

UNIVERSITATEA „POLITEHNICA” din BUCUREȘTI

ȘCOALA DOCTORALĂ ETTI-B

Nr. Decizie 619 din 23.11.2020

TEZĂ DE DOCTORAT

**DETERMINAREA ȘI EVALUAREA
EXPERIMENTALĂ A PERFORMANȚELOR
MICROCONTROLERELOR
CU ARHITECTURĂ PE 8 BIȚI**

Doctorand: Ing. Alexandru Vlădescu (Buturugă)

COMISIA DE DOCTORAT

Președinte	Prof. Dr. Ing. Gheorghe Brezeanu	de la	Univ. Politehnica București
Conducător de doctorat	Prof. Dr. Ing. Dan Alexandru Stoichescu	de la	Univ. Politehnica București
Referent	Prof. Dr. Ing. Bogdan Ionescu	de la	Univ. Politehnica București
Referent	Prof. Dr. Ing. Nicu Bizon	de la	Univ. din Pitești
Referent	Prof. Dr. Ing. Alexandru Șerbănescu	de la	Academia Tehnică Militară București

BUCUREȘTI 2021

Mulțumiri

Îmi exprim mulțumirea și recunoștința față de domnul profesor dr. ing. Dan Alexandru Stoichescu, conducătorul științific al acestei teze, pentru toată susținerea și îndrumarea acordată pe parcursul activității de cercetare.

De asemeni, doresc să mulțumesc doamnei ș.l. dr. ing. Rodica Claudia Constantinescu pentru susținerea acordată pe parcursul activității de cercetare și pentru colaborarea și coordonarea din cadrul proiectului „*Sistem de evaluare a MC-lor*”, prin programul UPB-GEX2017 și celorlalți membri din comisia de îndrumare: prof. dr. ing. Adriana Florescu și ș.l. dr. ing. Bogdan Cristian Florea.

Îi sunt recunoscător domnului ș.l. dr. ing. Valentin Pupezescu pentru ca a citit această teză, m-a încurajat și mi-a dat indicații de valoare.

Îmi exprim gratitudinea față de domnii profesori dr. ing. Gheorghe Brezeanu, dr. ing. Bogdan Ionescu, dr. ing. Nicu Bizon și dr. ing. Alexandru Șerbănescu pentru că au acceptat să facă parte din comisia mea de doctorat.

Dar nu în ultimul rând, mulțumesc familiei mele pentru tot sprijinul acordat pe toată perioada derulării studiilor.

Cuprins

Mulțumiri.....	iii
Cuprins.....	v
Lista tabelor	ix
Lista figurilor	xiii
Lista abrevierilor.....	xix
1 Introducere	1
1.1 Prezentarea domeniului de doctorat	1
1.2 Scopul tezei	2
1.3 Conținutul tezei	3
2 Noțiuni teoretice preliminare	5
2.1 Definiția generală a sistemelor și caracterizarea acestora	5
2.2 Sisteme de control automat electronice	7
2.3 Automate finite.....	8
2.4 Sistemele dedicate și tipurile de procesoare folosite la realizarea acestora	9
2.4.1 Microprocesoare de uz general	9
2.4.2 Circuite cu matrice de porți programabile (FPGA).....	9
2.4.3 Procesoare de semnal	10
2.4.4 Microcontrolere.....	10
3 Evaluarea performanțelor microcontrolerelor	13
3.1 Evaluarea sistemelor din punct de vedere al performanțelor	13
3.2 Evaluarea performanțelor microcontrolerelor în literatura de specialitate.....	14
3.3 Analiza microcontrolerelor din punct de vedere al timpului de execuție al secvențelor de program și al energiei consumate pentru execuția secvenței	15
4 Sistemul de măsurare a timpului de execuție: algoritmi de test și metodele de măsură	17
4.1 Algoritmi de test.....	17
4.1.1 Implementarea algoritmului NOP.....	18
4.1.2 Implementarea algoritmului MUL_INT	19
4.1.3 Implementarea algoritmului MUL_FLOAT	20
4.1.4 Implementarea algoritmului DIV	21
4.1.5 Implementarea algoritmului CHECKSUM8.....	22

4.1.6 Implementarea algoritmului SORT_ARRAY_8bit	23
4.1.7 Implementarea algoritmului SORT_ARRAY_16bit	24
4.1.8 Implementarea algoritmului FIND_MIN_ARRAY_RAM_8bit	25
4.1.9 Implementarea algoritmului FIND_MIN_ARRAY_RAM_16bit	26
4.1.10 Implementarea algoritmului FIND_MIN_ARRAY_FLASH_8bit	27
4.1.11 Implementarea algoritmului FIND_MIN_ARRAY_FLASH_16bit	28
4.1.12 Implementarea algoritmului AES128_ENCRYPT	29
4.1.13 Implementarea algoritmului AES128_DECRYPT	31
4.2 Algoritmii și impactul asupra performanțelor unității centrale de prelucrare	33
4.3 Metodele de măsurare a timpilor de execuție	34
4.3.1 Metoda externă	34
4.3.2 Metoda internă	36
4.4 Validarea experimentală a metodelor de măsurare a timpilor de execuție	38
4.5 Condiții de realizare a experimentelor	45
5 Sistemul de măsurare a timpului de execuție: arhitecturile unor microcontrolere și erorile de măsură	47
5.1 Prezentarea arhitecturilor microcontrolerelor studiate	47
5.1.1 Arhitectura PIC descriere și caracteristici	48
5.1.2 Arhitectura AVR, descriere și caracteristici	50
5.1.3 Arhitectura 8051, descriere și caracteristici	51
5.2 Parametrii care influențează performanța microcontrolerelor în domeniul timp ...	53
5.3 Minimizarea erorilor ce pot interveni în rezultatele experimentale	53
5.3.1 Eroarea introdusă de semnalul de ceas	54
5.3.2 Eroarea introdusă de metoda de măsurare utilizată	55
5.3.3 Eroarea introdusă de compilator și mediul de dezvoltare	55
6 Analiza comparativă a rezultatelor măsurătorilor timpului de execuție	57
6.1 Comparația rezultatelor la aceeași frecvență de ceas a microcontrolerelor	57
6.1.1 Rezultate experimentale – algoritmul MUL_INT	59
6.1.2 Rezultate experimentale – algoritmul MUL_FLOAT	62
6.1.3 Rezultate experimentale – algoritmul DIV	65
6.1.4 Rezultate experimentale – algoritmul CHECKSUM8	68
6.1.5 Rezultate experimentale – algoritmul SORT_ARRAY_8bit	71
6.1.6 Rezultate experimentale – algoritmul SORT_ARRAY_16bit	74
6.1.7 Rezultate experimentale – algoritmul FIND_MIN_ARRAY_RAM_8bit	77

6.1.8	Rezultate experimentale – algoritmul FIND_MIN_ARRAY_RAM_16bit.....	80
6.1.9	Rezultate experimentale – algoritmul FIND_MIN_ARRAY_FLASH_8bit	83
6.1.10	Rezultate experimentale – algoritmul FIND_MIN_ARRAY_FLASH_16bit...	86
6.1.11	Rezultate experimentale – algoritmul AES128_ENCRYPT	89
6.1.12	Rezultate experimentale – algoritmul AES128_DECRYPT	92
6.2	Analiza rezultatelor experimentale obținute pentru aceeași frecvență de ceas a microcontrolerului.....	95
6.3	Comparația microcontrolerelor la aceeași viteză de funcționare a unității centrale de procesare.....	96
6.4	Analiza rezultatelor experimentale obținute pentru aceeași viteză a unității centrale de prelucrare.....	103
7	Sistemul pentru determinarea consumului de curent	105
7.1	Scenariile de test folosite pentru determinarea „consumului de curent” al microcontrolerelor.....	106
7.1.1	Dezvoltarea scenariului IDLE.....	107
7.1.2	Dezvoltarea scenariului INTERNAL_MODULE.....	108
7.1.3	Dezvoltarea scenariului CORE_LOAD	109
7.2	Metodele de măsurare a curentului absorbit de microcontrolerul testat.....	111
7.2.1	Metoda de măsurare a curentului cu ajutorul unui osciloscop digital.....	111
7.2.2	Metoda de măsurare a curentului cu ajutorul unui multimetru digital.....	113
7.2.3	Măsurarea curentului cu ajutorul unui circuit integrat specializat	113
7.3	Validarea experimentală a scenariilor de test și a metodelor de măsurare a consumului de curent	114
7.3.1	Validarea experimentală a scenariului IDLE	114
7.3.2	Validarea experimentală a scenariului INTERNAL_MODULE	118
7.3.3	Validarea experimentală a scenariului CORE_LOAD	121
8	Rezultate experimentale obținute în urma măsurării consumului de curent.....	126
8.1	Descrierea experimentelor pentru măsurarea consumului de curent.....	127
8.2	Parametrii care influențează consumul de curent.....	128
8.3	Erorile ce pot interveni în rezultatele experimentale și minimizarea acestora.....	128
8.4	Comparație între rezultatele măsurătorilor curentului prin mai multe microcontrolere în cazul în care frecvența semnalului de ceas este aceeași	130
8.5	Comparație între rezultatele măsurătorilor consumului de curent al mai multor microcontrolere pentru aceeași viteză de prelucrare a unității centrale	144

9 Consumul de energie al microcontrolerelor.....	154
9.1 Sursele de erori ce pot interveni în rezultatele experimentale și minimizarea acestor	155
9.2 Rezultate experimentale consum de energie.....	155
10 Concluzii	164
10.1 Rezultate obținute.....	164
10.2 Contribuții originale	172
10.3 Lista lucrărilor originale	173
10.4 Perspective de dezvoltare ulterioară.....	175
Anexe	176
A.1. Timpul de execuție al microcontrolerelor – rezultate experimentale brute.....	176
A.2. Consumul de curent al microcontrolerelor – rezultate experimentale brute	188
A.3. Implementarea practică a montajelor de test pentru experimente	191
A.4. Implementarea programelor de test pentru experimente	197
Bibliografie	206

Capitolul 1

Introducere

Lucrarea de față se plasează în domeniul evaluării microcontrolerelor cu arhitectură pe 8 biți. Sistemele dedicate vizează rezolvarea anumitor operații în mod automat și se dorește ca aceste operații să fie executate într-un timp cât mai scurt și utilizând o cantitate cât mai mică de energie electrică.

Un element care influențează performanța unui sistem dedicat este elementul programabil care controlează întregul sistem. De cele mai multe ori acest element este un microcontroler. Această lucrare se concentrează pe evaluarea performanțelor microcontrolerelor cu arhitectură pe 8 biți.

Microcontrolerele cu arhitectură pe 8 biți sunt foarte răspândite în majoritatea aplicațiilor din industrie, iar în anul 2019 piața globală a microcontrolerelor cu arhitectură pe 8 biți a fost evaluată la aproximativ 8 miliarde de dolari și se află în continuă creștere datorită cererii mari din domeniul automobilelor și din domeniul IoT. Cu toate acestea, domeniul microcontrolerelor cu arhitectură pe 8 biți este mai puțin explorat în articolele de specialitate, mai ales din punct de vedere al performanței în domeniul timpului de execuție, consumului de curent și consumului de energie.

Scopul acestei lucrări este de a descrie o nouă metodologie pentru testarea microcontrolerelor din punct de vedere al timpului de execuție și al curentului absorbit și de a efectua un studiu comparativ între diverse microcontrolere. Studiul comparativ are ca scop analiza performanțelor microcontrolerelor și nu analiza diverselor programe folosite, cum ar fi mediul integrat de dezvoltare, compilatoarele sau orice alt program pentru calculator ajutător.

Pe parcursul lucrării sunt prezentate toate etapele necesare pentru a obține rezultate experimentale reproductibile, începând cu noțiunile teoretice necesare pentru înțelegerea și interpretarea rezultatelor experimentale și continuând cu descrierea metodologiei și a sistemului automat de testare a performanțelor, din punct de vedere hardware și software, și terminând cu analiza și interpretarea rezultatelor experimentale. Sunt folosite mai multe microcontrolere cu arhitectură pe 8 biți și este efectuat un studiu comparativ între acestea.

Teza este alcătuită din zece capitole dintre care unul introductiv și unul de concluzii. La finalul lucrării există mai multe anexe și o bibliografie. În introducere sunt prezentate: domeniul lucrării de doctorat, scopul ei și structura lucrării, cu descrierea fiecărui capitol.

Teza continuă cu capitolul 2, destinat noțiunilor teoretice necesare pentru înțelegerea în detaliu a domeniului tezei și a rezultatelor experimentale obținute pe parcursul cercetării. În capitolul 3 este prezentat un studiu despre analiza în literatura de specialitate a performanței microcontrolerelor.

Capitolul 4 conține selectarea, adaptarea și proiectarea algoritmilor de test ce au ca scop utilizarea la maxim a unității centrale de prelucrare. Algoritmii sunt evaluați și este

estimat impactul lor asupra funcționării microcontrolerelor. Analiza continuă cu explorarea metodelor de evaluare a performanței în domeniul timp pentru microcontrolere. Metodele de măsurare a timpului sunt validate experimental.

În capitolul 5 este prezentată metodologia de testare a performanțelor microcontrolerelor din punct de vedere al timpului de execuție. Sunt descrise caracteristicile arhitecturilor microcontrolerelor alese pentru a fi studiate și anume o serie de microcontrolere produse de compania Microchip Technology (inclusiv Atmel), ce fac parte din familii arhitecturale diferite (PIC, AVR și 8051). De asemenea, la finalul acestui capitol sunt descrise și erorile ce pot să intervină în procesul de analiză a performanței microcontrolerelor testate.

În capitolul 6 sunt prezentate rezultatele experimentale obținute pe o serie de microcontrolere pe 8 biți ce fac parte din familiile arhitecturale descrise în capitolul 5. Pentru fiecare algoritm de test descris în capitolul 4, sunt realizate două tipuri de comparații între microcontrolere: comparația la aceeași frecvență de funcționare a oscilatorului microcontrolerului și comparația la aceeași viteză a unității centrale de prelucrare. În acest capitol sunt prezentate rezultatele experimentale pentru timpii de execuție ai tuturor algoritmilor prezentați. Rezultatele sunt grupate după algoritmul de test pentru a permite o comparație mai ușoară între microcontrolere.

Capitolul 7 reflectă cercetarea și proiectarea scenariilor de test ce au ca scop măsurarea consumului de curent al microcontrolerelor studiate. Aceste scenarii sunt evaluate experimental pe cazuri particulare și apoi este estimat impactul lor asupra microcontrolerelor. Analiza continuă cu explorarea metodelor de evaluare a performanței microcontrolerelor din punct de vedere al consumului de curent. Aceste metode de măsurare a consumului de curent sunt validate experimental pe o serie de cazuri particulare.

În capitol 8 sunt prezentate rezultatele obținute în urma experimentelor destinate măsurării consumului de curent pentru microcontrolerele care execută trei scenarii de test prezentate în capitolul 7. De asemenea, sunt furnizate mai multe detalii despre metodologia de testare, despre condițiile în care sunt realizate experimentele și despre erorile ce pot surveni în procesul de măsurare a curentului. Pentru fiecare secvență de test descrisă în capitolul 7 sunt realizate două tipuri de comparații între microcontrolere: la aceeași frecvență de funcționare a oscilatorului microcontrolerului și la aceeași viteză a unității centrale de prelucrare din microcontroler. Rezultatele sunt grupate după scenariul de test pentru a permite o comparație ușoară între microcontrolere.

În capitolul 9 sunt prezentate rezultatele obținute în urma evaluării energiei electrice consumate de microcontrolere. Analiza este efectuată folosind microcontrolere care execută o serie de scenarii de test. De asemenea, sunt furnizate detalii despre erorile ce pot interveni în procesul de evaluare a consumului de energie. Pentru fiecare scenariu și algoritm de test descrise în capitolul 7 este realizată comparația între microcontrolere ce funcționează la aceeași frecvență de ceas.

Teza se încheie cu capitolul 10, în care sunt prezentate concluziile cercetării. În acest capitol sunt prezentate rezultatele obținute, contribuțiile originale, lista lucrărilor originale pe care se bazează această lucrare și, de asemenea, perspectivele de dezvoltare ulterioară.

Capitolul 2

Noțiuni teoretice preliminare

Un microcontroler (μC) poate fi privit ca un mic sistem de calcul pe un singur circuit integrat constând dintr-o unitate centrală de procesare (CPU), combinată cu dispozitive periferice, cum ar fi memorii, dispozitive de intrare/ieșire (I/O), magistrale, numărătoare (timere), convertoare analog-digitale și digital-analogice etc. Este utilizat conceptul de System on Chip (SoC). Din punct de vedere al arhitecturii, microcontrolerele se pot încadra în mai multe categorii; acestea sunt listate pe scurt mai jos și prezentate mai detaliat în capitolul 5.

Spre deosebire de microprocesoare, microcontrolerele sunt specializate în detectarea și tratarea întreruperilor provenite fie din exterior, de exemplu schimbarea stării unui pin sau recepționarea unui cuvânt de date prin interfața serială (ex UART), fie din interior, cum ar fi de la un modul numărător.

Deși puterea lor de calcul este mult mai mică decât a unor microprocesoare cu arhitectura x86, microcontrolerele sunt preferate deoarece costul de producție și timpul de dezvoltare a sistemelor sunt considerabil mai mici.

Există mai multe familii de microcontrolere, cum ar fi: de 8 biți, de 16 biți și de 32 biți. Ele pot consuma cantități extrem de mici de energie și, de multe ori, includ și funcția „sleep” (somm), în care consumul de energie poate fi redus până la nivel de câțiva nanoWați. Această funcție a făcut ca microcontrolerele să fie foarte populare în aplicații cum ar fi nodurile rețelelor de senzori wireless sau dispozitivele de supraveghere, deoarece, de obicei, sunt alimentate de la o baterie care trebuie să reziste cât mai mult timp.

Capitolul 3

Evaluarea performanțelor microcontrolerelor

Pentru verificarea parametrilor unui sistem dedicat se pot crea și folosi teste. Testele pentru parametrii funcționali validează dacă un parametru a fost implementat și funcționează așa cum s-a intenționat. Testele pentru parametrii non-funcționali, validează că un parametru se află în limitele așteptate. De exemplu, rapiditatea în execuție și

eficiența energetică sunt doi parametri non-funcționali care pot fi măsurați și validați folosind teste non-funcționale de performanță.

Analiza performanței unui microcontroler se mai numește și analiză comparativă sau *benchmarking*. Luând ca exemplu un calculator personal, conform [31], analiza comparativă constă în rularea unui program sau a unui set de programe care realizează anumite operații pe calculator, cu ajutorul cărora se poate analiza performanța microprocesorului din calculator.

Dintr-o altă perspectivă, am putea considera că *benchmarking* este echivalent cu testarea non-funcțională de performanță urmată de studiul comparativ al mai multor rezultate pentru diferite microcontrolere. Cei mai folosiți parametri ce caracterizează performanța unui microcontroler sunt timpul necesar pentru a executa o anumită secvență de program și cantitatea de energie consumată pentru a executa o anumită secvență de program.

În Figura 3.1 este reprezentată grafic o secvență generică utilizată pentru analiza performanțelor. Pe axa OX regăsim timpul necesar execuției unei secvențe de program care este special aleasă pentru a putea crea un grad de utilizare cât mai mare, ideal 100%. Pe axa OY se regăsește gradul de încărcare, sau gradul de utilizare al unității centrale de prelucrare, care poate fi măsurat între cele două stări „Idle” (utilizare 0%) și „Full load” (utilizare 100%).

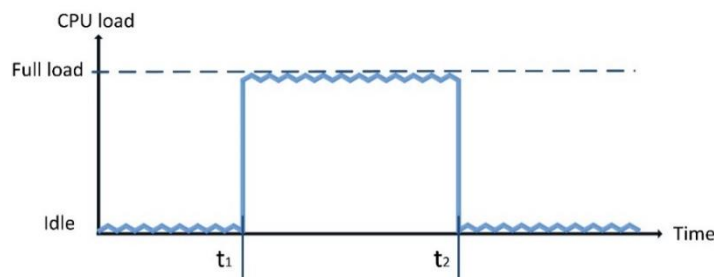


Figura 3.1 Timp de execuție și gradul de utilizare a unității centrale de prelucrare ^[5]

$$t_{executie} = t_2 - t_1 \quad (3.1)$$

Cu alte cuvinte, se pot măsura parametri de performanță folosindu-ne de timpul de execuție al secvenței de program, $t_{executie}$; $t_{executie}$ este un parametru de performanță și este calculat ca fiind diferența dintre momentul de timp când execuția se termină și momentul în care execuția secvenței de program a început. Aplicând acest concept, se poate extrage și parametrul de performanță ce descrie cantitatea de energie consumată.

În literatura de specialitate există un număr însemnat de articole și cărți ce tratează evaluarea performanțelor microprocesoarelor și calculatoarelor personale, la modul general [31], [32], [33]. Există și articole referitoare la tema acestei lucrări, dar ele nu sunt reprezentative.

În urma cercetărilor mele, am identificat și o serie de articole ce abordează performanța microcontrolerelor în diverse situații și scenarii specifice [38], [39] și [40]. Ele nu abordează cazul particular al microcontrolerelor cu arhitectură pe 8 biți ci se concentrează pe microcontrolere cu arhitecturi pe 16 biți și/sau pe 32 biți.

Pentru măsurarea timpilor de execuție și a curentului consumat sunt necesare o serie de instrumente. Acești parametri pot fi măsurați și determinați cu ajutorul unui sistem automat de testare. Un astfel de sistem nu există, însă, gata proiectat și realizat și chiar

dacă ar exista, nu ar putea fi generalizat pentru mai multe familii de microcontrolere cu arhitectură pe 8 biți.

Pentru a evidenția nevoia de instrumente și sisteme de testare a performanței microcontrolerelor am publicat o serie de articole ce tratează atât testarea performanțelor microcontrolerelor din punct de vedere al timpilor de execuție, cât și din punct de vedere al consumului de energie: [1], [2], [3], [4], [5], [6], [7]

Capitolul 4

Sistemul de măsurare a timpului de execuție: algoritmi de test și metodele de măsură

În acest capitol sunt analizați și prezentați o serie de algoritmi utilizabili în studiul performanțelor microcontrolerelor. Pentru fiecare algoritm sunt prezentate o serie de informații: descrierea, scopul și implementarea lor în limbajul de programare C.

În urma cercetării efectuate, au fost identificați mai mulți algoritmi software pentru sistemul de evaluare a microcontrolerelor. Ei sunt implementați ca o secvență de test într-un program de test și sunt folosiți pentru extragerea parametrilor de performanță ai microcontrolerelor; sunt concepuți și implementați în special pentru a nu fi dependenți de un microcontroler anume, adică sunt implementați într-un mod cât mai generic pentru a permite execuția lor pe oricare microcontroler de testat.

Programul de test reprezintă totalitatea codului C scris pentru un anumit microcontroler, iar secvența de test reprezintă setul de instrucțiuni ce formează un algoritm care are scopul de a crea încărcarea unității centrale de prelucrare din microcontroler, până la utilizare 100%. Parametrul de performanță ce se dorește a fi măsurat este timpul necesar execuției secvenței de test.

Algoritmii identificați în urma cercetării efectuate sunt:

- 4.1.1 algoritmul NOP
- 4.1.2 algoritmul MUL_INT
- 4.1.3 algoritmul MUL_FLOAT
- 4.1.4 algoritmul DIV
- 4.1.5 algoritmul CHECKSUM8
- 4.1.6 algoritmul SORT_ARRAY_8bit
- 4.1.7 algoritmul SORT_ARRAY_16bit
- 4.1.8 algoritmul FIND_MIN_ARRAY_RAM_8bit
- 4.1.9 algoritmul FIND_MIN_ARRAY_RAM_16bit

4.1.10 algoritmul FIND_MIN_ARRAY_FLASH_8bit

4.1.11 algoritmul FIND_MIN_ARRAY_FLASH_16bit

4.1.12 algoritmul AES128_ENCRYPT

4.1.13 algoritmul AES128_DECRYPT

În continuare sunt prezentați în detaliu doi dintre algoritmi, algoritmul MUL_INT și algoritmul FIND_MIN_ARRAY_RAM_8bit.

Algoritmul *MUL_INT* execută în mod repetat un set de operații de înmulțire a unor numere întregi. Algoritmul efectuează multiplicări succesive calculând expresia:

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

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

Acest număr (x) este ales special pentru a putea fi reprezentat folosind un tip de date fără semn pe 32 biți. Valoarea maximă fără semn ce poate fi stocată pe 32 biți este $2^{32} - 1 = 4.294.967.295$. În cazul compilatoarelor produse de compania Mikroelektronika se utilizează tipul de date „unsigned long”.

Algoritmul are ca scop încărcarea cât mai mare și utilizarea cât mai intensă a unității centrale de prelucrare din microcontroler. Prin execuția acestui algoritm putem să obținem informații despre capacitatea unui microcontroler de a procesa operații aritmetice intensive, folosind numere întregi.

Implementarea software în limbaj C este prezentată mai jos:

```
void RunTestMulInt(){
    int i;
    UINT32 x = 3;
    TP_1 = 1; //semnalizează începerea testului
    for(i = 0; i < 19; i++){
        x = x * 3;
    }
    TP_1 = 0; //semnalizează finalizarea testului
}
```

Algoritmul FIND_MIN_ARRAY_RAM_8bit constă în căutarea elementului cu valoare minimă dintr-un șir de date (vector) generat în mod aleatoriu, cu 64 de elemente de dimensiune 8 biți și stocat în memoria RAM. Șirul de date este generat tot timpul cu aceleași elemente pentru a putea obține rezultate consistente și reproductibile. Valoarea minimă este căutată parcurgând vectorul și comparând elementele cu valoarea minimă găsită anterior. În cazul acestui algoritm, șirul de date stocat în memoria de date.

Acest algoritm necesită o cantitate de memorie RAM (*MemDate*) de 64 octeți pentru stocarea șirului în care se face căutarea minimului. O posibilă problemă pentru microcontrolerele de generație mai veche este imposibilitatea rulării algoritmului deoarece nu dispun de suficientă memorie RAM.

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

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

Algoritmul are ca scop încărcarea cât mai mare și utilizarea cât mai intensă a unității centrale de prelucrare din microcontroler. Prin execuția algoritmului de căutare a elementului cu valoare minimă putem obține indicații despre capacitatea unui microcontroler de a lucra cu bucle de execuție și de a parcurge un șir de date stocat în memoria RAM. De asemenea, acest algoritm ajută și la evaluarea timpilor de acces pentru

Sistemul de măsurare a timpului de execuție: algoritmi de test și metodele de măsură

citire de date din memoria RAM atunci când se lucrează cu tipuri de date cu dimensiunea de 8 biți. Implementarea software în limbaj C este prezentată mai jos:

```
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; //semnalizează începerea testului
    dummyVar = arrayFindMin(array8bit,64);
    TP_1 = 0; //semnalizează finalizarea testului
}
```

Algoritmii software pentru sistemul de evaluare a microcontrolerelor prezentați până acum sunt rezultatul cercetărilor mele. Au fost identificați mai mulți algoritmi, fiecare cu particularitățile proprii.

Pentru fiecare algoritm a fost prezentat: scopul, utilizarea, implementarea și parametrul de performanță ce poate fi extras în urma execuției algoritmului respectiv ca parte a secvenței de test. În continuare sunt prezentați algoritmi clasificați după parametrii de performanță pe care îi pot evidenția:

- Prin execuția algoritmului *MUL_INT* putem să obținem informații despre capacitatea unui microcontroler de a procesa operații aritmetice intensive. În funcție de abilitatea unității aritmetică-logică de a efectua operații se vor putea observa diferențe de performanță între microcontrolerle testate.
- Prin execuția algoritmului *FIND_MIN_ARRAY_RAM_8bit* putem obține indicații despre capacitatea unui microcontroler de a lucra cu bucle de execuție ce utilizează date din memoria RAM citite de la adrese consecutive. În funcție de magistrala de date și de timpul de acces pentru citire a memoriei de date, se vor putea observa diferențe de performanță între microcontrolerle testate.

Conform [5], am identificat două posibile metode pentru a măsura timpii de execuție în cadrul microcontrolerelor: metoda internă și metoda externă. În acest capitol sunt explorate cele două metode de măsurare a timpilor de execuție și sunt aplicate practic pe o serie de cazuri particulare pentru a putea determina avantajele și dezavantajele lor.

Pe parcursul acestei lucrări am preferat metoda externă, deoarece necesită o complexitate mai mică a programului de test și instrumentele externe au o acuratețe foarte bună și rezultatele pot fi verificate direct de calculator.

Capitolul 5

Sistemul de măsurare a timpului de execuție: arhitecturile unor microcontrolere și erorile de măsură

În acest capitol este prezentată metodologia de măsurare a timpului de execuție al microcontrolerelor. Pentru a putea înțelege și interpreta mai bine rezultatele experimentale sunt descrise particularitățile arhitecturilor microcontrolerelor alese; o serie de microcontrolere produse de compania Microchip Technology (inclusiv Atmel), ce fac parte din familii arhitecturale diferite.

Microcontrolerele testate au fost alese astfel încât să acopere o gamă cât mai largă din portofoliul de microcontrolere pe 8 biți.

Aceste microcontrolere pot fi clasificate în trei categorii de arhitecturi total diferite: arhitectura PIC, arhitectura AVR și arhitectura 8051. Înainte de a prezenta rezultatele experimentale, au fost descrise cele trei arhitecturi pentru a identifica elementele arhitecturale ce pot influența performanțele și pentru a avea o înțelegere mai bună a rezultatelor experimentale.

Caracteristicile arhitecturale ce pot influența performanța de timp a microcontrolerelor sunt:

- complexitatea setului de instrucțiuni,
- timpul necesar execuției unei instrucțiuni,
- lățimea magistralei de date,
- lățimea magistralei de program,
- modurile de adresare pentru accesul la date și program,
- timpul necesar accesului în memoria de date,
- timpul necesar accesului în memoria de program,
- complexitatea unității aritmetică-logică,
- frecvența semnalului de ceas / viteza de funcționare.

Rezultatele experimentale pot fi influențate de unul din parametrii de mai sus sau, de cele mai multe ori, de o combinație a parametrilor menționați. Acest lucru se întâmplă deoarece un microcontroler este un minisistem de calcul format prin integrarea mai multor componente și module strâns interconectate.

Capitolul 6

Analiza comparativă a rezultatelor măsurătorilor timpului de execuție

În acest capitol sunt prezentate rezultatele măsurării timpilor de execuție ai microcontrolerelor selectate pentru algoritmi prezentați anterior. Pentru fiecare algoritm de test descris se realizează două comparații între microcontrolere: la aceeași frecvență a semnalului de ceas dat de oscilatorul microcontrolerului și la aceeași viteză a unității centrale de prelucrare. Rezultatele sunt grupate după algoritmul de test. Rezultatele experimentale grupate după microcontrolerul testat sunt prezentate în anexa 3.

Frecvența semnalului de ceas apare în foile de catalog ca o valoare maximă la care microcontrolerul poate funcționa. Experimentele au fost realizate la cinci frecvențe distincte pentru semnalul de ceas: 1 MHz, 2 MHz, 4 MHz, 8 MHz și 16 MHz.

Rezultatele experimentale obținute pentru fiecare algoritm sunt prezentate în tabelele 6.1 – 6.12 și în mai multe figuri. Figurile prezintă rezultatele experimentale pentru toate microcontrolerile și pentru toate frecvențele de funcționare în situația în care optimizările compilatoarelor sunt setate pe nivel zero sau pe nivel patru.

Tabelul 6.1 MUL_INT – valorile timpului de execuție pentru diferite frecvențe ale semnalului de ceas

MUL_INT Fosc Microcontroler	Timp pentru optimizarea nivel zero [ms]					Timp pentru optimizarea nivel patru [ms]				
	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

După interpretarea rezultatelor experimentale, se pot afirma următoarele:

- timpul necesar execuției algoritmului MUL_INT scade odată cu creșterea frecvenței semnalului de ceas pentru toate microcontrolerile testate;
- există grupuri de microcontrolere pentru care algoritmul MUL_INT este executat într-un timp similar;

- cele mai bune rezultate au fost obținute de microcontrolerelor cu arhitectura AVR, iar cele mai slabe rezultate au fost obținute de microcontrolerelor cu arhitectura PIC; între cele două grupuri de microcontrolere există un raport de performanță de aproximativ 1/30 în favoarea microcontrolerelor AVR;
- rezultatele obținute pentru optimizare nivel zero sunt similare cu rezultatele obținute pentru optimizare nivel patru, pentru toate microcontrolerelor studiate;

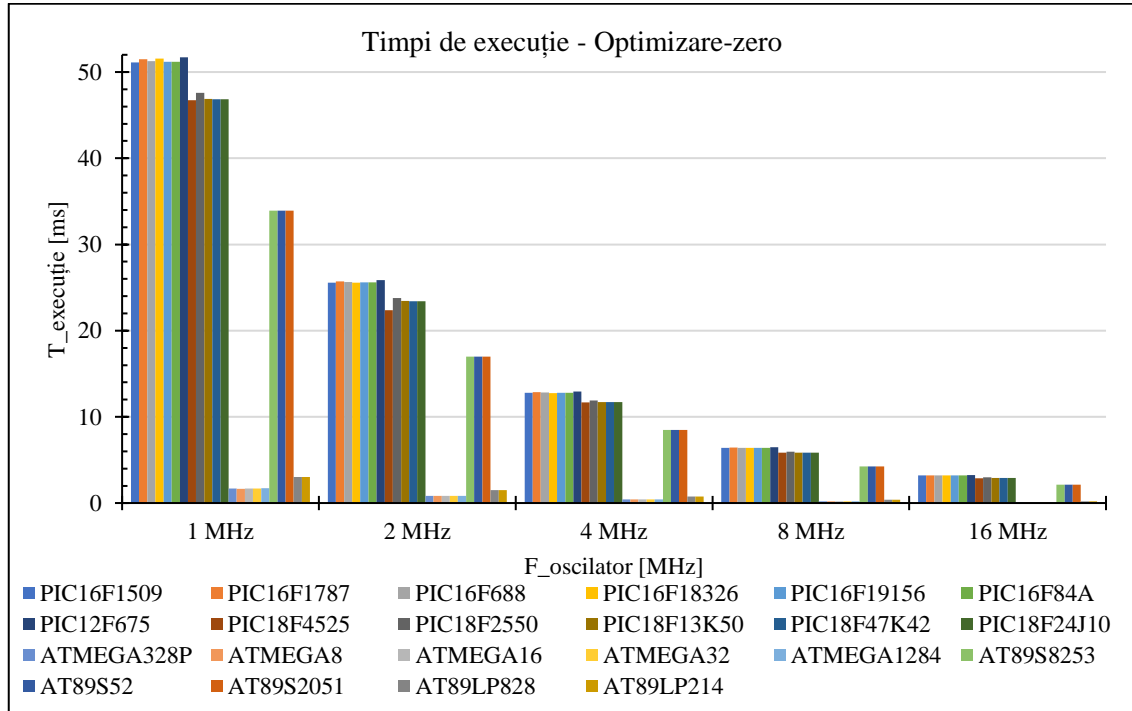


Figura 6.1 MUL_INT – dependența timpului de execuție de frecvența semnalului de ceas pentru optimizarea nivel zero

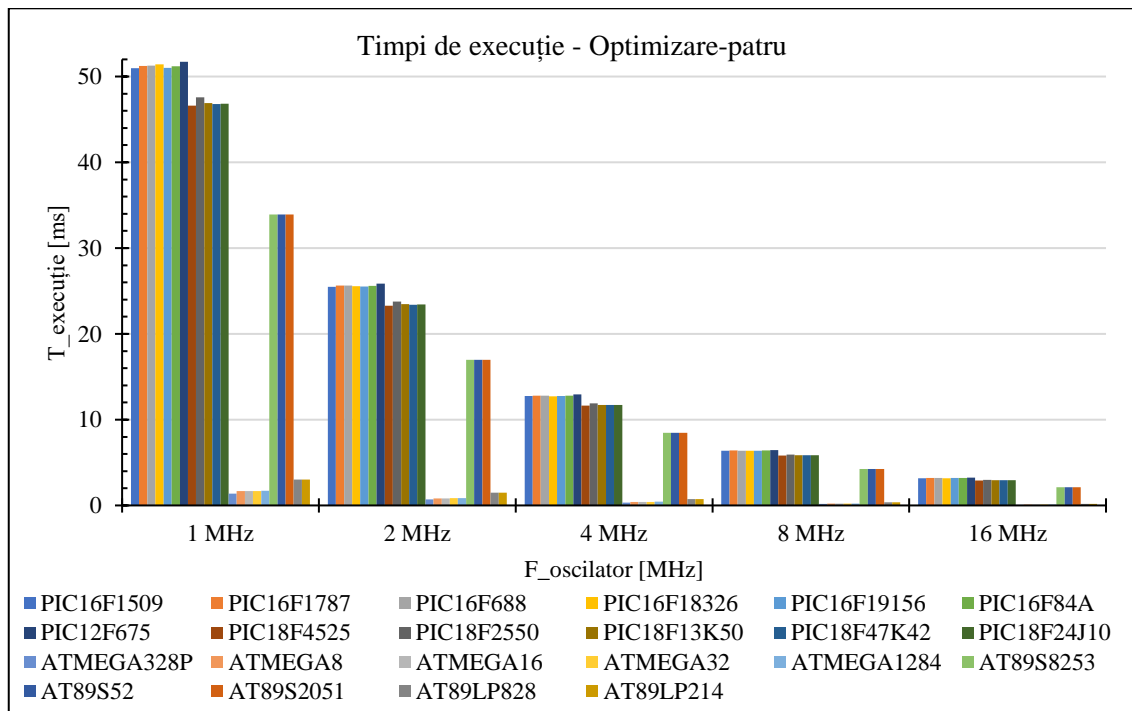


Figura 6.2 MUL_INT – dependența timpului de execuție de frecvența semnalului de ceas pentru optimizarea nivel patru

Tabelul 6.2 FIND_MIN_ARRAY_RAM_8bit – valorile timpului de execuție pentru diferite frecvențe ale semnalului de ceas

FIND_MIN_ARRAY RAM_8bit Fosc Microcontroler	Timp pentru optimizarea nivel zero [ms]					Timp pentru optimizarea nivel patru [ms]				
	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

După interpretarea rezultatelor experimentale, se pot face următoarele afirmații:

- timpul necesar execuției algoritmului FIND_MIN_ARRAY_RAM_8bit scade odată cu creșterea frecvenței semnalului de ceas pentru toate microcontrolerele testate;
- se remarcă existența unor grupuri de microcontrolere pentru care algoritmul FIND_MIN_ARRAY_RAM_8bit este executat într-un timp similar;
- cele mai bune rezultate au fost obținute de microcontrolerele cu arhitectura megaAVR, iar cele mai slabe rezultate au fost obținute de microcontrolerele cu arhitectura 8051 (familia AT89S); între cele două grupuri de microcontrolere există un raport de performanță de aproximativ 1/10 în favoarea microcontrolerelor cu arhitectură megaAVR.
- rezultatele obținute pentru optimizare nivel zero sunt similare cu rezultatele obținute pentru optimizare nivel patru, pentru microcontrolerele cu arhitectura PIC și 8051, dar pentru microcontrolerele cu arhitectura AVR se observă o creștere a performanțelor atunci când se utilizează optimizare de nivel patru;
- algoritmul nu a putut fi rulat pe microcontrolerele PIC16F84A și PIC12F675 din cauza capacității reduse a memoriei RAM.

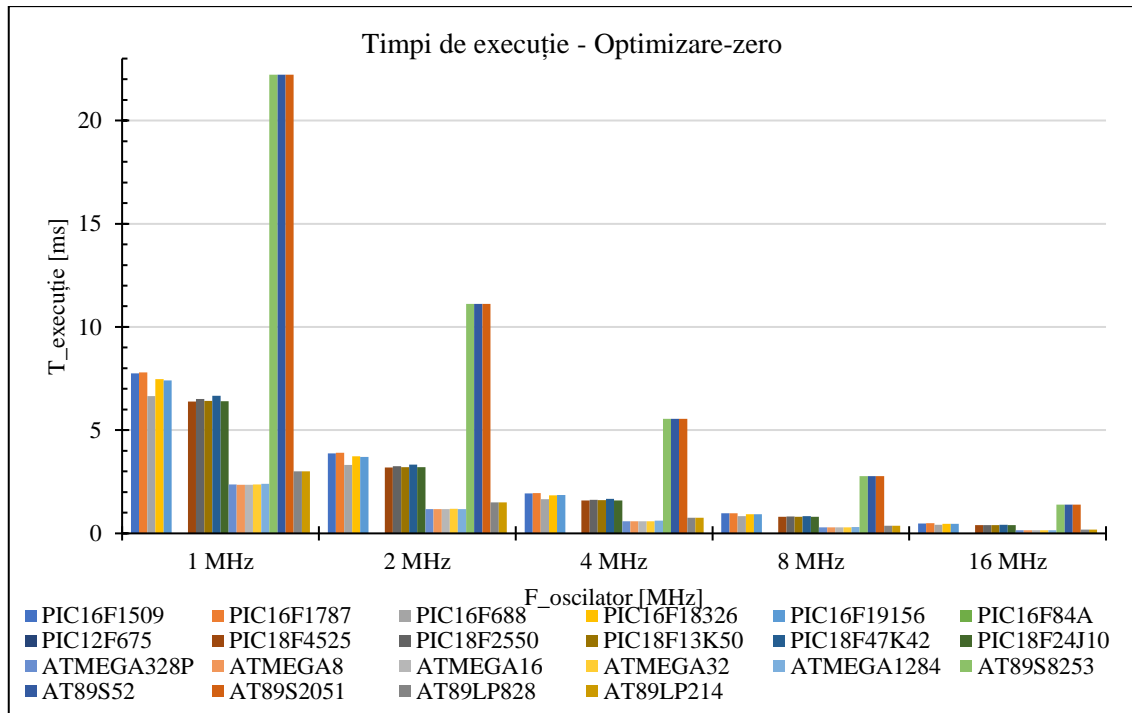


Figura 6.19 FIND_MIN_ARRAY_RAM_8bit – dependența timpului de execuție de frecvența semnalului de ceas pentru optimizarea nivel zero

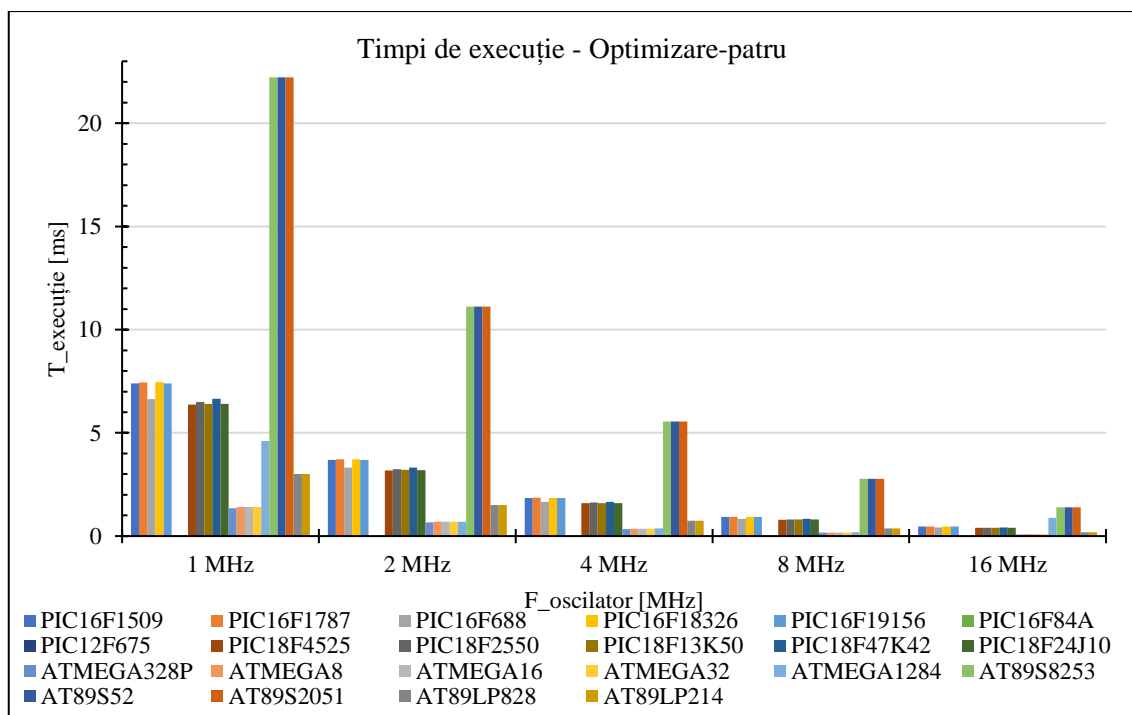


Figura 6.20 FIND_MIN_ARRAY_RAM_8bit – dependența timpului de execuție de frecvența semnalului de ceas pentru optimizarea nivel patru

Capitolul 7

Sistemul pentru determinarea consumului de curent

Parametrul de performanță ce se dorește a fi măsurat în acest capitol este curentul consumat de microcontroler. Un microcontroler poate consuma mai mult sau mai puțin curent, în funcție de configurația lui și de instrucțiunile pe care le execută într-un interval de timp.

Conform [1] și [3] consumul de curent al unui microcontroler poate fi influențat de mai mulți parametri și poate fi studiat experimental folosind mai multe scenarii bazate pe algoritmi. Aceste scenarii sunt implementate în secvențe de test.

Figurile 7.2 și 7.3 prezintă scenarii teoretice pentru consumul de curent al unui microcontroler ce execută o serie de secvențe de test diferite (ST0, ST1, ST2, etc). Se așteaptă ca desfășurarea în timp a secvențelor de test să producă o serie de variații ale consumului de curent al microcontrolerului ce le execută. Pentru secvențe de test diferite se așteaptă măsurarea unor consumuri de curent diferite. Desigur, pentru aceeași secvență de test se așteaptă un consum de curent aproximativ constant.

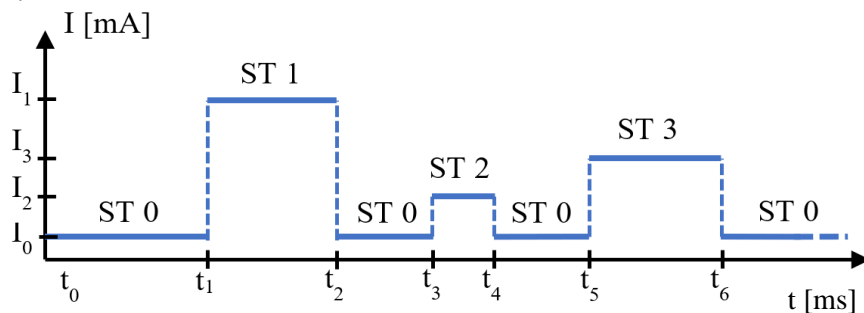


Figura 7.1 Consum de curent microcontroler – secvențe de test multiple

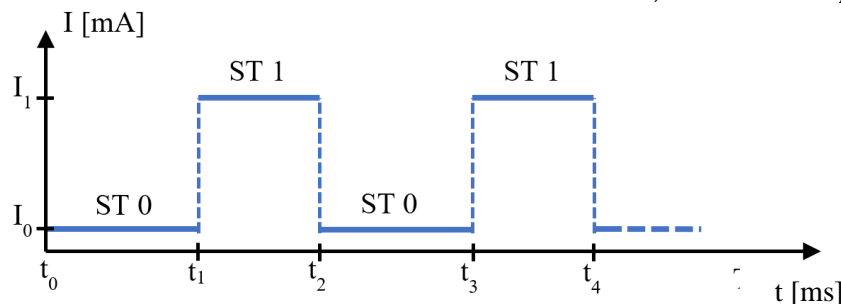


Figura 7.2 Consum de curent microcontroler – două secvențe de test

Pentru a determina experimental curentul consumat de microcontroler pe parcursul execuției unei secvențe de test, este necesară măsurarea valorilor I_0 și I_1 . Formulele (7.1)

și (7.2) dau consumul de curent al microcontrolerului în timpul execuției unei secvențe de test (I_{ST1}) și timpul necesar execuției unei secvențe de test (t_{ST1}). Desigur, formulele sunt valide dacă curentul consumat de microcontroler în timpul secvențelor de test este aproximativ constant.

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

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

În urma cercetării efectuate, scenariile de test identificate pentru evaluarea experimentală a consumului de curent al microcontrolerelor sunt:

7.1.1 scenariul IDLE

7.1.2 scenariul INTERNAL_MODULE

7.1.3 scenariul CORE_LOAD

În acest capitol sunt prezentate scenariile de test în detaliu. Pentru fiecare scenariu de test se prezintă algoritmi împreună cu scopul, utilizarea, implementarea și parametrul de performanță ce poate fi extras.

Scenariul *CORE_LOAD* folosește algoritmi ce creează o încărcare cât mai mare a unității centrale de procesare din microcontroler. Ei au fost prezentați în capitolul 4.1.

Acest scenariu se poate utiliza pentru a obține informații despre consumul de curent al microcontrolerului atunci când acesta execută o serie de instrucțiuni. Aceste instrucțiuni fac parte din algoritmi utilizați pentru a evalua performanța în timp a microcontrolerelor. În cadrul acestui scenariu se pot obține rezultatele experimentale ce arată curentul necesar unui microcontroler pentru a executa un algoritm.

Algoritmii *MUL_INT* și *DIV* au fost aleși deoarece, prin execuția acestor algoritmi, se pot extrage indicații despre consumul de curent al microcontrolerului în timp ce execută operații aritmetice intensive. Algoritmul *SORT_ARRAY_8bit* este important pentru extragerea informațiilor despre consumul de curent al unui microcontroler ce execută bucle intensive și transferuri de date folosind memoria de date, atât pentru scriere cât și pentru citire. Algoritmul *FIND_MIN_ARRAY_FLASH_8bit* este selectat, deoarece, prin execuția lui, se pot extrage indicații despre consumul de curent al unui microcontroler ce execută bucle intensive și transferuri de date utilizând memoria de program, pentru citirea unor date constante din memoria de program.

Scenariul *CORE_LOAD* poate fi folosit atât pentru analiza curentului instantaneu, cât și pentru analiza curentului mediu al microcontrolerului testat. Având în vedere că implementarea algoritmilor de test nu prezintă modificări față de variantele prezentate în capitolul 4.1, în continuare sunt prezentate două secvențe de test generice, valabile pentru toți cei patru algoritmi. Prin măsurarea consumului de curent în acest scenariu, se pot obține informații despre impactul execuției algoritmilor asupra curentului absorbit de microcontrolerul testat.

Mai jos sunt prezentate două implementări software ale algoritmului în limbaj C. Prima variantă este utilizată pentru măsurarea curentului instantaneu, iar a doua variantă este utilizată pentru măsurarea curentului mediu.

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

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

Se observă că, în cazul implementării secvenței de test pentru măsurarea consumului mediu de curent, execuția algoritmului este introdusă într-o buclă infinită. Acest lucru este necesar pentru a ține unitatea centrală din microcontroler ocupată cu execuția algoritmului de test, fără a mai fi nevoie să execute și porțiuni din programul de test, cum ar fi reinițializări sau selectări pentru algoritmii de test. Astfel, se poate măsura curentul mediu absorbit de microcontrolerul testat atunci când acesta este ocupat doar cu execuția algoritmului dorit în scenariul de test.

În continuare sunt explorate metodele folosite pentru măsurarea curentului consumat de microcontrolerul testat. Conform celor descrise în capitolul 3, evaluarea consumului de curent pentru un microcontroler necesită determinarea curentului consumat pentru a executa o anumită porțiune de program, așa cum este ilustrat în figura 3.1. Acest studiu poate fi extins la determinarea energiei consumate, dacă se ia în calcul atât timpul de execuție, cât și puterea consumată pe parcursul execuției.

Din punct de vedere al curentului măsurat se pot distinge două valori măsurate: curentul instantaneu și curentul mediu.

Pe parcursul analizei celor trei scenarii au fost utilizate două metode de măsurare a consumului instantaneu de curent al microcontrolerelor testate. Cele două metode sunt:

- metoda de măsurare ce utilizează un multimetru configurat ca ampermetru
- metoda de măsurare ce utilizează un osciloscop și o sondă de curent.

Prin ambele metode au fost obținute rezultate similare pentru consumul de curent instantaneu măsurat. Pe parcursul studiilor s-a constatat că valorile curentului instantaneu al microcontrolerelor studiate sunt aproximativ constante în timp, în funcție de scenariul utilizat; prin urmare, avantajul principal al metodei de măsurare bazată pe un osciloscop digital, adică frecvența de eșantionare mare, nu aduce un avantaj major în extragerea rezultatelor experimentale. Pe de altă parte, multimetrul digital utilizat în experimente, Aim TTi 1908, prezintă o acuratețe bună pentru măsurarea curenților în domeniul DC de $0.05\% \pm 5$ digiți. De aceea, pentru a determina consumul de curent se va utiliza metoda bazată pe un multimetru digital, configurat ca ampermetru.

Capitolul 8

Rezultate experimentale obținute în urma măsurării consumului de curent

În acest capitol sunt prezentate rezultatele obținute în urma experimentelor în care a fost măsurat consumul de curent pentru microcontrolerele studiate, care execută cele trei scenarii de test prezentate în capitolul 7. Sunt furnizate detalii despre metodologia de testare, despre condițiile în care sunt realizate experimentele și despre erorile ce pot interveni în procesul de măsurare a curentului consumat de microcontrolerele testate.

Pentru fiecare secvență de test descrisă în capitolul precedent sunt realizate două tipuri de comparații între microcontrolere: la aceeași frecvență de funcționare a oscilatorului microcontrolerului și la aceeași viteză a unității centrale de prelucrare din microcontroler. Rezultatele sunt grupate după algoritmul de test pentru a permite o comparație mai ușoară între microcontrolerele testate.

Experimentele pentru analiza consumului de curent sunt realizate folosind o serie de microcontrolere cu arhitectură pe 8 biți. Ele sunt testate în izolare, pentru a se minimiza erorile ce ar putea interveni. Pentru a extrage rezultatele experimentale, se măsoară curentul instantaneu al microcontrolerelor studiate pe parcursul execuției celor trei secvențe de test. Se utilizează metoda de măsurare bazată pe un multimetru configurat ca ampermetru. Pe parcursul experimentelor se măsoară curentul instantaneu în mai multe momente de timp consecutive. Se calculează, apoi, media lor aritmetică.

Analiza de performanță se efectuează pe o serie de microcontrolere produse de compania Microchip Technology. Aceste microcontrolere fac parte din trei categorii de arhitecturi diferite: arhitectura PIC, arhitectura AVR și arhitectura 8051. Caracteristicile acestor arhitecturi au fost prezentate în capitolul 5.1. Condițiile de realizare a experimentelor pentru măsurarea curentului sunt similare cu cele descrise în capitolul 5.

În continuare se analizează impactul frecvenței semnalului de ceas asupra performanței microcontrolerelor testate în domeniul consumului de curent. Experimentele sunt realizate la cinci frecvențe distincte ale semnalului de ceas: 2 MHz, 4 MHz, 8 MHz, 16 MHz și 20 MHz. Excepție fac microcontrolerele cu arhitectură MegaAVR pentru care frecvența maximă de funcționare nu poate depăși 16MHz.

Se analizează și consumul de curent al microcontrolerelor testate în timpul execuției scenariilor de test prezentate în detaliu și validate experimental în capitolul 7: IDLE, INTERNAL_MODULE, CORE_LOAD (algoritmii: MUL_INT, DIV,

Rezultate experimentale obținute în urma măsurării consumului de curent

SORT_ARRAY_8bit, *FIND_MIN_ARRAY_FLASH_8bit*). Rezultatele experimentale pentru fiecare microcontroler testat sunt prezentate în tabele și figuri.

Tabelul 8.1 Scenariul CORE_LOAD – MUL_INT – consum mediu de curent funcție de frecvența semnalului de ceas pentru diferite microcontrolere

CORE_LOAD - MUL_INT – I _{mediu_total} [mA]					
Microcontroler \ Fosc	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

Tabelul 8.2 Scenariul CORE_LOAD – impactul algoritmului MUL_INT asupra consumului mediu de curent al microcontrolerelor funcție de frecvența semnalului de ceas

CORE_LOAD - MUL_INT – I _{mediu_impact} [mA]					
Microcontroler \ Fosc	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

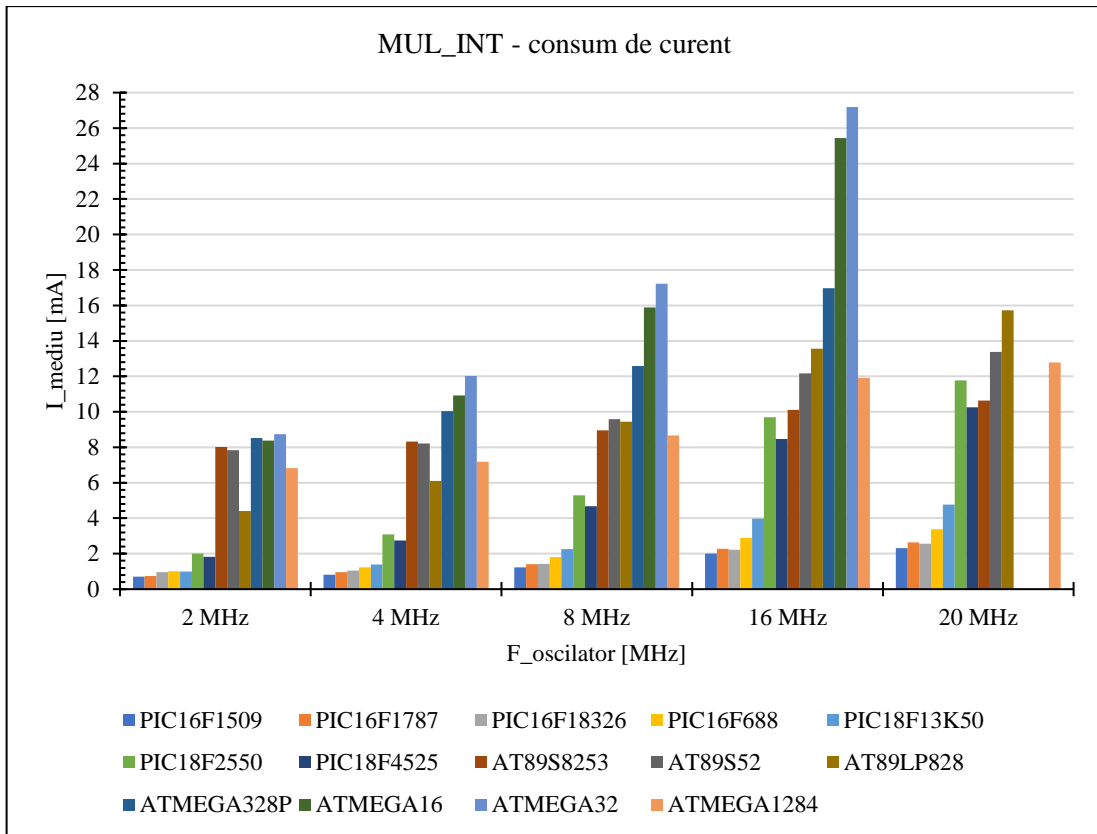


Figura 8.1 Scenariul CORE_LOAD – MUL_INT – consum mediu de curent prin microcontroler

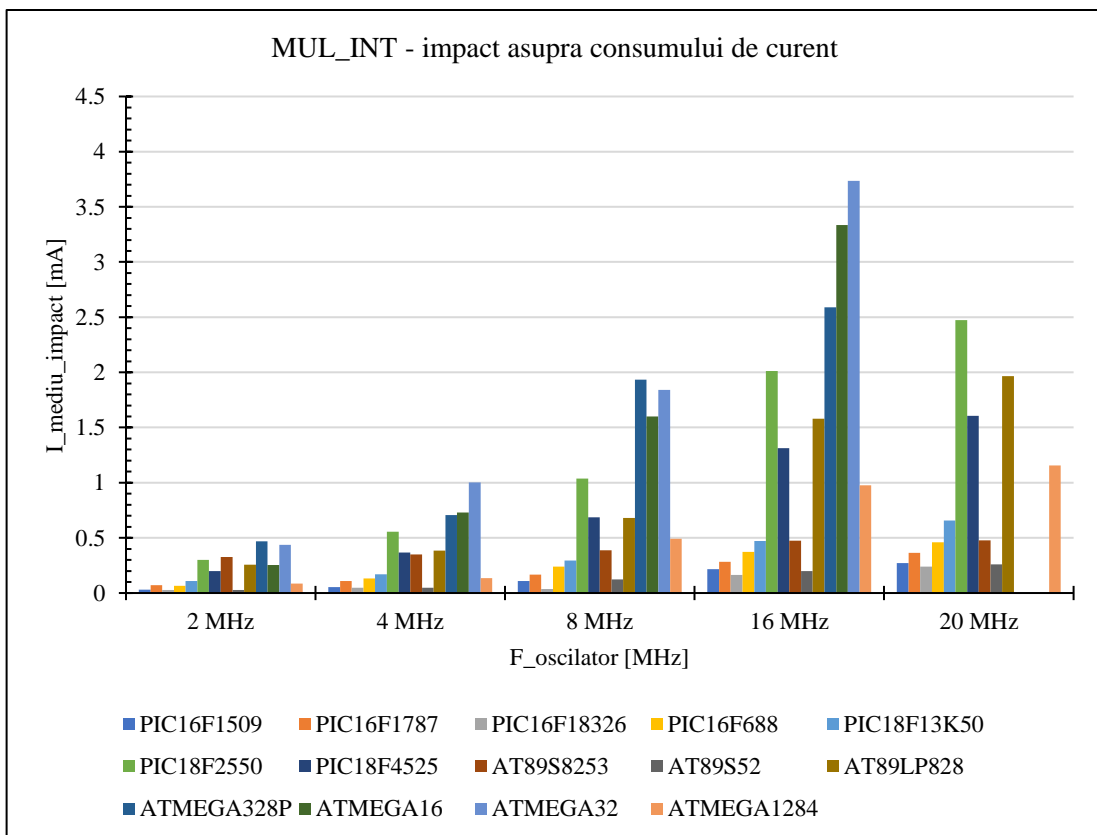


Figura 8.2 Scenariul CORE_LOAD – impactul algoritmului MUL_INT asupra consumului mediu de curent

Capitolul 9

Consumul de energie al microcontrolerelor

În acest capitol este examinată dependența de scenariile de test a energiei consumate de mai multe tipuri de microcontrolere. Pentru scenariile de test descrise în capitolul 7 este realizată comparația între mai multe microcontrolere, care funcționează la aceeași frecvență a semnalului de ceas ($F_{oscilator}$).

Rezultatele sunt grupate după scenariul de test și algoritmul de test pentru a permite o comparație mai ușoară între microcontrolerele testate. Rezultatele din acest capitol sunt obținute preluând rezultatele experimentale obținute pentru timpul de execuție al algoritmilor de test și pentru consumul de curent al microcontrolerelor în timp ce execută algoritmi de test.

În continuare se analizează impactul frecvenței de ceas asupra performanței microcontrolerelor testate în domeniul consumului de energie. Consumul de energie este calculat la patru frecvențe distincte ale semnalului de ceas: 2MHz, 4MHz, 8MHz și 16MHz. De asemenea, este analizat și consumul de energie al microcontrolerelor testate în timpul execuției secvențelor de test prezentate în detaliu și validate în capitolul 7: IDLE, CORE_LOAD (algoritmii: MUL_INT, DIV, SORT_ARRAY_8bit, FIND_MIN_ARRAY_FLASH_8bit).

Valorile energiei pentru fiecare frecvență de funcționare a microcontrolerelor testate obținute pentru scenariul CORE_LOAD și algoritmul MUL_INT sunt prezentate în tabelul 9.1 și figura 9.1.

Tabelul 9.1 Consumul de energie al microcontrolerelor examinate pentru scenariul

CORE_LOAD – MUL_INT

CORE_LOAD - MUL_INT - E [mJ]				
Microcontroler \ Fosc	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

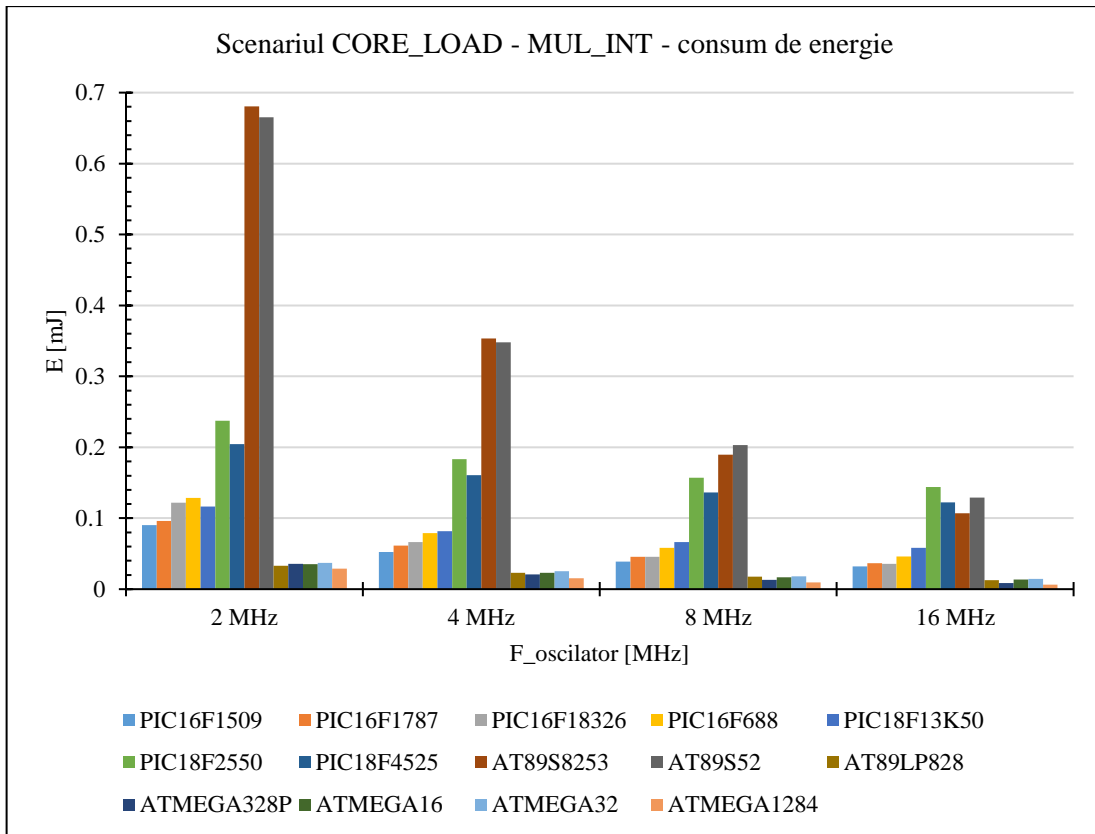


Figura 9.1 Consumul de energie al microcontrolerelor examinate pentru scenariul CORE_LOAD – MUL_INT

Capitolul 10

Concluzii

În capitolele 1 și 2 am prezentat domeniul tezei și noțiunile fundamentale necesare pentru studiul efectuat și am justificat decizia de a analiza performanțele microcontrolerelor cu arhitectură pe 8 biți. În capitolul 3 au fost trecute în revistă o serie de articole de specialitate și a fost prezentat nivelul actual al cercetării domeniul tezei din perspectiva evaluării performanței microcontrolerelor cu arhitectură pe 8 biți. În urma căutărilor mele, nu am identificat nicio lucrare în care să se evalueze explicit performanța din punct de vedere al timpului de execuție și al consumului de curent/energie pentru microcontrolerelor cu arhitecturi pe 8 biți.

Lucrarea a fost continuată în capitolele 4 - 9 cu descrierea experimentelor pentru determinarea performanței din punct de vedere al timpului de execuție și din punct de vedere al consumului de curent. La finalul fiecărui capitol au fost prezentate concluzii pe baza rezultatelor experimentale obținute.

Concluzii

În capitolul 4 am proiectat secvențele de test utilizate pentru evaluarea timpului de execuție. De asemenea, am analizat două metode de măsurare a timpului de execuție și le-am aplicat pentru a extrage primele rezultate experimentale parțiale. La finalul capitolului am ajuns la concluzia că toți cei treisprezece algoritmi de test pot fi folosiți pentru a evalua timpul de execuție și că este indicat să folosesc metoda externă de măsurare a timpului, care necesită o complexitate mai mică a programului de test, iar instrumentele externe au o acuratețe foarte bună, funcționează independent față de microcontroler și pot fi controlate direct de calculator. În capitolul 5, am elaborat metodologia de testare a performanței microcontrolerelor din punct de vedere al timpului de execuție.

Tot în capitolul 5, am selectat microcontrolerele pe baza cărora s-a realizat studiul și am descris caracteristicile arhitecturilor acestor microcontrolere. Analiza de performanță a fost efectuată pe o serie 22 de microcontrolere produse de compania Microchip Technology (inclusiv Atmel), ce fac parte din familii arhitecturale diferite. La finalul acestui capitol am descris erorile ce pot să intervină în procesul de măsurare a performanței microcontrolerelor testate.

În capitolul 6 au fost prezentate rezultatele experimentale obținute pentru timpii de execuție ai tuturor algoritmilor prezentați. În urma comparației timpilor de execuție ai algoritmilor folosiți, măsurați experimental la aceeași frecvență de ceas a oscilatorului microcontrolerului, am tras următoarele concluzii:

- Dacă frecvența de funcționare a microcontrolerelor comparate este constantă se constată că timpii de execuție variază în funcție de microcontroler și de arhitectura lui.
- Timpul necesar execuției algoritmilor de test scade odată cu creșterea frecvenței de ceas a microcontrolerului testat.
- S-a remarcat existența unor grupuri de microcontrolere pentru care fiecare algoritm a fost executat într-un timp similar. Aceste grupuri de microcontrolere depind de arhitecturile folosite: PIC, 8051 și AVR.
- Optimizările compilatorului pot influența rezultatele experimentale.
- Dimensiunea memoriei de date și de program impactează direct dacă un program de test poate sau nu poate rula pe un anumit microcontroler.

În capitolul 7 am elaborat și proiectat scenariile de test ce au ca scop măsurarea consumului de curent al microcontrolerelor. Au fost identificate 3 scenarii de test. Fiecare scenariu de test a fost evaluat experimental pe cazuri particulare, obținându-se rezultate experimentale intermediare. Apoi a fost estimat impactul acestor scenarii asupra performanței microcontrolerelor studiate. De asemenea, au fost cercetate și validate experimental metodele de evaluare a performanței microcontrolerelor din punct de vedere al consumului de curent. Aceste metode de măsurare ale consumului de curent sunt validate experimental pe o serie de cazuri particulare. În final, am ajuns la concluzia că este indicat să folosesc un multimetru digital configurat ca ampermetru.

În capitolul 8 am prezentat rezultatele experimentale obținute în urma măsurării consumului de curent al microcontrolerelor studiate, care execută cele trei scenarii de test prezentate în capitolul 7. Pentru comparația la aceeași frecvență de ceas a oscilatorului

microcontrolerului, pe baza rezultatelor experimentale pentru consumul de curent, am tras următoarele concluzii:

- Pentru scenariul de test CORE_LOAD s-a observat din nou o creștere a consumului de curent față de scenariul IDLE pentru toți algoritmi de test și pentru toate microcontrolerele testate. În acest scenariu, cel mai mic consum de curent a fost obținut de microcontrolerele cu arhitectură PIC (familia PIC16), iar cel mai mare consum de curent a fost obținut, de asemenea, de microcontrolerele cu arhitectură megaAVR.
- Analizând consumul de curent al fiecărei familii de microcontrolere studiate s-a observat că dintre microcontrolerele cu arhitectură PIC, cel mai mic consum de curent a fost obținut de microcontrolerul PIC16F1509, iar cel mai mare consum de curent a fost obținut de microcontrolerul PIC18F2550. Analizând microcontrolerele cu arhitectură megaAVR, cel mai mic consum de curent a fost obținut de microcontrolerul ATMEGA1284, iar cel mai mare consum de curent a fost obținut de microcontrolerul ATMEGA32. În final, dintre microcontrolerele cu arhitectură 8051, cel mai mic consum de curent a fost obținut de microcontrolerul AT89S8253, iar cel mai mare consum de curent a fost obținut de microcontrolerul AT89S52.
- Consumul de curent al microcontrolerelor studiate variază odată cu frecvența de ceas, cu încărcarea unității centrale de prelucrare și cu intensitatea utilizării transferurilor de date folosind memoria de date sau folosind memoria de program.

În capitolul 9 am examinat dependența de scenariile de test a energiei consumate de mai multe tipuri de microcontrolere. Pentru fiecare scenariu și algoritm de test descrise în capitolul 7 am realizat comparația între microcontrolerele ce funcționează la aceeași frecvență de ceas. Am determinat astfel câtă energie a fost necesară pentru ca un microcontroler să execute un anumit algoritm de test.

Analizând în detaliu rezultatele pentru secvența CORE_LOAD am constatat că energia necesară execuției algoritmilor de test variază de la microcontroler la microcontroler. De asemenea, frecvența de ceas joacă un rol foarte important în consumul de energie deoarece influențează în mod direct atât timpul de execuție al algoritmilor de test cât și consumul de curent al microcontrolerelor. Pentru fiecare algoritm de test am observat următoarele:

- Pentru toți algoritmi de test consumul de energie scade odată cu creșterea frecvenței semnalului de ceas.
- În cazul algoritmului MUL_INT cel mai mic consum de energie a fost atins de microcontrolerele megaAVR, iar dintre acestea, cel mai mic consum de energie a fost obținut de ATMEGA1284 pentru toate frecvențele de ceas. Cel mai mare consum de energie a fost obținut de microcontrolerele cu arhitectura 8051 (familia AT89S) pentru frecvențele de ceas de 2 MHz, 4 MHz și 8 MHz. La frecvența de ceas de 16 MHz, PIC18F2550 a avut cel mai mare consum de energie.

Concluzii

- În cazul algoritmului FIND_MIN_ARRAY_FLASH_8bit cel mai mic consum de energie a fost obținut de microcontrolerele PIC16, dintre acestea consumul de energie minim fiind obținut de PIC16F1509 pentru toate frecvențele de ceas. Cel mai mare consum de energie a fost obținut de microcontrolerele cu arhitectura 8051 (familia AT89S) pentru toate frecvențele.
- De asemenea, microcontrolerele cu arhitectură PIC (familia PIC16) au fost cele mai eficiente din punct de vedere al consumului de energie. Microcontrolerele cu arhitectură megaAVR au fost aproximativ la fel de eficiente, obținând rezultate puțin mai slabe. Cele mai puțin eficiente dintre microcontrolerele testate au fost cele cu arhitectură 8051 (familia AT89S).
- Din punct de vedere al frecvenței de ceas a microcontrolerelor studiate s-a observat cum consumul de energie electrică al microcontrolerelor testate a variat cu frecvența oscilatorului; consumul de curent crește odată cu creșterea frecvenței de ceas dar timpul de execuție al secvențelor de test scade foarte mult cu creșterea frecvenței de ceas, fapt ce duce la o scădere a consumului de energie electrică. În caz contrar, chiar dacă un microcontroler execută mai lent o secvență de test, acesta poate consuma mai puțin curent, ceea ce poate însemna o cantitate mai mică de energie consumată.
- Din punct de vedere al încărcării unității centrale de prelucrare, consumul de energie al microcontrolerelor studiate crește cu cât încărcarea este mai mare.

Având în vedere rezultatele obținute în lucrare se poate afirma că scopul inițial al lucrării a fost atins, adică au fost identificate metode pentru evaluarea performanței unor familii de microcontrolere cu arhitectură pe 8 biți, din punct de vedere al timpului de execuție, al consumului de curent și al consumului de energie. Aceste metode au fost aplicate practic pe un set de microcontrolere cu arhitectură pe 8 biți din arhitecturile cele mai reprezentative, și anume arhitectura PIC (din categoriile de mijloc, de mijloc-îmbunătățit și de vârf, reprezentate de familiile PIC12, PIC16 și PIC18), arhitectura AVR (categoria megaAVR) și arhitectura 8051 (familiile AT89S și AT89LP). Rezultatele experimentale obținute au fost folosite pentru a analiza performanța microcontrolerelor testate, iar rezultatele obținute sunt reproductibile. Pe baza rezultatelor experimentale s-au putut determina caracteristicile de performanță pentru toate microcontrolerele incluse în studiu. În linii mari, dintre microcontrolerele studiate în această lucrare, microcontrolerele cu arhitectură AVR au dovedit cea mai bună performanță din punctul de vedere al timpului de execuție, iar microcontrolerele cu arhitectură PIC au dovedit cea mai bună performanță din punctul de vedere al consumului mediu de curent și din punctul de vedere al consumului de energie electrică.

În concluzie, teza a îndeplinit scopul inițial și conține o multitudine de contribuții originale ce sunt enumerate în continuare.

Contribuții originale

Contribuțiile originale aduse în această lucrare sunt enumerate mai jos:

- Identificarea parametrilor de performanță ce trebuie analizați astfel încât să reflecte situații și scenarii reale. Capitolele 3, 4 și 7, lucrările [3] și [5].

- Selectarea, proiectarea și implementarea algoritmilor de test pentru măsurarea parametrilor de performanță. Capitolele 4 și 7, lucrările [3] și [4].
- Stabilirea metodologiei de testare astfel încât parametrii mășurați să nu fie afectați de erori de măsură interne sau externe. Capitolele 5 și 7.
- Proiectarea sistemelor automate pentru măsurarea și testarea performanțelor microcontrolerelor. Capitolele 5 și 7, lucrările [1], [6] și [7].
- Implementarea fizică a montajelor și programelor de test ce alcătuiesc sistemele automate pentru testarea performanțelor microcontrolerelor. Anexele 3 și 4.
- Extragerea caracteristicilor de performanță din punct de vedere al timpului de execuție pentru o serie de microcontrolere cu arhitectură pe 8 biți, în urma rulării experimentelor. Capitolul 6, lucrarea [4].
- Interpretarea rezultatelor experimentale obținute pentru timpul de execuție și caracterizarea microcontrolerelor din punct de vedere al timpului de execuție. Capitolele 6 și 10, lucrările [1], [4] și [5].
- Extragerea caracteristicilor de performanță din punct de vedere al consumului de curent pentru o serie de microcontrolere cu arhitectură pe 8 biți, în urma derulării experimentelor. Capitolul 8, lucrarea [2].
- Interpretarea rezultatelor experimentale obținute pentru consumul de curent și caracterizarea microcontrolerelor din punct de vedere al consumului de curent. Capitolele 8 și 10, lucrările [1], [2] și [3].
- Extragerea caracteristicilor de performanță referitoare la consumul de energie pentru o serie de microcontrolere cu arhitectură pe 8 biți, în urma derulării experimentelor. Capitolul 9.
- Interpretarea rezultatelor experimentale obținute pentru consumul de energie și caracterizarea microcontrolerelor din punct de vedere al consumului de energie. Capitolul 9.
- Realizarea unui studiu comparativ între microcontrolerele cu arhitectură pe 8 biți folosite în experimente. Capitolele 6, 8, 9 și 10, anexele 1 și 2, lucrarea [1].

Lista lucrărilor originale

Mai multe contribuții originale din teză au făcut subiectul articolelor publicate enumerate mai jos:

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.

Perspective de dezvoltare ulterioară

Având în vedere rezultatele obținute până acum și aprecierea primită la conferințele de specialitate doresc să continui această activitate de cercetare pe trei mari direcții:

- includerea mai multor familii de microcontrolere cu arhitectură pe 8 biți,
- extinderea studiului pentru microcontrolerele cu arhitectură pe 16 biți și pentru microcontrolere cu arhitectură pe 32 biți,
- analiza mai multor parametri de performanță ai microcontrolerelor.

Prin includerea în studiu a mai multor familii de microcontrolere cu arhitectură pe 8 biți înțeleg compararea rezultatelor experimentale obținute pe un set mai mare și mai divers de microcontrolere. Spre exemplu, în studiul extins ar putea fi incluse

microcontrolere cu arhitectură pe 8 biți produse de companiile Infineon Technologies, NXP Semiconductors sau Texas Instruments. Extinderea studiului poate duce la obținerea unei imagini mai clare asupra performanțelor microcontrolerelor cu arhitectură pe 8 biți.

Prin extinderea studiului pentru microcontrolerele cu arhitectură pe 16 biți și pentru microcontrolere cu arhitectură pe 32 biți se poate obține o imagine completă a performanțelor microcontrolerelor în general. Această extindere este motivată de concurență aprigă pentru microcontrolerele cu arhitectură pe 8 biți. Microcontrolerele cu arhitectură pe 16 biți și 32 biți devin din ce în ce mai variate, mai performante și, foarte important, mai ieftine.

O altă posibilă cale de dezvoltare constă în evaluarea mai multor parametri de performanță ai microcontrolerelor. Spre exemplu, în studiul extins ar putea fi studiate mai multe secvențe de test din care să reiasă parametri de performanță adiționali, creând astfel o imagine și mai clară asupra performanței microcontrolerelor.

Bibliografie

- [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 – ediția a III-a”, Editura Politehnica press, București, 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 Benchmarking”, 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 (IIC), 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] Mikroelektronika, „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 Khoo 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 introduction 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.