement instruments, the fact that they can work independently from the microcontroller and can be directly controlled from a computer. In chapter 5 I developed the test methodology for evaluating a microcontrollers execution time performance.

Also in chapter 5, I selected the microcontrollers to be studied and described their architectural characteristics. The performance analysis was done on a series of 22 microcontrollers manufactured by Microchip Technology (including Atmel), that were part of different architectural families. At the end of this chapter, I described the errors that can intervene in the microcontrollers' performance measurement process.

In Chapter 6 the experimental results for the execution times of all the algorithms were presented. After comparing the execution times for all the used algorithms, measured for the same microcontroller oscillator frequency, the following conclusions were drawn:

- If the microcontrollers' operating frequency is constant, the execution times vary depending on the microcontroller and its architecture.
- The time needed to execute the test algorithms decreases as the tested microcontroller's clock frequency increases.
- There are groups of microcontrollers for which each algorithm was executed in a similar amount of time. These groups are delineated by the architecture type: PIC, 8051 and AVR.
- The compiler optimizations can influence the experimental results.
- The data and program memory sizes directly impact if a test program can run on a specific microcontroller.

In chapter 7 I designed and developed the test scenarios that target the microcontrollers' current consumption. Three test scenarios were identified. Each scenario was evaluated experimentally using specific cases and intermediary experimental results were obtained. Then, the impact of these scenarios on the microcontrollers' performance was studied. The experimental methods for evaluating a microcontroller's current consumption performance were also researched and validated on a series of specific cases. Finally, I concluded that the preferred measurement method should be using a digital multimeter configured as an ammeter.

In chapter 8 I presented the experimental results for the current consumption of the studied microcontrollers, while they execute the three test scenarios detailed in chapter 7. From the experimental results for the current consumption comparison using the same microcontroller oscillator frequency I drew the following conclusions:

- For the CORE_LOAD test scenario an increase in the current consumption was observed when compared to the IDLE scenario, for all test algorithms and all of the tested microcontrollers. In this scenario the lowest current

consumption was obtained by the PIC architecture microcontrollers (PIC16 family) and the highest current consumption was obtained by the megaAVR architecture microcontrollers.

- By analyzing the current consumption of each studied microcontroller family the following has been observed: from the PIC architecture microcontrollers the lowest current consumption was obtained by PIC16F1509 and the highest current consumption was obtained by the PIC18F2550 microcontroller. From the megaAVR family, the lowest current consumption was observed for the ATMEGA1284 microcontroller and the highest current consumption was observed for the ATMEGA32. Finally, for the 8051 architecture the lowest current consumption was achieved by the AT89S8253 microcontroller and the highest by the AT89S52 microcontroller.
- The microcontrollers' current consumption varies with the clock frequency, with the load of the central processing unit and with the usage of data transfers to or from the data and program memories.

In chapter 9 I examined the dependency of the consumed energy on the test scenarios for several microcontroller types. For each scenario and test algorithm described in chapter 7 I performed the comparison between microcontrollers that operate at the same clock frequency. I thus determined how much energy was needed for a microcontroller to execute a certain test algorithm.

By analyzing in detail, the results for the CORE_LOAD scenario I concluded that the energy required to execute the test algorithms varies for each microcontroller. Also, the clock frequency plays an important role in the energy consumption since it directly influences the algorithm execution time and the microcontrollers' current consumption. For each test algorithm the following was observed:

- For all test algorithms the energy consumption decreases as the clock frequency increases.
- For the MUL_INT algorithm the lowest energy consumption was reached by the megaAVR microcontrollers, of which ATMEGA1284 had the lowest energy consumption for all clock frequencies. The highest energy consumption was obtained by the 8051 architecture (AT89S family) for 2MHz, 4MHz and 8MHz operating frequencies. For 16MHz, PIC18F2550 had the highest energy consumption.
- For the FIND_MIN_ARRAY_FLASH_8bit algorithm the lowest energy consumption was obtained by the PIC16 microcontrollers, of which PIC16F1509 had the lowest consumption for all clock frequencies. The highest energy consumption was obtained by the 8051 microcontrollers (AT89S family) for all operating frequencies.
- Also, the PIC architecture microcontrollers (PIC16 family) had the best energy efficiency. The megaAVR microcontrollers had approximately similar efficiency. The least efficient of the tested microcontrollers were the ones with the 8051 architecture (AT89F family).

- From the clock frequency perspective, the energy consumption of the studied microcontrollers varies as the frequency changes; the current consumption increases as the clock frequency increases but the execution time for the test sequences dramatically decreases which, in turn, results in a decrease of the energy consumption. Conversely, even if a microcontroller takes longer to execute a test sequence, it can draw less current, meaning a lower amount of consumed energy.
- From the central processing unit load perspective, the studied microcontrollers' energy consumption increases as the load increases.

Considering the results obtained in this thesis we can state that its initial goal has been reached, meaning that I have identified methods to evaluate the 8-bit microcontroller performance, from the execution time, current consumption, and energy consumption perspectives. These methods were applied to a set of 8-bit microcontrollers representing the most commonly used architectures: PIC (mid-range, enhanced mid-rand and top, represented by the PIC12, PIC16 and PIC18 families), AVR architecture (megaAVR category) and 8051 architecture (AT89S and AT89LP families). The experimental results were used to analyze the performance of the tested microcontrollers and the obtained results are reproducible. Based on the experimental result the performance characteristics of all the microcontrollers included in the study were determined. Broadly, from the studied microcontrollers, the AVR architecture microcontrollers showed the best execution time performance, while the PIC architecture microcontrollers showed the best average current consumption performance and energy consumption performance.

As a conclusion, the thesis has fulfilled its initial purpose and contains a multitude of original contributions, listed in the following paragraph.

Original contributions

The original contributions to this thesis are listed below:

- Identifying the performance parameters that need to be analyzed so that they reflect real life situations and scenarios. Chapters 3, 4, 7 and papers [3] and [5].
- The selection, design, and implementation of the performance parameter measurement test algorithms 4 and 7, papers [3] and [4].
- Establishing the test methodology so that the measured parameters would not be affected by internal or external measurement errors. Chapters 5 and 7.
- Designing the automated systems for measuring and testing microcontrollers' performance. Chapters 5 and 7, papers [1], [6] and [7].
- The physical implementation of the hardware and test programs that make up the automated microcontroller performance test systems. Annexes 3 and 4.
- Extracting the execution time performance characteristics for a series of 8-bit microcontrollers, after running the experiments. Chapter 6, paper [4].

- Interpreting the experimental results obtained for the execution time and characterizing the microcontrollers based on this parameter. Chapters 6 and 10, papers [1], [4] and [5].
- Extracting the current consumption performance characteristics for a series of 8-bit microcontrollers, after running the experiments. Chapter 8, paper [2].
- Interpreting the experimental results obtained for the current consumption and characterizing the microcontrollers based on this parameter. Chapters 8 and 10, papers [1], [2] and [3].
- Extracting the energy consumption performance characteristics for a series of 8-bit microcontrollers, after running the experiments. Chapter 9.
- Interpreting the experimental results obtained for the energy consumption and characterizing the microcontrollers based on this parameter. Chapter 9.
- Performing a comparative study between the 8-bit microcontrollers used in the experiments. Chapters 6, 8, 9 and 10, annexes 1 and 2, paper [1].

List of original papers

Several original contributions from the thesis were the subject of the published articles listed below:

1. [ISI] **Al. Vlădescu**, R. Constantinescu, and D. Stoichescu, „Time and power performance study on 8-bit microcontrollers", Proc. SPIE 11718 (2020), Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies X, 20-23 August 2020, DOI:10.1117/12.2572087, Constanța, România, SPIE. Lucrarea a primit premiul „Excellent Paper Award - Poster Session".
2. [ISI] **Al. Buturugă**, R. Constantinescu, and D. Stoichescu, „Power monitoring in embedded systems using PAC1934", Electronics, Computers and Artificial Intelligence, 11th Edition, 27 June -29 June 2019, WOS:000569985400018, DOI:10.1109/ECAI46879.2019.9041967, Pitești, România, IEEE.
3. [ISI] **Al. Buturugă**, R. Constantinescu, and D. Stoichescu, „Current consumption analysis for 8-bit microcontrollers", Electronics, Computers and Artificial Intelligence, 11th Edition, 27 June -29 June 2019, WOS:000569985400005, DOI:10.1109/ECAI46879.2019.9041951, Pitești, România, IEEE.
4. [ISI] **Al. Buturugă**, R. Constantinescu, and D. Stoichescu, „A practical approach to microcontroller performance evaluation", Proc. SPIE 10977, Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies IX, 23-26 August 2018, WOS:000458717900046; DOI:10.1117/12.2324922, Constanța, România, SPIE.
5. [ISI] **Al. Buturugă**, R. Constantinescu, D. Stoichescu, „Time measurement techniques for microcontroller performance analysis", International Symposium for Design and Technology in Electronic Packaging (SIITME), 26-29 October 2017, WOS:000428032300004, DOI:10.1109/SIITME.2017.8259853, Constanta, Romania, IEEE
6. [ISI] **Al. Buturugă**, D. Stoichescu, R. Constantinescu, „Universal system for automation of test setups", Advanced Topics in Optoelectronics,

Microelectronics, and Nanotechnologies VIII, 25th – 28th August 2016, WOS:000391359600032, DOI:10.1117/12.2243244, Constanța, România, SPIE. Articolul a fost citat în lucrarea „Standalone analog active cell-balancing circuit for automotive battery management systems", B. Anton, A. Florescu, Ș.G. Rosu, Rev. Roum. Sci. Techn. Electrotechn. et Energ, Ed. Academiei Romane, 306-313, 000454752800012.

7. [ISI] **Al. Buturugă**, D. Stoichescu, R. Constantinescu, „Universal system for automation of small tasks", International Symposium on Fundamentals of Electrical Engineering (ISFEE), 30 June – 2 July 2016, WOS:000392434400009, DOI:10.1109/ISFEE.2016.7803157, Bucharest, Romania, IEEE. Articolul a fost citat în lucrarea „Standalone analog active cell-balancing circuit for automotive battery management systems", B. Anton, A. Florescu, Ș.G. Rosu, Rev. Roum. Sci. Techn. Electrotechn. et Energ, Ed. Academiei Romane, 306-313, 000454752800012.

Pe parcursul activității de cercetare, o parte din rezultate au fost incluse și în rapoartele de cercetare create pentru proiectul „Sistem de evaluare a MC-lor", din cadrul programului UPB-GEX, identificat UPB-EXCELENTA-2017, număr de contract 41/25.09.2017.

Future development perspectives

Considering the results obtained so far and the appreciation received at specialty conferences I wish to continue this research activity in three main areas:

- including more 8-bit microcontroller families,
- extending the study for 16-bit and 32-bit microcontrollers,
- analyzing more microcontroller performance parameters.

By including more 8-bit microcontroller families in the study the experimental results can be compared for a wider and more diverse range of microcontrollers. For example, in the extended study 8-bit microcontrollers manufactured by Infineon Technologies, NXP Semiconductors or Texas Instruments could be included. Expanding the study could lead to gaining a clearer picture of 8-bit microcontroller performance.

By extending the study to 16-bit and 32-bit microcontrollers we can obtain a complete overview of microcontrollers' performance in general. This is motivated by a fierce competition for 8-bit microcontrollers. 16-bit and 32-bit microcontrollers are becoming more and more varied, performant and cheaper.

Another development possibility consists of evaluating more microcontroller performance parameters. For example, the extended study could include more test sequences that would highlight additional performance parameters, thus creating an even clearer view on the microcontrollers' performance.

Bibliography

[1] Al. Vlădescu, R. Constantinescu, and D. Stoichescu, „Time and power performance study on 8-bit microcontrollers", Proc. SPIE 11718 (2020), Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies X, 20-23 August 2020, DOI:10.1117/12.2572087, Constanța, România, SPIE.

[2] Al. Buturugă, R. Constantinescu, and D. Stoichescu, „Power monitoring in embedded systems using PAC1934", Electronics, Computers and Artificial Intelligence, 11th Edition, 27 June -29 June 2019, WOS:000569985400018, DOI:10.1109/ECAI46879.2019.9041967, Pitești, România, IEEE.

[3] Al. Buturugă, R. Constantinescu, and D. Stoichescu, „Current consumption analysis for 8-bit microcontrollers", Electronics, Computers and Artificial Intelligence, 11th Edition, 27 June -29 June 2019, WOS:000569985400005, DOI:10.1109/ECAI46879.2019.9041951, Pitești, România, IEEE.

[4] Al. Buturugă, R. Constantinescu, and D. Stoichescu, "A practical approach to microcontroller performance evaluation", Proc. SPIE 10977, Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies IX, 23-26 August 2018, WOS:000458717900046; DOI:10.1117/12.2324922, Constanța, România, SPIE.

[5] Al. Buturugă, R. Constantinescu, D. Stoichescu, „Time measurement techniques for microcontroller performance analysis", International Symposium for Design and Technology in Electronic Packaging (SIITME), 26-29 October 2017, WOS:000428032300004, DOI:10.1109/SIITME.2017.8259853, Constanta, Romania, IEEE.

[6] Al. Buturugă, D. Stoichescu, R. Constantinescu, „Universal system for automation of test setups", Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies VIII, 25th – 28th August 2016, WOS:000391359600032, DOI:10.1117/12.2243244, Constanța, România, SPIE.

[7] Al. Buturugă, D. Stoichescu, R. Constantinescu, „Universal system for automation of small tasks", International Symposium on Fundamentals of Electrical Engineering (ISFEE), 30 June – 2 July 2016, WOS:000392434400009, DOI:10.1109/ISFEE.2016.7803157, Bucharest, Romania, IEEE.

[8] N. Luhmann, „Introduction to Systems Theory - 1st Edition", Ed. Polity Press, ISBN 978-0745645728, Cambridge, UK, 2013.

[9] D. A. Stoichescu, B. Florea, R.C. Constantinescu, „Echipamente electronice pentru reglaj automat", Ed. Printech, ISBN 978-606-23-0200-9, Bucuresti, 2014.

[10] L. Skyttner, „General Systems Theory – Ideas and Applications", Ebook , ISBN 978-981-4493-81-9, Gävle, Sweden, 2001.

[11] C. Dumitrache, E. Minca, F. Dragomir, O. E. Dragomir, „Teoria sistemelor automate. Fundamente teoretice și aplicații MATLAB", Ed. Matrix Rom, ISBN: 978-973-755-647-9, București, România.

[12] D. Lacamera, „Embedded Systems Architecture", Ed. Packt Publishing Limited, ISBN: 9781788832502, Birmingham, UK, 2018.

[13] E. White, „Making Embedded Systems: Design Patterns for Great Software", Ed. O'Reilly, ISBN-10 978-1449-30214-6, Sebastopol, Canada, 2012.

[14] R. Oshana, M. Kraeling, „Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications", Ed. Elsevier Science, ISBN 9780124159174, 2013.

[15] C. Burileanu, „Arhitectura microprocesoarelor”, Ed. Denix, ISBN 973-95811-1-0, București, România, 1994.

[16] J. Dürre, H. Blume, „SF3: A scalable and flexible FPGA-framework for education and rapid prototyping,” 2017 IEEE International Conference on Microelectronic Systems Education (MSE), 2017, pp. 35-38, doi: 10.1109/MSE.2017.7945080.

[17] A. Kamenov, „Digital Signal Processing for Audio Applications: Volume 1 - Formulae”, ISBN: 9780692912195, UK, 2017.

[18] S. Ghoshal, „8051 Microcontrollers, 2/e: Internals, Instructions, Programming &Interfacing”, Ed. Pearson, ISBN: 978-9332535756, London, UK, 2014.

[19] E. Williams, „Make: AVR Programming: Learning to Write Software for Hardware”, Ed. Maker Media, Inc, ISBN 9781449355784, USA, 2014.

[20] C. Hellebuyck, „Beginner's Guide to Embedded C programming”, vol. 1, Ed. Electronic Products, Charleston, ISBN 978-1438231594, 2009.

[21] R. Rădescu, „Arhitectura sistemelor de calcul – ediţia a III-a”, Editura Politehnica press, Bucureşti, 2009.

[22] P. Zahradnik, B. Šimák, "Education in real-time digital signal processing using digital signal processors," 2012 35th International Conference on Telecommunications and Signal Processing (TSP), 2012, pp. 625-628, doi: 10.1109/TSP.2012.6256372.

[23] C. Hellebuyck, „Beginner's Guide to Embedded C programming”, vol. 2, Ed. Electronic Products, Charleston, ISBN 978-1448628148, 2010.

[24] M. R. Machado, T. R. Júnior, M. R. Silva and J. B. Martins, "Smart Water Management System using the Microcontroller ZR16S08 as IoT Solution," 2019 IEEE 10th Latin American Symposium on Circuits & Systems (LASCAS), 2019, pp. 169-172, doi: 10.1109/LASCAS.2019.8667571.

[25] S. Nuratch, „A universal microcontroller circuit and firmware design and implementation for IoT-based realtime measurement and control applications”, 2017 International Electrical Engineering Congress (iEECON), 2017, pp. 1-4, doi: 10.1109/IEECON.2017.8075906.

[26] I. Rosadi, S. P. Sakti, „Low-cost wireless sensor network for small area in a building”, 2017 International Seminar on Sensors, Instrumentation, Measurement and Metrology (ISSIMM), 2017, pp. 115-118, doi: 10.1109/ISSIMM.2017.8124273.

[27] S. Sarin, H. Hindersah and A. S. Prihatmanto, „Fuzzy PID controllers using 8-Bit microcontroller for U-Board speed control”, 2012 International Conference on System Engineering and Technology (ICSET), 2012, pp. 1-6, doi: 10.1109/ ICSEngT. 2012.6339355.

[28] A. Drumea, P. Svasta, „Designing low cost embedded systems with ethernet connectivity”, 2011 IEEE 17th International Symposium for Design and Technology in Electronic Packaging (SIITME), 2011, pp. 217-220, doi: 10.1109/SIITME.2011.6102721.

[29] K. H. Pries, J. M. Quigley, „Testing Complex and Embedded Systems”, Ed. CRC Press, ISBN 9781439821404, Boca Raton, USA, 2011.

[30] P. Moharikar, J. Guddeti, „Automated test generation for post silicon microcontroller validation” 2017 IEEE International High Level Design Validation and Test Workshop (HLDVT), 2017, pp. 45-52, doi: 10.1109/HLDVT.2017.8167462.

[31] P.J. Fleming, J.J. Wallace, „How not to lie with statistics: the correct way to summarize benchmark results”, Communications of the ACM. 29 (3): 218–221. ISSN 0001-0782, DOI:10.1145/5666.5673, New York, USA.

[32] D. L. Lilja, „Measuring Computer Performance: A Practitioner's Guide", Ed. Cambridge University Press, Minneapolis, USA, ISBN 100-521-64670-7, 2005.

[33] D. Kaeli, K. Sachs, „Computer Performance Evaluation and Benchmarching", Ed. Springer, Austin, Texas, USA, ISBN 3-540-93798-6, 2009.

[34] R. Jain, The Art of Computer Systems Performance Analysis: Techniques for Experimental Design Measurement Simulation and Modeling, New York, USA, 1991.

[35] K. Kramer, T. Stolze and A. Oppelt, „Microprocessor Benchmarks - A Detailed Look at Techniques, Problems and Solutions", 2011 21st International Conference on Systems Engineering, 2011, pp. 337-340, doi: 10.1109/ICSEng.2011.67.

[36] D. You, Y. Hwang, Y. Ahn and K. Chung, „A Test Method for Power Management of SoC-based Microprocessors", 2011 12th International Workshop on Microprocessor Test and Verification, 2011, pp. 28-31, doi: 10.1109/MTV.2011.14.

[37] M. A. Abou-Of, A. H. Taha and A. A. Sedky, „Trade-off between low power and energy efficiency in benchmarking", 2016 7th International Conference on Information and Communication Systems (ICICS), 2016, pp. 322-326, doi: 10.1109/IACS.2016.7476072.

[38] Z. Nakutis, „Embedded Microcontrollers Benchmarking using Sliding Window Algorithm", Elektronika ir Elektrotechnika, 01 Jan 2007, ISSN 1392-1215, Lithuania

[39] M. Levy, „Understanding Microcontroller Performance Analysis Techniques", Digi-Key Electronics, 08 July 2011, USA.

[40] A. Weiss, „The standardization of embedded benchmarking: pitfalls and opportunities", International Conference on Computer Design, 01 February 1999, DOI: 10.1109/ICCD.1999.808586, Connecticut, USA.

[41] Z. Nakutis, „Embedded Systems Power Consumption Measurement Methods Overview", MATAVIMAI, 27 Jan 2015, ISSN 1392-1223, Lithuania.

[42] U. Pesovic, Z. Jovanovic, S. Randjic and D. Markovic, "Benchmarking performance and energy efficiency of microprocessors for wireless sensor network applications," 2012 Proceedings of the 35th International Convention MIPRO, 2012, pp. 743-747.

[43] S. Pujari, A. Yeotkar, V. Shingare, S. Momin and B. Kokare, „Performance analysis of microcontroller and FPGA based Signal Processing a case study on FIR filter design and implementation", 2015 International Conference on Industrial Instrumentation and Control (ICIC), 2015, pp. 252-257, doi: 10.1109/IIC.2015.7150748.

[44] T. Fryza and M. Waldecker, „Precise Measurement of Power Consumption and Execution Time for Embedded Platforms", 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP), 2018, pp. 1-4, doi: 10.1109/IWSSIP.2018.8439486.

[45] K. Kramer, T. Stolze and T. Banse, „Benchmarks to Find the Optimal Microcontroller-Architecture", 2009 WRI World Congress on Computer Science and Information Engineering, 2009, pp. 102-105, doi: 10.1109/CSIE.2009.928.

[46] M. Jones, J .Scott, „The Energy Efficiency of 8-bit Low-power Microcontrollers", Proceedings of the 18th Electronics New Zealand Conference, 21-22 Nov, 2011.

[47] S. Gal-On, M Levy, „Exploring CoreMark™ – A Benchmark Maximizing Simplicity and Efficacy", Embedded Microprocessor Benchmark Consortium Whitepapers, USA.

[48] J. A. Poovey, T. M. Conte, M. Levy and S. Gal-On, "A Benchmark Characterization of the EEMBC Benchmark Suite," in IEEE Micro, vol. 29, no. 5, pp. 18-29, Sept.-Oct. 2009, doi: 10.1109/MM.2009.74.

[49] T.H. Cormen, C.E. Leiserson, L.R. Rives, C. Stein, "Introduction to algorithms – Third Edition", Ed. The MIT Press, Massachusetts, ISBN 978-0-262-03384-8, 2009.

[50] H. Kaur, J. Singh, „MIPS Processor With Reduced Dynamic Power", ISSN 0976-8491, IJCST Vol. 4, Issue 1, Jan - March 2013.

[51] Microchip Technology, articol online, „What are the differences among the 8-bit PIC® MCU sub-families?", Microchip Developer Help, www.microchip.com, 2020

[52] Mikroelectronika, „Manuale utilizare compilator mikroC PRO for PIC, AVR și 8051", 2009, Serbia.

[53] Microchip Technology, Nota de aplicație AN575, F. Testa, „IEEE 754 Compliant Floating Point Routines", 1997

[54] A. Ramakrishnan, J. M. Conrad, „Analysis of floating point operations in microcontrollers", 2011 Proceedings of IEEE Southeastcon, 2011, pp. 97-100, doi: 10.1109/SECON.2011.5752913.

[55] P. Beckmann, "History of Pi", Ed. St. Martin's Press, ISBN 978-0-88029-418-8, 1989.

[56] Document ISO 1155:1978 „Information processing — Use of longitudinal parity to detect errors in information messages", Noiembrie 1978

[57] S. Henry, Warren Jr., „Hacker's Delight (1 ed.)", Ed. Addison Wesley, 2002, Boston, USA, ISBN 978-0-201-91465-8.

[58] S. M. Cheema, N. Sarwar and F. Yousaf, „Contrastive analysis of bubble & merge sort proposing hybrid approach", 2016 Sixth International Conference on Innovative Computing Technology (INTECH), 2016, pp. 371-375, doi: 10.1109/INTECH.2016.7845075.

[59] R. Edjlal, A. Edjlal and T. Moradi, „A sort implementation comparing with Bubble sort and Selection sort", 2011 3rd International Conference on Computer Research and Development, 2011, pp. 380-381, doi: 10.1109/ICCRD.2011.5763927.

[60] R. Dasgupta, „Anatomy of RTOS and Analyze the Best-Fit for Small, Medium and Large Footprint Embedded Devices in Wireless Sensor Network", 2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008), 2008, pp. 598-603, doi: 10.1109/SENSORCOMM.2008.139.

[61] Moo Kit Lee, Wei Khoon Teng, Raj Kumar Krishnasamy and Wei Tee Ng, „Delay-line based embedded memory access time measurement: Circuit, implementation and characterization techniques", 2012 4th Asia Symposium on Quality Electronic Design (ASQED), 2012, pp. 259-264, doi: 10.1109/ACQED.2012.6320512.

[62] Microchip Technology, Nota de aplicație AN953, D. Flowers, „Data Encryption Routines for the PIC18", 2005.

[63] Y. A. Nasser, M. A. Bazzoun and S. Abdul-Nabi, „AES algorithm implementation for a simple low cost portable 8-bit microcontroller", 2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC), 2016, pp. 203-207, doi: 10.1109/ICDIPC.2016.7470819.

[64] V. H. Soumya, M. B. Neelagar and K. V. Kumaraswamy, „Designing of AES Algorithm using Verilog", 2018 4th International Conference for Convergence in Technology (I2CT), 2018, pp. 1-5, doi: 10.1109/I2CT42659.2018.9058322.

[65] C. Moreno, S. Fischmeister, „Accurate Measurement of Small Execution Times-Getting Around Measurement Errors", in IEEE Embedded Systems Letters, vol. 9, no. 1, pp. 17-20, March 2017, doi: 10.1109/LES.2017.2654160.

[66] H. Wang, W. Zhou, Z. Li, S. Qian, W. Jiang, C. Wang, „A time and frequency measurement method based on delay-chain technique", 2008 IEEE International Frequency Control Symposium, 2008, pp. 484-486, doi: 10.1109/FREQ.2008.4623046.

[67] R. Szplet, P. Kwiatkowski and J. Tyburski, „Precise Time Digitizer Based on Counting Method and Multiphase In-Period Interpolation", 2019 Joint Conference of the IEEE International Frequency Control Symposium and European Frequency and Time Forum (EFTF/IFC), 2019, pp. 1-3, doi: 10.1109/FCS.2019.8856004.

[68] Microchip Technology, articol online, „Get Started With The PIC16F1 Enhanced MCU Architecture", Microchip Developer Help, www.microchip.com, 2020.

[69] Microchip Technology Inc., foi de catalog microcontroler „PIC16F1787", DS40001637C, www.microchip.com, 2012-2014.

[70] Microchip Technology Inc., foi de catalog microcontroler „PIC16F688", DS41203E, www.microchip.com, 2009.

[71] Microchip Technology Inc., foi de catalog microcontroler „PIC16F84A", DS35007C, www.microchip.com, 2001-2013

[72] Microchip Technology Inc., foi de catalog microcontroler „PIC12F675", DS41190G, www.microchip.com, 2010.

[73] Microchip Technology Inc., foi de catalog microcontroler „PIC16F1509", DS40001609G, www.microchip.com, 2011-2015.

[74] Microchip Technology Inc., foi de catalog microcontroler „PIC16F18326", DS40001839E, www.microchip.com, 2016-2019.

[75] Microchip Technology Inc., foi de catalog microcontroler „PIC16F19156", DS40001923B, www.microchip.com, 2017-2019.

[76] Microchip Technology Inc., foi de catalog microcontroler „PIC18F4525", DS39626E, www.microchip.com, 2008.

[77] Microchip Technology Inc., foi de catalog microcontroler „PIC18F2550", DS39632E, www.microchip.com, 2009.

[78] Microchip Technology Inc., foi de catalog microcontroler „PIC18F13K50", DS40001350F, www.microchip.com, 2008-2015.

[79] Microchip Technology Inc., foi de catalog microcontroler „PIC18F47K42", DS40001919E, www.microchip.com, 2017-2019.

[80] Microchip Technology Inc., articol online, „8-Bit AVR® Core" , Microchip Developer Help, www.microchip.com, 2020

[81] Microchip Technology Inc., articol online, „8-bit AVR® Microcontrollers" , Microchip Developer Help, www.microchip.com, 2020.

[82] R. Barnett, „Embedded C Programming and the Atmel AVR", Ed. Cengage Learning, ISBN: 9781418039592, USA, New York, 2003.

[83] S. Korbel, V. Janes, „Interesting applications of Atmel AVR microcontrollers", Euromicro Symposium on Digital System Design, 2004. DSD 2004., 2004, pp. 499-506, doi: 10.1109/DSD.2004.1333318.

[84] Microchip Technology Inc., foi de catalog microcontroler „ATMEGA328P", DS40002061A, www.microchip.com, 2018.

[85] Microchip Technology Inc., foi de catalog microcontroler „ATMEGA8", Rev.2486AA–AVR–02/2013, www.microchip.com, 2013.

[86] Microchip Technology Inc., foi de catalog microcontroler „ATMEGA16", Rev. 2466T–AVR–07/10, www.microchip.com, 2010.

[87] Microchip Technology Inc., foi de catalog microcontroler „ATMEGA32”, 2503Q–AVR–02/11, www.microchip.com, 2011.
[88] Microchip Technology Inc., foi de catalog microcontroler „ATMEGA1284”, DS40002070B, www.microchip.com, 2020.
[89] Intel Corporation, Notă de aplicație AP-69, J. Wharton, „An introdution to Intel MCS-51 single-chip microcontroller family”, AFN-01502A-01, Mai 1980
[90] S. Ghoshal, „8051 microcontroller internals, instructions, programming and interfacing”, Ed. Dorling Kindersley, ISBN 978-81-317-3143-7, India, 2010
[91] Microchip Technology Inc., foi de catalog microcontroler „AT89S8253”, 3286P–MICRO–3/10, www.microchip.com, 2010.
[92] Microchip Technology Inc., foi de catalog microcontroler „AT89S52”, 1919D–MICRO–6/08, www.microchip.com, 2008.
[93] Microchip Technology Inc., foi de catalog microcontroler „AT89S2051”, 3390E–MICRO–6/08, www.microchip.com, 2008.
[94] Microchip Technology Inc., foi de catalog microcontroler „AT89LP828”, 3654A–MICRO–8/09, www.microchip.com, 2009.
[95] Microchip Technology Inc., foi de catalog microcontroler „AT89LP214”, 3538E–MICRO–11/10, www.microchip.com, 2010.
[96] S. Rabinovich, „Measurement Errors and Uncertainties: Theory and Practice”, Ed. Springer, ISBN-13: 978-0387253589, Basking Ridge, USA, 2005.
[97] J. Buonaccorsi, „Measurement Error: Models, Methods, and Applications”, Ed. CRC Press, ISBN-13: 978-1420066562, Boca Raton, USA, 2010.
[98] G. Crotti, A. D. Femine, D. Gallo, D. Giordano, C. Landi and M. Luiso, „Measurement of Absolute Phase Error of Digitizers”, 2018 Conference on Precision Electromagnetic Measurements (CPEM 2018), 2018, pp. 1-2, doi: 10.1109/CPEM.2018.8501177.
[99] R. Dochia, D. Bogdan and C. Burileanu, „Model for software power estimation of an 8-bit microcontroller”, CAS 2011 Proceedings (2011 International Semiconductor Conference), 2011, pp. 443-446, doi: 10.1109/SMICND.2011.6095842.
[100] R. Dochia, D. Bogdan and C. Burileanu, „Laboratory automation for the power measurement of a microcontroller using Perl/Tk”, ISSCS 2011 - International Symposium on Signals, Circuits and Systems, 2011, pp. 1-4, doi: 10.1109/ISSCS.2011.5978690.
[101] G. Crotti et al., „Low cost measurement equipment for the accurate calibration of voltage and current transducers”, 2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014, pp. 202-206, doi: 10.1109/I2MTC.2014.6860735.
[102] Y. Kabalci and E. Kabalci, „The low cost voltage and current measurement device design for power converters”, 2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2016, pp. 1-6, doi: 10.1109/ECAI.2016.7861183.
[103] Keysight Technologies - Tektronix, Application note, „Making Accurate Current Measurements on Power Supplies with Oscilloscopes”, 2016.
[104] P. Gray, „Analysis and design of analog integrated circuits – Fourth edition”, Ed. John Wiley & Sons, New York, ISBN 0-471-32168-0, 2001.
[105] Keysight Technologies - Tektronix, Application note, „Measuring Low Current Consumption with a Digital Multimeter”, 2015.

[106] J. Gu, T. Ishihara and K. Lee, „Loop instruction caching for energy-efficient embedded multitasking processors", 2012 IEEE 10th Symposium on Embedded Systems for Real-time Multimedia, 2012, pp. 97-106, doi: 10.1109/ESTIMedia.2012.6507036.

[107] Qiang Wu, V.J. Reddi; Youfeng Wu; Jin Lee; D. Connors; D. Brooks; M. Martonosi; D.W. Clark, „A dynamic compilation framework for controlling microprocessor energy and performance", 38th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'05), 2005, pp. 12 pp.-282, doi: 10.1109/MICRO.2005.7.

[108] M. Engin, „Energy Efficiency of Embedded Controllers", 2019 8th Mediterranean Conference on Embedded Computing (MECO), 2019, pp. 1-4, doi: 10.1109/MECO.2019.8760289.

[109] M. Saadatmand, A. Cicchetti and M. Sjödin, „A methodology for designing energy-aware secure embedded systems", 2011 6th IEEE International Symposium on Industrial and Embedded Systems, 2011, pp. 87-90, doi: 10.1109/SIES.2011.5953687.

[110] E. Zulkas, E. Artemciukas, D. Dzemydiene and E. Guseinoviene, „Energy consumption prediction methods for embedded systems", 2015 Tenth International Conference on Ecological Vehicles and Renewable Energies (EVER), 2015, pp. 1-5, doi: 10.1109/EVER.2015.7112932.