Imperial Journal of Interdisciplinary Research (IJIR)
Vol-3, Issue-4, 2017
ISSN: 2454-1362, http://www.onlinejournal.in

Survey on Power and Energy Management in Mobile Phones

Anusha S K[1] & Mrs. Geetha V[2]
[1]Dept. of ISE, R. V. College of Engineering, Bengaluru, India
[2]Asst. Professor, Dept. of ISE,
R. V. College of Engineering, Bengaluru, India

Abstract— *Smart mobile phones are becoming increasingly popular because of the enormous features it provides. The urge to have slimmer and faster device has led to several limitations, one of them being power and energy dissipation. Therefore effective management of power and energy has become a major concern.*

1. Introduction

Deep sub-micron technology has made a rapid progress in this era. Whatever fitted on boards earlier, fits on chip nowadays(SoC). This rapid development led to the growth of various electronic appliances especially wireless devices such as mobile phones. A single device is expected to act as a camera, music and video player, game station, navigator, call placer, message sender etc. Resource management in such devices have to be done carefully without hampering the features it provides. However, to meet all the requirements that it is supposed to cater, the device may be subjected to high power consumption which inturn leads to reduced battery life. Therefore power has to be managed effectively for optimised battery life.

Power consumption may be broadly classified into two types. They are static and dynamic.

2. Static and Dynamic Power

Static power is the power dissipated during steady state. For a given circuit, if I_{leak} stands for leakage current and V_{DD} stands for supply voltage, the static power is given by:

$$P_{stat} = I_{leak} * V_{DD}$$

It depends purely on supply voltages. There is almost nil activity in the circuit.

Dynamic power is the power dissipated when all the parts are active. It comprises of power consumed during the charging and discharging of capacitors. If C_{dyn} is the switching capacitance, f represents the clock frequency and α is switching activity, the dynamic power is given by:

$$P_{dyn} = \alpha * C_{dyn} * V_{DD}^2 * f$$

Total power dissipation is the summation of static and dynamic which is given by:

$$P_{total} = P_{stat} + P_{dyn}$$

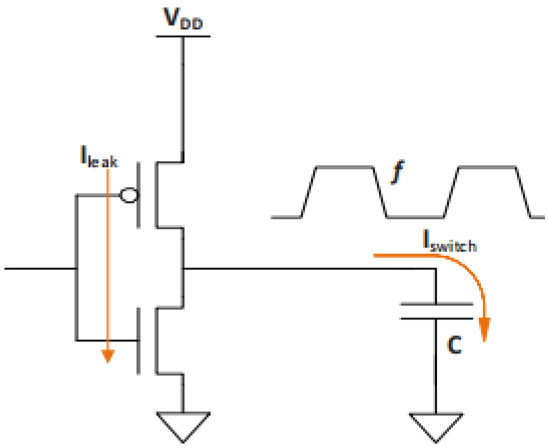


Figure: Inverter set up.

3. Literature Survey

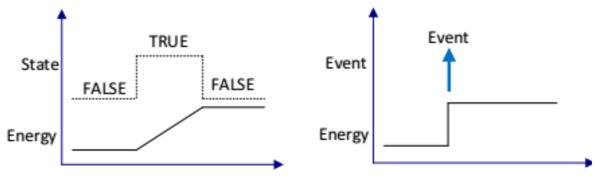
Power consumption models have been distinguished into two types namely, low level power modelling and high level power modelling.

Low level power modelling calculate the power from electrical descriptions such as circuit level, gate level and RTL level. Circuit level considers the system in terms of transistors which are very complex. Therefore, it consumed large simulation time and required heavy effort.

High level power modelling deals with instruction and functional units of the program to calculate power consumption.
The current drawn while executing each processor instruction was measured in instruction level. The drawback was that, it required large memory to store

the current measurement for each instruction in the ISA.
The basic idea of functional level power modelling is to identify the functional blocks that influence the power consumption.

Another modelling technique is based on events and state. In case of state based technique, all the functional units are assumed to be a particular state. A change in state will affect the power consumption. In case of event based technique, any event leads to power consumption.



State based Event based

Figure: State based and event based modeling

Modelling techniques were designed based on three levels. They are: system, application and user levels. Let us see the categories in detail in the following sections.

A. System Level

The model is designed to calculate the power of entire system at a given time. Power computation can be done either by inserting an external device or by monitoring the system APIs. In the former, the behavior of the mobile device under various external conditions is captured. In the later, the software APIs gather battery related information. Studies suggest that power consumption is calculated more accurately in former case.

B. Application Level

The model is designed based on per application power consumption. This method is more detailed compared to the system level. A tool is proposed in [1] which measures power in various applications. It predicts power consumed by an application in the development phase. At pre-silicon level, this tool called SPOT (System Power Optimization Tool) is used for power calculation. SPOT generates the power estimates which has error of 3-4% of the actual power measured on a device. The only limitation is that this tool is not compatible with android SDK.

C. User Level

The model is designed based on the usage pattern. According to the survey made by [2], an application was designed which gathers users information. The results showed that the usage of a particular application differed from each individual. This led to the raise of power models

In [3], a power model was created considering the usage pattern. They consider a mobile device having a certain number of states, these states were put into a tuple with a condition on phone components. The conclusion drawn was that different users spend different time in each state.

In [4], a daily and weekly user profile was created, reflecting daily chores, i.e. sleeping, exercising, eating, relaxing, working etc.; for both weekdays and weekends. For every activity, the battery consumption and time spent in each state was measured. After which, a log file containing the measurements was processed by a server which was the starting point of model build.

4. Issues

Various categories of models were analyzed. The issue with respect of each of them are as follows. In case of power calculation by external device, though the result is more accurate, the process is more static and hence it is less adapted. Therefore, APIs came into picture which are more adaptable. In case of user level modelling, it is difficult to make assumptions of the user making use of a particular functionality. Since the requirements of the user keeps changing from time to time, this method may not be feasible.

5. Power Management for Network Interfaces(NI)

The workload of a system is not constant. There are active and idle periods. With appropriate mechanisms, it is possible to reduce power consumption may be by turning off the device etc. However, if the workload is again changed to active state in the very next second, then energy consumption will be doubled the conserved energy. In [5], USB-Wi-Fi adapter was used. Current and voltage were measured and NI model was obtained. The setup of the experiment is shown in the following figure.

Imperial Journal of Interdisciplinary Research (IJIR)
Vol-3, Issue-4, 2017
ISSN: 2454-1362, http://www.onlinejournal.in

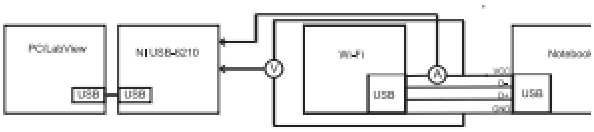


Figure: USB-Wi-Fi Adapter.

The power consumed power according to their experiment is shown in the following figures.
If P_{on} is on state of NI and P_{off} is the off state of NI. Power reduction is calculated as the difference between P_{on} and P_{off} given by the formula,

$$P_{red} = P_{on} - P_{off}$$

The average power of both the states as well as power reduction is shown in the table.

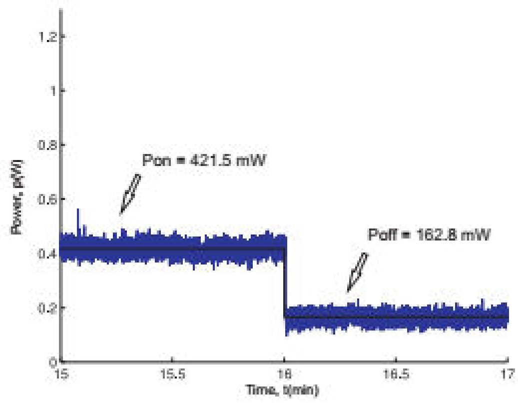


Figure: P_{on} represents the network interface when it is turned on and idle, P_{off} represents network interface is turned off.

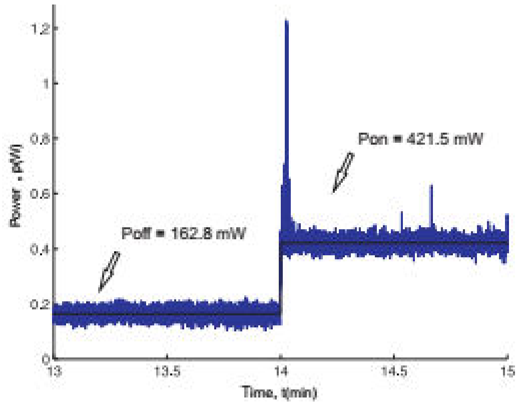


Figure: P_{off} represents the network interface when it is turned off and idle, P_{on} represents network interface is turned on.

	Power consumption (mW)
P_{on}	421.5
P_{off}	162.8
Power reduction($P_{on} - P_{off}$)	258.7

Table: P_{on} and P_{off} values in milli Watts.

6. Methodology

Performance of several workloads have been experimented and comparison of the simulation is made by adopting four power policies namely, Time out [7], L Curve [6], Predictive [8] and Predictive without automatic turn out. These policies were executed using power management software.

The block diagram of simulation is shown below.

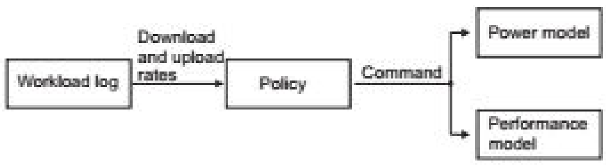


Figure: Block diagram of the simulation

A. *Time out Policy*

According to Luiz [9], this policy is based on timeout value represented by δ. The timeout value must reduce the power consumption as well as maintain the performance error without exceeding the limit. When the limit is exceeding, it is automatically turned off. If the user interface is needed again for usage, then the NI has to be turned on.

B. *Predictive policy*

According to Hwang [8], the estimation of next idle time is based on the average of previous idle state estimates. The formula to calculate estimate is given by [5]

$$In+1 = a\ In + (1 - a)In$$

α ranges between 0 and 1. Since it is predictive policy, it has lower performance penalties compared to other policies.

C. *L Curve Policy*

According to L curve policy proposed by Lu and Micheli, [6], the NI is turned off based on active interval rather than the idle interval of the workload. However, this policy applies only to those workloads whose active versus idle curve is in L shape.

D. Predictive Policy without Automatic Turn On

Though this policy aims in providing less performance penalty, power consumption is however increased. The reason behind this is due to the NI being turned on too early. In this case, the time interval between the NI being turned on and the arrival of new user request may be greater than the interval at which the NI is off.

7. Results Inferred

According to the simulation carried out by [5], following results are inferred.
In case of Time out policy, performance penalty and time out value are inversely proportional. If the timeout value is increased, the performance penalty is decreased. The reason being, long time out leads to few shutdowns and hence less time interval when user will not be able to use the NI.

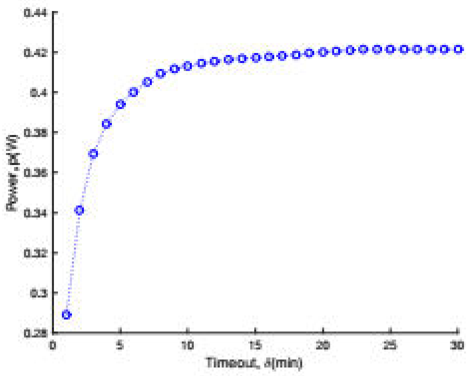


Figure: Time out simulation result

In case of predictive policy, as the value of α increases, the estimated idle interval decreases. The simulation was carried out with sequential increase in α value ranging from 0.1 to 1.0.

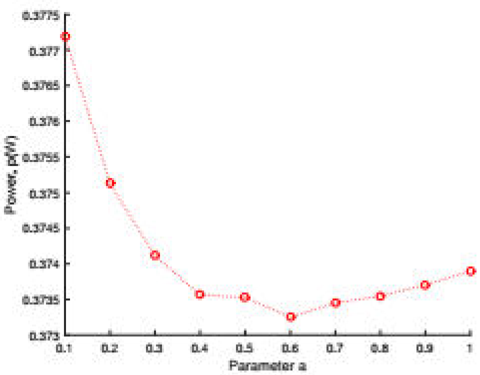


Figure: Predictive simulation result

In case of L curve policy, with the increase in threshold, there is decrease in power. Figure shows performance penalty as a function of threshold.

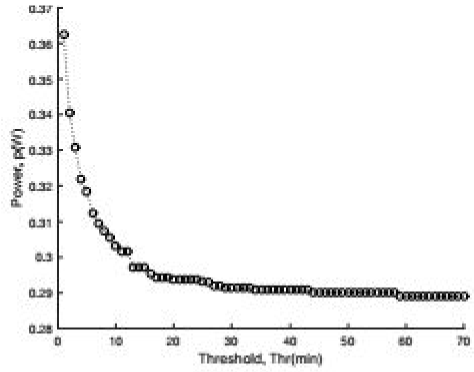


Figure: Lcurve simulation.

In case of predictive policy without automatic turn on policy, α predicts when the device has to be turned off. However, it does not influence the time at which the interface has to be turned off. Here, the performance penalty is higher and power consumption is lower.

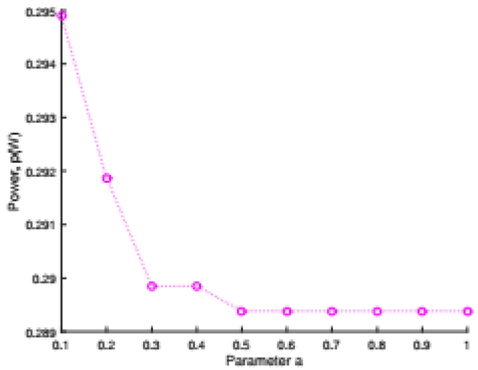


Figure: Predictive policy with automatic turn on simulation result.

The energy savings for each policy is shown in the table below.

Policy	Maximum power saving	Performance penalty
Timeout	31.46%	21.12%
L Curve	31.44%	21.34%
Predictive without automatic turn on	31.34%	20.92%
Predictive	11.44%	4.18%

Table: Power conserved

8. Conclusion

In this survey, power consumption models have been discussed based various categories. Four simulation policies were realized and behavior of these policies were analyzed.

9. References

[1] M. Ferroni, "On Power and Energy Consumption Modelling for Smart Mobile Devices". In *2014 International Conference on Embedded and Ubiquitous Computing*.

[2] Narseo Vallina-Rodriguez, Pan Hui, Jon Crowcroft, and Andrew Rice. Exhausting battery statistics: understanding the energy demands on mobile handsets. In *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds.*

[3] Joon-Myung Kang, Sin seok Seo, and J.W.-K. Hong. Usage pattern analysis of smartphones. In *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*, pages 1 –8, sept. 2011.

[4] Joon-Myung Kang, Sin seok Seo, and James Won-Ki Hong. Personalized battery lifetime prediction for mobile devices based on usage patterns. *Journal of Computing Science and Engineering*, 5(4):338–345, 2011.

[5] Bruna M. J. C, "Comparison of dynamic power management policies for Wi-Fi interfaces". *In Industry Applications (INDUSCON), 2016 12th IEEE International Conference.*

[6] Y.-H. Lu and G. De Micheli, "Comparing system level power management policies," *IEEE Design & Test of Computers*, vol. 18, no. 2, pp.10–19, 2001.

[7] B. Vilar, F. De M Silva, S. Luiz, J. Silva, H. Almeida, and A. Perkusich, "Dynamic timeout power policy for network interfaces," in *Proc. Conf. (ICCE-Berlin), 2014*, Sept 2014, pp. 279–283.

[8] C.-H. Hwang and A. C.-H. Wu, "A predictive system shutdown method for energy saving of event-driven computation," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 5, no. 2, pp. 226–241, 2000.

[9] S. Luiz, A. Perkusich, B. Cruz, B. Neves, and G. da S Araujo, "Optimization of timeout-based power management policies for network interfaces," *Consumer Electronics, IEEE Transactions on*, vol. 59, no. 1, pp. 101–106, Feb 2013.