An Empirical Methodology for Power Analysis of CMOS Integrated Circuits

Momcilo V. Krunic[1], Ivan Povazan[1], Jelena V. Kovacevic[2], Vlado M. Krunic[3]
*[1]RT-RK, Institute for Computer Based Systems LLC,
Narodnog Fronta 23a, 21000 Novi Sad, Serbia
[2]Department of Computer Engineering and Communications, Faculty of Technical Sciences,
University of Novi Sad,
Trg Dositeja Obradovica 6, 21000 Novi Sad, Serbia
[3]Department of Mathematics and Informatics, Faculty of Natural Sciences and Mathematics,
University of Banja Luka
momcilo.krunic@rt-rk.com*

***Abstract*—Energy consumption is becoming one of the most significant aspects of CMOS Integrated Circuits (IC), especially for those applied in embedded devices whose autonomy depends upon battery lifespan. Therefore, an empirical methodology for determination of power and energy dissipation may provide valuable information to IC designers, as well as software developers, which could impact design process and lead to more energy-efficient solutions. This paper presents a novel methodology for determination of static and dynamic components of energy dissipation for those CMOS ICs that do not support turning off clock distribution entirely, but provide ability to divide a clock frequency. For that purpose, we used an Eclipse based IDE that provides a user friendly interface for dividing a clock frequency on ultra-low power embedded DSP platform, which was used as a target device. Measurements were performed using a true RMS multimeter. Various experiments were conducted using different scenarios, on single and multi cores, in order to validate the described empirical methodology, and the outcome confirmed what was expected, that the obtained results are stable and accurate.**

***Index Terms*—Power measurement; energy dissipation; CMOS integrated circuits; embedded software; performance evaluation.**

I. Introduction

The research described in this paper represents continuous efforts to develop an accurate energy consumption estimation model for embedded applications. Initial research, presented in [1] and [2], focused on modeling energy dissipation on ultra-low power heterogeneous multicore DSP platform, whilst executing an embedded application. The same target platform was used in this research, but the spotlight was on decomposition of energy consumption from the highest level of abstraction, such as an instruction, to the lowest level, a CMOS transistor in this case. Also, a significant part of this research are novel empirical methodologies introduced to evaluate each component.

A CMOS (Complementary Metal Oxide Semiconductor) [3] is considered a dominant technology that has been used for decades in the production of IC, primarily due to: reliability, low consumption, low costs, and scalability [4]. Although this research uses just a small subset of CMOS IC, an embedded DSP target platform, for validation of derived conclusions through various experiments, one should be aware that the presented results and methodologies are applicable to any other CMOS IC, which implicates universality of the solution.

The driving force that inspired this research can be found in exigency to reduce the amount of energy dissipation on a mobile device, in this case a hearing aid, in order to achieve greater autonomy of the device. The initial idea was to develop a profiling tool [5] and its extension [1], which should provide a direct and vivid link between firmware solution and energy consumption. One of the most challenging obstacles on that road was certainly to identify and measure all the components that contribute to energy dissipation, at the instruction level.

It has been established that instruction level energy consumption estimation can be achieved using two different approaches: physical measurements on real hardware, and simulation based modelling. Research presented in [6] and [7] has shown that the first approach provides more accurate information. That conclusion, along with the lack of hardware simulation model for the target device, leads us to choose the first approach, and to perform measurements on the target device. For that purpose, a novel measurement methodology that provides accurate information about static and dynamic energy components was established. The methodology is currently in the patent process [8].

The rest of the paper is organized in the following order. Section II looks at related papers. Section III describes components and causes of energy dissipation within CMOS IC. In section IV we described the proposed measurement methodology. Section V provides an insight into the experimental results and validation methodology. Section VI describes future work and concludes the paper.

Manuscript received 8 December, 2016; accepted 25 May, 2017.

This research was partially funded by the Ministry of Education and Sciences of the Republic of Serbia under project No. TR-32031. This research was performed in cooperation with RT-RK Institute for Computer Based Systems.

II. RELATED PAPERS

Unsustainable hunger for energy is becoming reality since the number of on-chip devices doubles every two years. As we are going forward, by shrinking the size of technology process nodes, the problem with energy becomes even worse (Fig. 1).

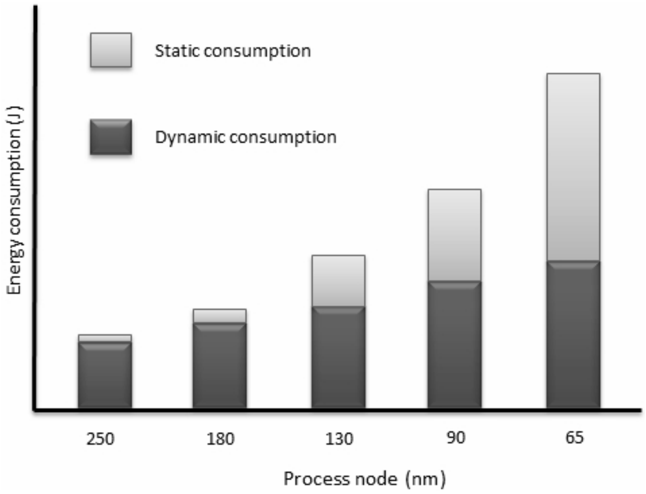


Fig. 1. Static and dynamic energy consumption depending on process node.

Table I provides accurate information about the amount of energy consumed per cm^2 depending on different sizes of technology process nodes. The source of data represented in Fig. 1 and Table I can be found at [9]. It is obvious that dynamic power per cm^2 is practically inversely proportional to the size of process node, unlike the static power per cm^2, whose increase highly depends on node size shrinking. Therefore, one could make a conclusion that it is important to distinguish between static and dynamic components in order to achieve a more accurate energy consumption estimation model.

TABLE I. POWER INCREASE DEPENDING ON TECHNOLOGY PROCESS NODE.

Process node	90nm	65nm	45nm
Dynamic power/cm^2	1X	1.4X	2X
Static power/cm^2	1X	2.5X	6.5X
Total power/cm^2	1X	2X	4X

An accurate methodology for power analysis of CMOS IC may provide an insight into the energy dissipation that could help developers and IC designers create more energy-efficient solutions. There are many solutions that deal with power of CMOS IC, but few that provide accurate distinction between static and dynamic components of energy dissipation. The solution presented in this paper provides a methodology for that kind of power analyses.

Empirical measurements proposed by [6], [10]–[12] do not take into consideration the importance of distinction between static and dynamic components. The current drawn by the process whilst executing an instruction was measured in order to obtain instruction's base cost, but discussion about decomposition of the measured value into static and dynamic components was omitted. Also, in [6] and [10] it has been noted that inter-instruction effect could be excluded, because of small influence on the overall consumption, which is not the case with the target platform used in this research, where consumption influenced by that effect in some cases exceeds base cost several times.

Power consumption model presented in [13], similarly to our model presented in this paper, and in [1], deals with multicores, but with several significant differences. First of all, our model is applicable to heterogeneous multicore processors, and evaluates power at instruction level, rather than at core utilization. Also, the crucial difference between models presented in here and the one presented in [13] could be found in the fact that evaluation model [13] considers just dynamic power consumption of multi-core processors, opposite to the model presented in this paper that takes into account both aspects of power consumption: static and dynamic.

Research presented in [14]–[17] introduce advanced measurement systems that are capable to measure energy consumed between two cycles, but still, an appropriate discussion about static and dynamic power consumption measurement was missing.

Energy consumption estimation model presented in [18] introduces static energy as a distinct parameter within the model. The described model [18] has been tested using ARM7TDMI processor core. However, it did not clearly state the methodology for determining static power, nor the empirical results for that component measured on the target core. Also, validation of the proposed model in [18] was performed only for fixed clock cycle values, which is not the case with the research presented in this paper.

Methodologies for energy consumption estimation presented in [19] and [20] depend on hardware simulation model and because of that cannot be applied to the target platform used in this research.

In the initial research [1], static energy was promoted as one of the parameters within the energy consumption estimation model. The problem occurred after launching validation tests at different clock frequencies. It was concluded that validation tests failed due to incorrect value assigned to static energy, which becomes the main motivation for this research.

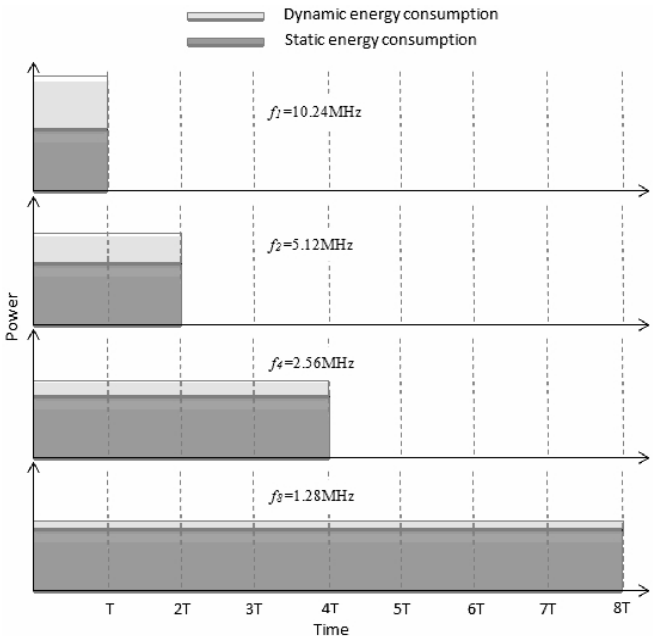


Fig. 2. Static and dynamic energy consumption.

It is worth mentioning that distinct power analysis of dynamic and static components becomes especially important for estimation of energy consumption on low-power CMOS IC with variable clock period, because of two facts: dynamic energy and static power do not depend on time (Fig. 2). This ascertainment will be explained further in the following section.

III. POWER AND ENERGY CONSUMPTION OF CMOS INTEGRATED CIRCUITS

Average power of software application – P, could be defined as arithmetic mean of powers of each individual cycle [1]

$$P = \frac{1}{N}\sum_{k=0}^{N\text{-}1} P_{c(k)}, \tag{1}$$

where N represents number of executed cycles, and $P_{c(k)}$ denotes the average power of the *k-th* cycle. As stated earlier, energy dissipation of CMOS IC consists of two basic components [4], [21]: static and dynamic. Bearing that in mind, power of executed cycle – P_c, could be defined as

$$P_c = P_{stat} + P_{dyn}, \tag{2}$$

where P_{stat} denotes static power dissipation, and P_{dyn} stands for dynamic power dissipation. Energy dissipation within a cycle – E_c, could be defined then as

$$E_c = P_c \times t = P_c \times \frac{1}{f} = \frac{P_{stat} + P_{dyn}}{f} = \frac{P_{stat}}{f} + \frac{P_{dyn}}{f}, \tag{3}$$

where f represents clock cycles frequency, and t denotes clock period. Figure 2 represents power and energy consumption within one cycle, at four different frequencies, each one half less than the previous. It can be noticed that static power is constant, as well as dynamic energy, therefore reducing the frequency, and static energy consumption becomes dominant. This statement will be theoretically explained and validated through the experiment on the target platform.

Overall energy dissipation, during software runtime, could be defined as multiplication of average power – P_c, and runtime value t_T

$$E = P \times t_T. \tag{4}$$

In order to evaluate runtime value, the Performance equation [22]–[24] could be used for that purpose

$$t_T = CT \times CPI \times IC. \tag{5}$$

CT represents clock time, CPI denotes clocks per instruction, and IC denotes instruction count.

A. Static Component

Static energy consumption arises as a direct consequence of leakage currents, during a stable state of CMOS transistor. Leakage current occurs when there is no transition of input state. It is caused by several factors [4], [25], [26], depicted in Fig. 3. Leakage current – I_{leak} of CMOS transistor, could be defined as a sum of all those components

$$I_{leak} = I_{REV} + I_{SUB} + I_G + I_{GIDL}, \tag{6}$$

where I_{REV} represents reverse-bias p-n junction diode leakage, I_{SUB} denotes subthreshold leakage, I_G represents gate leakage through the oxide, and I_{GIDL} denotes gate induced drain leakage.

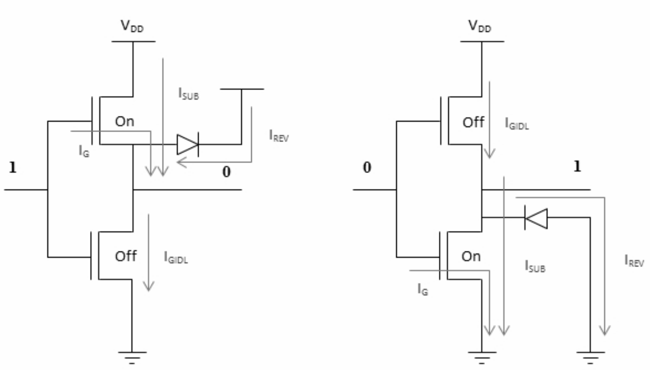


Fig. 3. Static leakage currents of CMOS inverter in two different states.

Leakage current of CMOS IC – I_{stat} could be defined as a sum of leakage currents of each individual CMOS transistor – I_{leak} that is under the voltage supply

$$I_{stat} = \sum_{m=0}^{M-1} I_{\text{leak(m)}}, \tag{7}$$

where M denotes the number of CMOS transistors under the voltage supply. If there is a consistent voltage supply, then we could claim that static power is constant, and it can be defined as multiplication of overall leakage current – I_{stat} and voltage supply – V_{DD}

$$P_{stat} = I_{stat} \times V_{DD}. \tag{8}$$

This means that nothing but voltage supply can influence a static power over time. It is also worth noticing that changing of clock frequency should not affect leakage current. That assumption will be used as a crucial characteristic for static power evaluation. Static energy consumed within one clock cycle could be expressed as

$$E_{stat} = P_{stat} \times T = I_{stat} \times V_{DD} \times T. \tag{9}$$

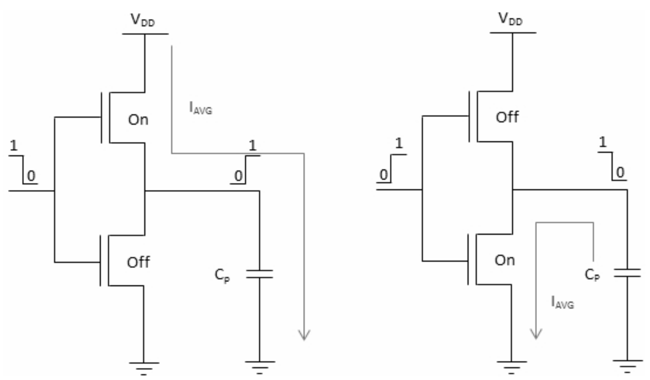


Fig. 4. Dynamic currents of CMOS inverter in two different states.

B. Dynamic Component

Dynamic energy dissipation of CMOS IC occurs during a switching activity of logical states. There are three different causes that contribute to dynamic energy consumption [27]: capacitive load – C_P, internal capacitance – C_I, and transients. In case CMOS IC is triggered by clock cycles with the same interval of impulse and pause, dynamic power dissipation caused by capacitive load could be expressed with (10)

$$P_P = V_{DD} \times I_P = \frac{C_P \times V_{DD}^2}{T} = C_P \times V_{DD}^2 \times f. \tag{10}$$

Similarly, dynamic power dissipation caused by internal capacitances can be defined

$$P_I = V_{DD} \times I_I = \frac{C_I \times V_{DD}^2}{T} = C_I \times V_{DD}^2 \times f. \tag{11}$$

Dynamic power dissipation caused by transients can be defined using the expression

$$P_T = t_S \times V_{DD} \times I_{\max} \times f, \tag{12}$$

where t_S represents duration of transient current I_{max}. Finally, expression for dynamic power can be defined

$$\begin{gathered} P_{dynT} = P_P + P_I + P_T = \\ = (V_{DD} \times f)((C_P + C_I) \times V_{DD} + t_S \times I_{\max}). \end{gathered} \tag{13}$$

The member that contains transition time could be excluded from the expression (13), since t_S value is quite low, making its contribution negligible. Equivalent capacitance – C_E can be defined as a sum of capacitive load and internal capacitances

$$C_E = C_P + C_I. \tag{14}$$

Now, using (13) and (14), simplified expression for dynamic power of CMOS transistor – P_{dynT} can be defined

$$P_{dynT} = C_E \times V_{DD}^2 \times f. \tag{15}$$

From (15) it is clear enough that dynamic power depends on clock cycle frequency, but dynamic energy does not (18).

If we consider that CMOS IC contains K active transistors during some particular cycle, then dynamic power of CMOS IC, during that cycle, could be defined as:

$$C_{eff} = \sum_{n=0}^{K-1} C_{E(n)}, \tag{16}$$

$$\begin{gathered} P_{dyn} = \sum_{n=0}^{K-1} P_{dynT(n)} = V_{DD}^2 \times f \times \sum_{n=0}^{K-1} C_{E(n)} = \\ = V_{DD}^2 \times f \times C_{eff}, \end{gathered} \tag{17}$$

where C_{eff} denotes *effective capacitance*, which represents overall capacitances that should be charged and discharged during a cycle, and it could be used as quantitative measure of dynamic energy. Figure 5 illustrates this idea

$$E_{dyn} = \frac{P_{dyn}}{f} = V_{DD}^2 \times C_{eff}. \tag{18}$$

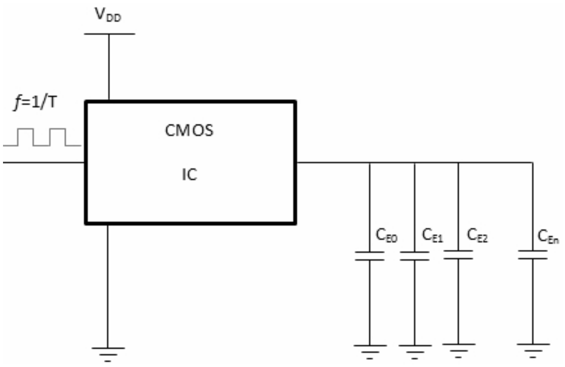


Fig. 5. Illustration of effective capacitances.

From (18) it is obvious that dynamic energy, dissipated within one cycle, does not depend on clock period, which is in accordance with the statement expressed earlier, outlined in Fig. 2. This characteristic makes effective capacitance interesting, because once it is determined, using one clock frequency, it could provide an accurate estimation of dynamic power on other frequencies.

After inclusion of (8) and (15) into (2), expression for power of one cycle derives into

$$P_c = I_{stat} \times V_{DD} + C_{eff} \times V_{DD}^2 \times f. \tag{19}$$

Afterwards, derivation of expression for average power of software application (1), whose execution last N cycles, follows

$$\begin{gathered} P = \frac{1}{N} \sum_{k=0}^{N\text{-}1} P_{c(k)} = \\ = I_{stat} \times V_{DD} + \frac{1}{N} \sum_{k=0}^{N\text{-}1} V_{DD}^2 \times f \times C_{eff(k)}. \end{gathered} \tag{20}$$

In case just one instruction has been executed over time, effective capacitance would be the same in each cycle, which implicates that the expression for average power (19) may derive into the expression for cycle power (20). This observation is emphasized, because it will be used as an important characteristic during experiments.

IV. PROPOSED MEASUREMENT METHODOLOGY

Distinct measurement of static and dynamic power dissipation demands determination of one component in order to calculate the other. Dynamic power dissipation could be suppressed in case that clock distribution has been disabled on the entire CMOS IC, but that is not the case with most target platforms. Even if it is possible to turn off the clock distribution, one should have a methodology to verify that.

Ultra-low power heterogeneous multicore DSP target platform designed for hearing aids was used for this research. It was developed using 90 nm technology process node. It contains five different DSP cores. One DSP core plays the role of microcontroller, which controls and synchronizes events on the hardware. Two DSP cores are of general purpose, used for various actions. The remaining two DSP cores are designed and optimized to perform numerical accelerations. Beside all that, there are several peripherals used for I/O communication, mutual synchronization, and scaling of voltage and frequency. It is important to emphasize that clock distribution on this target platform could be turned off almost entirely. Clock distribution cannot be disabled only for the microcontroller, since that could initiate unexpected behaviour of the target platform. In order to properly setup the target platform for measurements it is necessary to download appropriate configuration for parameters such as: clock frequency and distribution, voltage scaling, special function registers (SFR), etc. The prerequisite for that task is proper clock distribution.

In order to distinguish between static and dynamic power dissipation we propose a simple, yet effective measurement methodology. The fact that static power (8) is independent, and dynamic power (17) is directly proportional to clock frequency was used as the main characteristic to detect static power dissipation. The idea was to measure overall power dissipation, at one clock frequency – P_{m1}; afterwards, clock frequency should be divided by factor d, and measurements should be performed once again on the new configuration – P_{m2}. Based on those two measurements, constant and variable part of measured power dissipation can be identified. The following two equations describe this methodology:

$$P_{dyn} + P_{stat} = P_{m1}, \tag{21}$$

$$\frac{P_{dyn}}{d} + P_{stat} = P_{m2}. \tag{21}$$

Since measurements should be performed at the same voltage level, (21) and (21) deriving into the following:

$$I_{dyn} + I_{stat} = I_{m1}, \tag{22}$$

$$\frac{I_{dyn}}{d} + I_{stat} = I_{m2}. \tag{23}$$

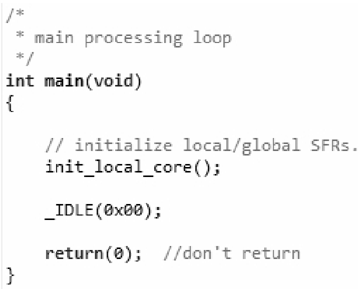
Based on those two equations, an expression for the leakage current of the CMOS IC can be derived

$$I_{stat} = \frac{d \times I_{m2} - I_{m1}}{d-1}, \tag{24}$$

I_{m1} represents the current measured during the base clock frequency, and I_{m2} denotes the current measured after the clock frequency has been divided by factor d.

V. EXPERIMENTAL RESULTS AND VALIDATION

Experiments were performed at four different frequencies, base clock frequency (10.24 MHz) and three others, divided by factors: two, four and eight, respectively. Measurements were performed using a true RMS multimeter. Target platform was configured to turn off clock distribution for all peripherals and DSP cores, except for microcontroller (uC). Test image for uC was built using the source code provided in Fig. 6. First of all, a function “init_local_core” is called, which configures SFRs and, more importantly, clock frequency. So, each time the frequency should be modified, it is done over the “init_local_core” function. Afterwards, the core is put to sleep by executing the IDLE instruction. Therefore, dynamic power dissipation, measured during experiments, could be attributed to the IDLE instruction.



```
/*
 * main processing loop
 */
int main(void)
{

    // initialize local/global SFRs.
    init_local_core();

    _IDLE(0x00);

    return(0);  //don't return
}
```

Fig. 6. Source code of the test image.

TABLE II. MEASURED VALUES AT DIFFERENT CLOCK FREQUENCIES.

Freq [MHz]	I_m [µA]	V_{DD} [V]	P_m [µW]
10.24	740	1.25	925
5.12	550	1.25	687.5
2.56	452.7	1.25	565.875
1.28	404.3	1.25	505.375

Table II provides experimental results measured at four different clock frequencies. Now that I_m values are available, they could be included into the expression (24).

TABLE III. CALCULATED VALUES OF STATIC CURRENT.

Freq [MHz]	10.24	5.12	2.56	1.28
10.24	X	360 µA	356.9 µA	356.3 µA
5.12	360 µA	X	355.4 µA	355.7 µA
2.56	356.9 µA	355.4 µA	X	355.9 µA
1.28	356.3 µA	355.7 µA	355.9 µA	X

Table III contains calculated values of static current – I_{stat}, using all combinations of measured values provided in Table II and (24). Based on the data provided in Table III, average static current and standard deviation can be calculated:

$$I_{stat} = \frac{1}{n}\sum_{i=1}^{n} I_{stat(i)} = 356{,}7\mu A, \tag{25}$$

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(I_{stat(i)} - I_{stat})^2} = 1.955\mu A. \tag{26}$$

According to the values calculated in (25) and (26) relative standard deviation (RSD) can be obtained:

$$RSD = \frac{\sigma}{I_{stat}} \times 100\% = 0.548\%. \tag{27}$$

Based on RSD, it can be concluded that calculated leakage current of the CMOS IC is stable, and the methodology valid.

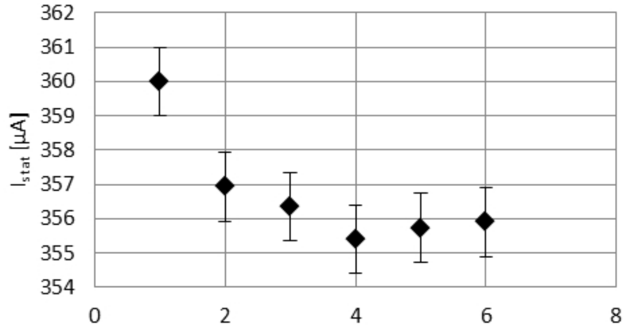


Fig. 7. Standard deviation of static current.

Using the calculated value of static current (25), the exact amount of power dissipation caused by static and dynamic components can be determined (Fig. 8).

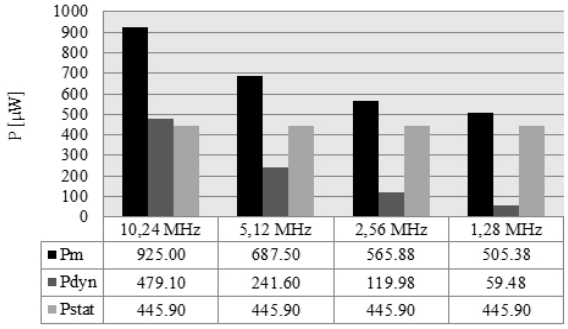


Fig. 8. Power dissipation of CMOS IC.

From Fig. 8 it can be concluded that decrease of clock frequency causes reduction of power dissipation, which is the direct consequence of dynamic power dissipation. If frequency is increased, there will be more switching activity, and therefore overall power will be increased. However, if we take a look at Fig. 9, it is obvious that frequency value decrease implicates increase of energy dissipation within one cycle. This is caused by static power dissipation, which is constant over time, meaning that if the clock period is increased, more static energy will be consumed.

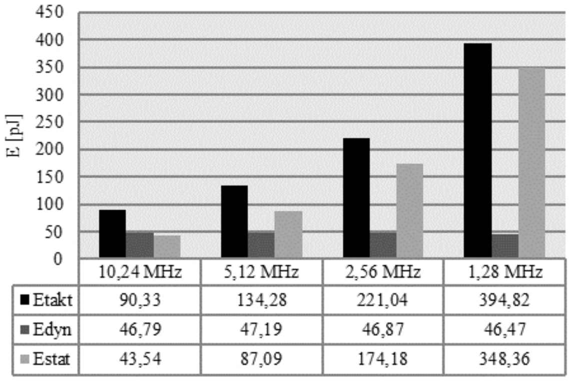


Fig. 9. Energy dissipation of CMOS IC during one clock cycle.

Based on vivid presentation of power (Fig. 8) and energy (Fig. 9) dissipation, two conclusions could be made:

1. When core is in idle state, frequency should be decreased as much as possible, in order to reduce switching activity, and thus dynamic energy dissipation.
2. When core is in execution mode, frequency should be increased as much as possible, in order to reduce static energy dissipation.

In Fig. 8 dynamic power dissipation was calculated using (28), (25) and I_m values from Table II.

TABLE IV. FIR FILTER ENERGY CONSUMPTION.

Freq [MHz]	N	I_m [µA]	V_{DD} [V]	E_m [nJ]
10.24	1485	1253	1.25	227.14
5.12	1485	809	1.25	293.30
2.56	1485	583	1.25	422.73
1.28	1485	471	1.25	683.04

$$P_{dyn} = P_m - P_{stat} = V_{DD} \times (I_m - I_{stat}). \qquad (28)$$

In order to calculate the influence of the clock frequency value on the overall energy consumption, an experiment has been conducted using FIR filter implementation, running on four different frequencies.

Table IV represents the outcome of that experiment, where N represents the number of cycles that has been consumed by one execution of the FIR filter main loop, I_m denotes average current measured during the experiment, V_{DD} represents power supply voltage, and E_m denotes energy dissipation. From Table IV it can be calculated that FIR filter implementation, on the target platform used in this research, consumes around three times less energy on the base clock frequency compared to the frequency that is eight times lower.

As stated above, dynamic power dissipation could be estimated based on effective capacitances. Using (17), and dynamic power (Fig. 8), at base frequency (479.1 µW), we can calculate effective capacitances of the core running in idle state

$$C_{eff} = \frac{P_{dyn}}{V_{DD}^2 \times f} =$$
$$= \frac{479.1}{1.25^2 \times 10.24} \times 10^{-12} = 29.94 pF. \qquad (29)$$

This actually means that 29.94 pF will be charged and discharged, during one clock cycle, on the uC whilst running in idle state.

Based on the value of effective capacitances, calculated in (29), and using (17), dynamic power dissipation at different frequencies could be calculated.

TABLE V. CALCULATED AND ESTIMATED VALUES OF DYNAMIC POWER DISSIPATION.

Freq [MHz]	Pe_dyn [µW]	P_{dyn} [µW]	Accuracy
10.24	479.1	479.1	100.00 %
5.12	239.55	241.6	99.99 %
2.56	119.775	119.98	99.99 %
1.28	59.8875	59.48	99.99 %

Table IV contains estimated and measured values of dynamic power dissipation at different clock frequencies, in case one DSP core is running in idle state whilst others do not receive the clock.

Since uC is running in idle state, effective capacitance will be the same, over time, so (20) may derive into

$$P = I_{stat} \times V_{DD} + C_{eff} \times V_{DD}^2 \times f. \qquad (30)$$

Using (25), (29), and (30) overall power dissipation of uC whilst running in idle state at different clock frequencies can be calculated.

TABLE VI. CALCULATED AND ESTIMATED VALUES OF OVERALL POWER DISSIPATION.

Freq [MHz]	Pe [μW]	P_m [μW]	Accuracy
10.24	925	925.00	100.00 %
5.12	685.45	687.50	99.70 %
2.56	565.675	565.88	99.96 %
1.28	505.7875	505.38	99.92 %

Table V represents estimated – P_e, and measured – P_m values of overall power dissipation. It is obvious that estimation based on the presented methodology provides a high level of accuracy, for any value of clock frequency.

Using the measurement methodology, presented so far, effective capacitances of all DSP cores whilst running in idle state or executing NOP instruction have been determined. Those values are represented in Table VII.

TABLE VII. EFFECTIVE CAPACITANCES FOR IDLE AND NOP.

DSP	C_{IDLE}[pF]	C_{NOP}[pF]
uC	29.94	58.38
DSP0	7.42	32.19
DSP1	5.16	30.23
DSP2	6.09	27.03
DSP3	6.33	25.78

The next experiment should provide answers to the following questions:

1. What level of power dissipation estimation accuracy could be achieved during a multicore session, using different scenarios, expression (20), calculated value for static current (25), and values represented in Table VII.
2. What amount of energy, if any, could be preserved using the idle state instead of executing NOP instruction simultaneously.

In order to answer those questions, experiment has been conducted where DSP cores were in one of the following states: idle, nop, or reset, and based on the state, effective capacitance takes a value of: C_{IDLE}, C_{NOP}, or zero, successively, during the estimation.

Column "Acc" in Table VIII, represent estimation accuracy that has been achieved during the multicore experiment, which is quite high with average error rate of 0.34 %.

It is common for most embedded firmware applications to wait until an event occurs [28]. That could be achieved by executing NOP instruction within an endless loop or by putting the core into idle state. Figure 10 contains results for energy dissipation during a one clock cycle, for two different cases, when all DSP cores are in idle state, and when all DSP cores execute NOP instruction simultaneously. As it can be seen, the difference is significant, especially at the base frequency where energy consumption during NOP execution is more than twice higher (2.43 times) compared to idle state.

TABLE VIII. MEASURED AND ESTIMATED POWER DISSIPATION IN MULTICORE SESSION.

uC	DSP 0	DSP 1	DSP 2	DSP 3	P_m [μW]	P_E [μW]	Acc [%]
idle	idle	idle	idle	idle	1325.00	1322.54	99.81
idle	idle	idle	idle	nop	1635.00	1633.74	99.92
idle	idle	idle	nop	nop	1968.75	1968.78	100.00
idle	idle	nop	nop	nop	2367.50	2369.90	99.90
idle	nop	nop	nop	nop	2758.75	2766.22	99.73
nop	nop	nop	nop	nop	3215.00	3223.66	99.73
nop	nop	nop	nop	res	2807.50	2811.18	99.87
nop	nop	nop	res	res	2375.00	2378.70	99.84
nop	nop	res	res	res	1893.75	1895.02	99.93
nop	res	res	res	res	1396.25	1379.98	98.83
idle	res	res	res	res	933.75	922.54	98.80
idle	idle	res	res	res	1047.50	1041.26	99.40
idle	idle	idle	res	res	1128.75	1123.82	99.56
idle	idle	idle	idle	res	1226.25	1221.26	99.59
idle	nop	res	idle	nop	1947.50	1947.50	100.00

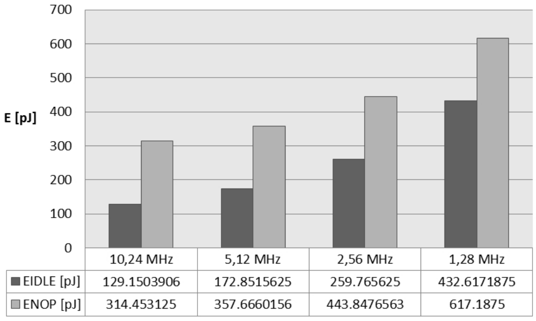


Fig. 10. Energy dissipation for NOP and IDLE.

Taking that into consideration, one should consider using idle state instead of simultaneously executing a NOP instruction, while waiting for an event, in order to develop an energy efficient firmware solution.

VI. CONCLUSIONS

This paper presents a research of power and energy dissipation of CMOS IC. A mathematical model and measurement methodology have been proposed that may provide an insight into static and dynamic power and energy dissipation. In order to validate the proposed solutions, various experiments have been conducted. Also, several general conclusions have been derived that may impact energy consumption during firmware development.

First, an approach has been proposed to defining dynamic energy over the well-defined term called effective capacitance, which represents a quantitative measure of dynamic energy.

Second, a unique mathematical model of average power dissipation (20) has been derived in function of: power supply voltage, static current, clock frequency, number of executed cycles, and effective capacitance of each individual cycle. In case all values are available, this model may provide cycle accurate average power dissipation of CMOS IC, whilst executing software application [1].

Third, unique measurement methodology for obtaining static and dynamic power of CMOS IC using different clock frequencies has been presented (24). A patent application has been created for this methodology [8]. Experimental results presented in Table III, and expressions (26) and (27) confirm this methodology.

Fourth, by conducting two different experiments, at four different clock frequencies, two general conclusions have been derived and confirmed:

1. When core is running in idle state, frequency should be decreased as much as possible, in order to reduce dynamic energy dissipation (Fig. 8).
2. When core is in execution mode, for example executing FIR filter implementation, frequency should be increased as much as possible, in order to reduce static energy dissipation (Table IV).

Fifth, various experiments have been conducted using different clock frequencies (Table V, and Table VI), as well as using different scenarios (Table VIII), in order to determine the level of power estimation accuracy that can be achieved using the proposed mathematical model (20) and measurement methodologies. From Table V, Table VI, and Table VIII, it can be calculated that estimation accuracy average error rate does not exceed 0.34 %.

Sixth and final, an experiment and comparative analyses about energy consumption when the core is in idle mode, and whilst simultaneously executing NOP instruction have been conducted, which are two common approaches when application is waiting for an event to occur. Experimental results, Fig. 10, unequivocally confirm that idle state consumes far less energy, 2.43 times at the base clock frequency, than NOP execution; therefore, one should use idle state instead of NOP instruction in order to develop an energy efficient firmware solution.

Future work will focus on instruction set and peripherals. The idea is to obtain effective capacitances for each individual instruction, inter-instruction effect, and peripheral, and to integrate those values into the mathematical model (20). That way, accurate energy consumption estimation of software application at any clock frequency should be allowed.

REFERENCES

[1] M. V. Krunic, M. V. Popovic, V. M. Krunic, N. B. Cetic, "Energy consumption estimation for embedded applications", *Elektronika ir Elektrotechnika*, vol. 22, no. 3, pp. 44–49, 2016. [Online]. Available: https://doi.org/10.5755/j01.eie.22.3.15313

[2] M. Krunic, I. Povazan, M. Popovic, J. Kovacevic, "Data flow CAD fool for Firm1w are development and power consumption estimation in multi-core hearing aids", in *IEEE Int. Conf. Consumer Electronics (ICCE 2016)*, Las Vegas, NV., 2016. [Online]. Available: https://doi.org/10.1109/ICCE.2016.7430736

[3] UPC, "Basic CMOS concepts", 2016. [Online]. Available: http://docencia.ac.upc.edu/master/MIRI/NCD/assignments/Tema%20 1-EN.pdf

[4] R. J. Baker, *CMOS Circuit Design, Layout, and Simulation*. IEEE PRESS, 2011.

[5] I. Povazan, M. Krunic, M. Popovic, "A profiling tool for heterogeneous multi-core systems", in *ECBS-EERC 2015*, Brno, Czech Republic, 2015. [Online]. Available: https://doi.org/10.1109/ ECBS-EERC.2015.31

[6] V. Tiwari, S. Malik, A. Wolfe, "Power analysis of embedded software: A first step towards software power minimization", *IEEE Trans. VLSI Systems*, vol. 2, no. 4, pp. 437–445, 1994. [Online]. Available: https://doi.org/10.1109/92.335012

[7] N. Sung, T. Austin, T. Mudge, D. Grunwald, "Challenges for architectural level power modeling", in *Power aware computing*, New York: Springer US, 2002, pp. 317–338.

[8] M. Krunic, N. Cetic, M. Popovic, J. Kovacevic, "Methodology for measuring the static power dissipation of embedded DSP platform", *Serbia Patent P-2016/0787*, 20 09 2016.

[9] G. Panic, "Power Consumption", in *A Methodology for Designing Low Power Sensor Node Hardware Systems*, Senftenberg, Cottbus-Senftenberg, 2014, pp. 11–12.

[10] P. Joshi, N. Kumari, K. Gurumurthy, "Instruction level power analysis for low power VLSI applications", in *6th Int. Conf. Emerging Trends in Engineering and Technology (ICETET 2013)*, Nagpur, 2013. [Online]. Available: https://doi.org/10.1109/ICETET.2013.12

[11] M.-C. Lee, V. Tiwari, S. Malik, M. Fujita, "Power analysis and low-power scheduling techniques for embedded DSP software", in *Proc. Eighth Int. Symposium on System Synthesis*, Cannes, 1995. [Online]. Available: https://doi.org/10.1109/ISSS.1995.520621

[12] S. Nikolaidis, T. Laopoulos, "Instruction-level power consumption estimation embedded processors low-power applications", in *Int. Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Crimea, 2001. [Online]. Available: https://doi.org/10.1109/IDAACS.2001.941998

[13] R. Basmadjian, H. de Meer, "Evaluating and modeling power consumption of multi-core processors", in *Third Int. Conf. Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy)*, Madrid, 2012. [Online]. Available: https://doi.org/ 10.1145/2208828.2208840

[14] V. Konstantakos, A. Chatzigeorgiou, S. Nikolaidis, T. Laopoulos, "Energy consumption estimation in embedded systems", in *Instrumentation and Measurement Technology Conference (IMTC 2006)*, Sorrento, Italy, 2006. [Online]. Available: https://doi.org/ 10.1109/IMTC.2006.328405

[15] S. Nikolaidis, N. Kavvadias, P. Neofotistos, "Base instruction cost measurements", in Instruction level power measurements and analysis, Thessaloniki, *Energy-Aware SYstem-on-chip design of the HIPERLAN/2 standard*, 2002, pp. 14–15.

[16] S. Nikolaidis, N. Kavvadias, P. Neofotistos, "Inter-instruction effect cost measurements", in Instruction level power measurements and analysis, Thessaloniki, *Energy-Aware SYstem-on-chip design of the HIPERLAN/2 standard*, 2002, pp. 15–16.

[17] N. Kavvadias, P. Neofotistos, S. Nikolaidis, K. Kosmatopoulos, T. Laopoulos, "Measurements analysis of the software-related power consumption of microprocessors", *IEEE Trans. Instrumentation and Measurement*, vol. 53, no. 4, pp. 1106–1112, 2004. [Online]. Available: https://doi.org/10.1109/TIM.2004.830784

[18] M. Bazzaz, M. Salehi, A. Ejlali, "An accurate instruction-level energy estimation model and tool for embedded systems", *IEEE Trans. Instrumentation And Measurement*, vol. 62, no. 7, pp. 1927–1934, 2013. [Online]. Available: https://doi.org/10.1109/TIM.2013.2248288

[19] B. Klass, D. E. Thomas, H. Schmit, D. F. Nagle, "Modeling interinstruction energy effects in a digital signal processor", in *Proc. Digital Signal Processor, Power-Driven Microarch. Workshop in Conjunction with Int. Symp. Comput. Arch*, Barcelona, Spain, 1998.

[20] G. Callou, P. Maciel, E. Tavares, E. Andrade, B. Nogueira, C. Araujo, P. Cunha, "Energy consumption and execution time estimation of embedded system applications", *Microprocessors Microsyst.*, vol. 35, no. 4, pp. 426–440, 2011. [Online]. Available: https://doi.org/ 10.1016/j.micpro.2010.08.006

[21] R. Oshana, "Power optimization techniques using DSP", in *DSP Software Development Techniques for Embedded and Real-Time Systems*, 2006, pp. 234–242.

[22] LSU, "School of Electrical Engineering and Computer Science Louisiana State University", 2016. [Online]. Available: http://www.ece.lsu.edu/ee4720/2012/lsli02.pdf

[23] RIT, "Computer Engineering", Rochester Institute of Technology, 2016. [Online]. Available: http://meseec.ce.rit.edu/eecc550-winter2011/550-12-6-2011.pdf

[24] UMN, "The University of Minnesota", 2016. [Online]. Available: https://www.d.umn.edu/~gshute/arch/performance-equation.xhtml

[25] G. Panic, "Static power loss", in *A Methodology for Designing Low Power Sensor Node Hardware Systems*, Senftenberg, Cottbus-Senftenberg, 2014, pp. 15–16.

[26] D. Mirkovic, D. Milovanovic, "Istrazivanje metoda projektovanja analognih i integrisanih kola sa mešovitim signalima u uslovima velikih struja curenja i temperaturskog drifta nanometarskih tehnologija", 2011. (in Serbian)

[27] D. M. Popovic, "Disipacija CMOS kola", in *Osnovi elektronike*, Beograd, Elektrotehnicki fakultet Beograd, 2006, pp. 112–113.

[28] R. Oshana, "Event-driven applications", in *DSP Software Development Techniques for Embedded and Real-Time Systems*, Oxford, Embedded Technology, 2006.