UCLA
UCLA Electronic Theses and Dissertations

Title
Power Simulator for Smartphones

Permalink
https://escholarship.org/uc/item/7z1502c5

Author
Gkolemis, Nikolaos

Publication Date
2013

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA

Los Angeles

Power Simulator

for Smartphones

A thesis submitted in partial satisfaction

of the requirements for the degree Master of Science

in Electrical Engineering

by

Nikolaos Gkolemis

2013

ABSTRACT OF THE THESIS

Power Simulator

for Smartphones

by

Nikolaos Gkolemis

Master of Science in Electrical Engineering

University of California, Los Angeles, 2013

Professor He Lei, Chair

The ability to accurately estimate the power and energy dissipation of a smartphone, based on different running applications, has been a crucial demand for both mobile companies and mobile users. Pre-existing solutions are either calibrated for specific devices or lack accuracy due to the use of oversimplified power models or operating restrictions. The developed power simulator for

smartphones described here is the first complete attempt to combine generalized power models together with sequential and parallel application execution algorithms and a user friendly graphical user interface in order to create a power and energy estimation tool that can be used by both researchers and smartphone users to accurately simulate real mobile usage profiles and examine their impact on power and energy consumption.

The thesis of Nikolaos Gkolemis is approved.

Danijela Cabric

William J. Kaiser

Lei He, Committee Chair

University of California, Los Angeles

2013

I would like to thank Professor Lei He and Feng Shi for their advice and guidance during development.

Table of Contents

Chapter 1: Introduction 1

1.1 Objectives and Contributions 1

Chapter 2: Previous Work 4

Chapter 3: Motivation for Tool Development 6

Chapter 4: Power Models of Smartphone's Hardware Modules 8

4.1 Applications Processor (CPU) 10

4.2 Graphics Processor (GPU) 12

4.3 3G Modem – 4G/LTE Modem 14

4.4 Wifi 16

4.5 Low Power DDR2 Memory 18

4.6 Audio Codec 19

4.7 GPS 20

4.8 LCD Screen 21

Chapter 5: Application Power Profiles 24

5.1 Standby Application 24

5.2 Interaction Application 26

5.3 Music Playing Application 27

5.4 Video Streaming (Wifi) Application 29

5.5 Calling Application 32

5.6 Messaging Application 34

5.7 Browsing (Wifi) Application 37

5.8 Browsing (Cellular) Application 39

5.9 Navigation (Cellular) Application 42

5.10 Gaming (Offline) Application 46

Chapter 6: Parallel Execution Algorithm 49

6.1 Motivation for Algorithm Development 49

6.2 Parallel Execution Capabilities of Modeled Applications 49

6.3 Description of Algorithmic Steps 51

Chapter 7: Power Simulator's User Interface 53

7.1 Available Configurations 54

7.2 Application Selection 58

7.3 Simulation Phase 61

7.4 Reported Results 61

Chapter 8: Demonstration of Power Simulator 66

8.1 1st Usage Profile – Sequential Execution 66

8.2 2nd Usage Profile – Parallel Execution 74

Chapter 9: Conclusion and Future Work 85

9.1 Conclusion..... 85

9.2 Future Work 87

References..... 88

Table of Figures

Figure 1: Iphone 3GS Block Diagram (source: phoneWreck)..... 8

Figure 2: Power Profile of Standby Application..... 25

Figure 3: Powel Profile of Interaction Application..... 27

Figure 4: Power Profile of Music Playing Application 29

Figure 5: Power Profile of Video Streaming Application 32

Figure 6: Power Profile of Calling Application..... 34

Figure 7: Power Profile of Messaging Application 36

Figure 8: Power Profile of Browsing (Wifi) Application 39

Figure 9: Power Profile of Browsing (Cellular) Application..... 42

Figure 10: Power Profile of Navigation (Cellular) Application 45

Figure 11: Power Profile of Gaming (Offline) Application..... 47

Figure 12: Demonstration of Hardware Configuration Selection Window 54

Figure 13: Modules of 1st Hardware Configuration..... 55

Figure 14:Modules of 2nd Hardware Configuration 55

Figure 15: Modules of 3rd Hardware Configuration 56

Figure 16:Demonstration of GUI Window after Hardware Configuration Selection.. 57

Figure 17: Demonstration of GUI Window after the Custom Configuration Option is Selected 57

Figure 18: Demonstration of Selection of Various Hardware Modules Under the Custom Configuration Option..... 58

Figure 19: Demonstration of Applications Selection and Runtime Period Setup 59

Figure 20: Demonstration of Error Message Display for Non-Parallel Allowed Application Execution Attempt 60

Figure 21: Demonstration of Error Message Display for 3 Parallel Applications Execution Attempt 60

Figure 22: Result Window – Demonstration of Maximum Power per Hardware Module 62

Figure 23: result Window – Demonstration of Energy Consumed per Application.... 62

Figure 24: Result Window – Demonstration of Overall Simulation Power Over Time Plot 63

Figure 25: Result Window - Demonstration of Application's Power Over Time Plot 64

Figure 26: Configuration Selection for the 1st Usage Example 66

Figure 27: Application Selection and Parameter Setting for the 1st Usage Example . 67

Figure 28: Maximum Power per Hardware Module for the 1st Usage Example 69

Figure 29: Overall Simulation Power Over Time Plot for the 1st Usage Example..... 70

Figure 30: Energy Dissipation per Application for the 1st Usage Example..... 70

Figure 31: Power Over Time and Average Hw Modules Power for a unique application for the 1st Usage Example 71

Figure 32: Power Over Time and Average Hw Modules Power for Selected Application for the 1st Usage Example 71

Figure 33: New Maximum Power per Hw Module for the 1st Usage Example..... 72

Figure 34: New Overall Simulation Power Over Time Plot for the 1st Usage Example 72

Figure 35: New Power over Time and Average Hw Modules Power Results for Selected Application for the 1st Usage Example 73

Figure 36: Configuration Selection for the 2nd Usage Example 74

Figure 37: Application Selection and Parameter Setting for the 2nd Example 75

Figure 38: Maximum Power per Hw Module for the 2nd Usage Example 81

Figure 39: Overall Simulation Power Over Time Plot for the 2nd Usage Example.... 82

Figure 40: Energy Dissipation per Application for the 2nd Usage Example 82

Figure 41: Power Over Time and Average Hw Module Power for a Selected Application for the 2nd Usage Example 83

Figure 42: Power Over Time and Average Hw Modules Power for a Selected Application for the 2nd Usage Example 83

Figure 43: Power Over Time and Average Hw Modules Power for a Selected Application for the 2nd Usage Example 84

List of Tables

Table 1: Application Processor Power Model Parameters 12

Table 2: GPU Power Model Parameters 14

Table 3: 3G - 4G/LTE Modem Power Model Parameters 16

Table 4: Wifi Power Model Parameters 17

Table 5: Low Power DDR2 Memory Power Model Parameters 19

Table 6: Audio Codec Power Model Parameters 20

Table 7: GPS Power Model Parameters 21

Table 8: LCD Screen Power Model Parameters 23

Table 9: Standby Application's Hw Modules Operation Mode per Power State..... 25

Table 10: Interaction Application's Hw Modules Operation Modes per Power State. 26

Table 11: Music Playing Application's Hw Modules Operation Mode per Power State 28

Table 12: Video Streaming Application's Hw Modules Operation Mode per Power State 31

Table 13: Calling Application's Hw Modules Operation Mode per Power State 34

Table 14: Parallel Execution Capabilities of Modeled Applications 50

Table 15: Execution Time Window Division per Power State of a Sequential Application of the 1st Usage Example 68

Table 16: Execution Time Window Division per Power State of a Sequential Application of the 1st Usage Example 69

Table 17: Division of Simulation Time in Sequential and Parallel Execution Time Windows for the 2nd Usage Example 76

Table 18: Time and Power State Matrix of one Parallel Executed Application in the 1st Parallel Region of the 2nd Usage Example 76

Table 19: Time and Power State Matrix of one Parallel Executed Application in the 1st Parallel Region of the 2nd Usage Example 77

Table 20: Unified Time and Power State Matrix of the 1st Parallel Region of the 2nd Usage Example 77

Table 21: Time and Power State Matrix of a Parallel Executed Application in the 2nd Parallel Region of the 2nd Usage Example 78

Table 22: Time and Power State Matrix of a Parallel Executed Application in the 2nd Parallel Region of the 2nd Usage Example 79

Table 23: Unified Time and Power State Matrix of the 2nd Parallel Region of the 2nd Usage Example 79

Table 24: Frequency and Utilization Ratio Results of Parallel Execution for Time Areas of the 1st Parallel Region of the 2nd Usage Example 80

Table 25: Frequency and Utilization Ratio Results of Parallel Execution for Time Areas of the 2nd Parallel Region of the 2nd Usage Example 81

Chapter 1: Introduction

Smartphones, which are the most popular mobile devices, have rapidly replaced the conventional feature phones thanks to their high performance and various functionalities. They revolutionized the entire phone business and became a crucial aspect of our everyday lives. While year-by-year smartphones become more powerful and pack even more features, one thing that didn't evolve so fast was the phone's power source (battery). With the mobile demand for power constantly growing and battery capacity of mobile devices being limited, creating ways to accurately monitor and understand the power behavior of the smartphone's hardware modules and applications can help researchers develop more power efficient phones and can assist users in planning their phone use accordingly to ensure maximum battery life.

1.1 Objectives and Contributions

This thesis aims to provide a universal smartphone power and energy estimation tool based on simulated mobile application profiles. This tool, known as SPEED (Smartphone Power & Energy Estimation Delivered), will be capable of simulating the power behavior of any smartphone by using generalized power models to describe the power dissipation of each of the hardware modules found in a modern smartphone. Using SPEED the user can create realistic mobile usage profiles, combining any number of the available applications in a sequential or parallel execution manner, simulate them and then analyze the reported profile's

requirements in power and energy. The tool is composed of four parts and the major contributions of each part are listed below:

- Power Models for Smartphone's Hardware Modules
  - Power modeling of every major hardware module contained in modern smartphones.
  - Generalized power models based on parameter value setting. The user can simulate the power behavior of any existing hardware module by setting the appropriate parameter values.
  - Small number of user defined parameters to ensure ease of use.
  - Tuning of models to guarantee reliable results.

- Mobile Application Profiles
  - Power modeling of ten of the most frequently used mobile applications.
  - Creation of different power states within the applications to accurately model the variation in power dissipation during the application's lifetime.

- Parallel Execution Algorithm
  - Development of algorithm that calculates the total power of parallel executed applications - without overestimating power -, by taking into account the Dynamic Voltage and Frequency Scaling (DVFS) capabilities of modern processors.

- Simulator's Graphical User Interface Design
  - Design of user friendly GUI that makes it easy for the user to set parameter values for hardware modules and set up mobile usage profiles by selecting the desired applications and their execution period.
  - Display of various power results (max active power of each of the smartphone's hardware modules, power dissipation over time for the entire simulation time window, power profiles together with average hardware modules' consumed power for each of the selected applications).
  - Display of energy results (total consumed energy during the simulation time window, energy consumption of each of the selected applications).

Chapter 2: Previous Work

Smartphone's power estimation has been the focus of both academic and industry research work the past few years. Earlier work – like the one seen on [1] and [2] - focused on the development of power models for the hardware modules of specific devices. These models were derived based on physical power measurements conducted while a variety of single applications were running on the phone under test. Both the voltage and current were measured using a voltmeter and a current sense resistor for each module. A specific power model was then constructed for each module using coefficient values that would lead to the same power value for the module as the one measured for each application.

These models even though they were pretty accurate for the specific devices measured, couldn't be used as generalized power models due to the significant variation in power dissipation of a module's implementations between different vendors or different generations. Work on [4] ignored power modeling and using the same physical measuring techniques focused on the breakdown of power between different modules for a variety of frequently used mobile applications.

Later work tried to focus more on the creation of generalized power models for single hardware modules. Authors in [9], [10], [11] and [12] investigated the power behavior of 3G and 4G/LTE Modems and proposed detailed power models for both of them. [6], [7] and [8] all examine various SDRAM implementations and propose parameterized mobile DRAM memory power models. Work in [5] presents a simplistic power model for GPUs based on physical power measurements in a variety of phones. In [10] researchers focus on the power

modeling of the 3G Modem and the Wifi module, while authors in [11] analyze earlier proposed analytic power models for the Wifi, 3G and 4G/LTE Modems and propose an easier to use universal power model for all these three hardware modules with a small set of parameters. Finally [3] proposes a more accurate and generalized power model for CPUs considering the number of cores and the DVFS abilities of modern processors. It also presents a novel non-linear power model for the LCD Screen while adopting earlier proposed power models for the rest of the smartphone's hardware modules.

Another large area of focus for previous work has been the creation of power measuring tools that can be installed in smartphones as applications and can calculate the total power consumption in real time. Some of these tools also display a per hardware module power estimation (Google's PowerTutor). Real time power monitoring tools can predict the overall system's power by calculating the power of the smartphone's hardware modules. This technique relies on the collection of the parameter values of each of the smartphone's hardware modules in realtime. These tools only appeared recently and even though they promise real time power tracing, they often severely over/underestimate power consumption or remain non-operational in a large number of smartphones for reasons that are discussed in the next section.

Chapter 3: Motivation for Tool Development

My motivation for this work was to create a universal smartphone simulator that can accurately estimate the power and energy dissipated based on different application profiles. As described in the Previous Work section, earlier work on power modeling of the entire smartphone was mostly focused on creating models for specific phones - work in [1] and [2]-. These models, despite being accurate for the specific phones they were created for, cannot be applied for every smartphone since hardware configurations and technology can differ significantly between older and newer smartphones or even between same generation phones. Work in [3] is the only previous attempt following a more universal approach, with many of their power models also adopted in this thesis with minor changes where deemed necessary. But while some of their hardware modules' power models are the most accurate yet (CPU, LCD Screen), others lack substantial parameters (3G Modem), or are even completely ignored (GPU, 4G/LTE Modem).

More importantly, all of the above mentioned papers and [4] demonstrate their power models using a limited set of sequentially executed applications. No parallel execution is being demonstrated in any of these papers, even though it takes place in the majority of the time when we are using our smartphones. On top of that, all previous work only demonstrates average power consumption for each of the smartphone's hardware modules during the application's execution, failing to identify and display the changes in power between different power states within the application's runtime.

Finally real time power measuring tools like PowerTutor, discussed in the Previous Work section, may not be very effective and accurate since each hardware module manufacturer may be using a different set of parameters that are not in compliance with the parameters used by the tool's power models. To make things even worse, mobile operating systems like Apple's OS and Blackberry's OS don't allow the collection of hardware module parameters for security reasons, making real time power measuring tools non operational in every smartphone that runs one of these operating systems. Even though Google's Android software supports the monitoring and collection of these parameters, these tools often fall short, under/over estimating the power of specific modules (etc CPU, GPU, Modem 4G/LTE), while reporting more accurate values for other modules (etc LCD Screen, GPS). - Above results were extracted by using Google's PowerTutor on a Samsung's Galaxy SII smartphone running Android software -.

The current work tries to create a novel smartphone power and energy estimation tool that deals with all the disadvantages of previous work mentioned above, while ensuring ease of use and expandability.

Chapter 4: Power Models of Smartphone's Hardware Modules

Each one of the hardware modules that consist a smartphone was modeled using a specific equation that could best describe the power behavior of the specific module. These equations were derived after analyzing and comparing previously proposed power models found in a variety of published papers. Finally the tunable parameters available to the end user in each module's power model were selected in a way that would make the model user friendly and simple but yet still accurate.

An example of the hardware modules consisting the interior of a modern smartphone can be seen in the picture below:

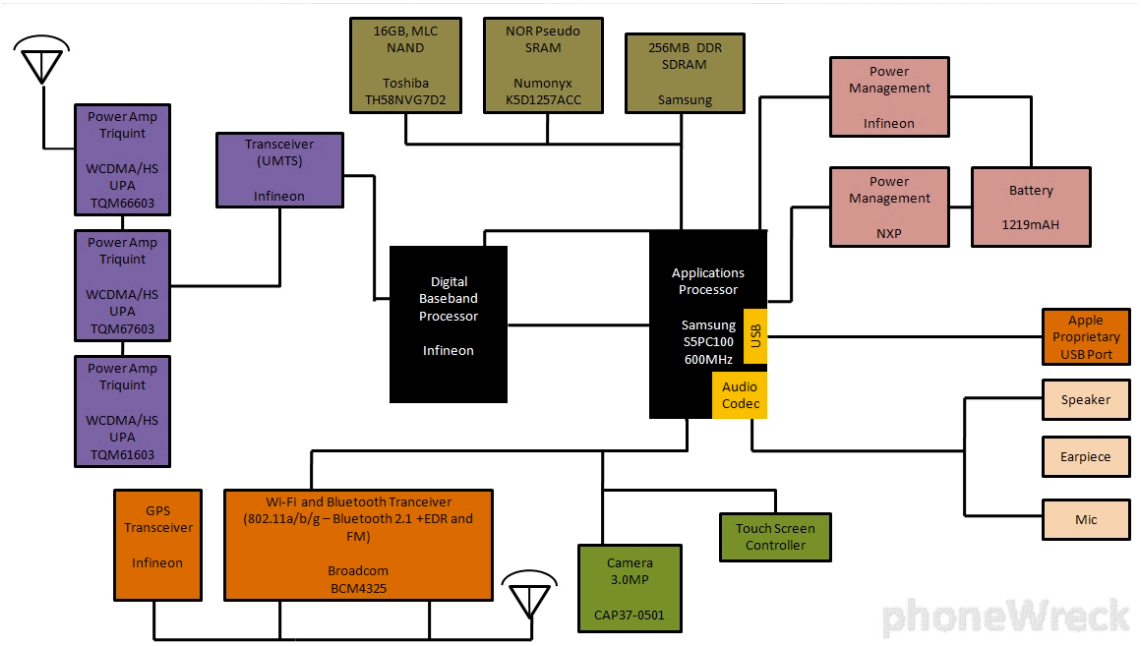

Figure 1: Iphone 3GS Block Diagram (source: phoneWreck)

For the proposed smartphone simulator described in this thesis, power models were created for the following hardware modules:

- Applications Processor (CPU)
- Graphics Processor (GPU)
- 3G Modem (including Baseband Processor)
- 4G/LTE Modem (including Baseband Processor)
- WiFi
- Low Power DDR2 Memory
- Audio Codec (including Codec, DSP and Power Amplifier)
- GPS
- LCD Screen

Each of the above power models is presented in detail bellow:

4.1 Applications Processor (CPU)

The most important module of every phone is it's CPU or Applications Processor. The CPU is the "heart" of every computation that takes place in our smartphone. As a result it's also one of the most power intensive modules.

Previous work on CPU power modeling seen in [1] and [2] fails to accurately model the power characteristics of modern CPUs, as it only considers utilization ratio and allows for only two different frequencies for core operation (High and low). Moreover, these models don't offer support for multi-core processors and also ignore the static power of the CPU.

In this thesis the power model for CPU described in [3] is used. This model is the most accurate yet, as it takes into account both the Dynamic Voltage & Frequency Scaling (DVFS) capabilities of modern processors, as well as the number of cores of the CPU. Moreover, the static power of each of the cores is also included in the calculations in order to provide even more accurate results. Finally, the voltage and frequency scaling as well as the utilization percentage are independent for each core.

The CPU power model equation is shown below:

$$\text{CPU_power} = \beta_{freq}(freq_{core0}) * util_{core0} + \beta_{freq}(freq_{core1}) * util_{core1} + \beta_{core0on} * core0_{on} + \beta_{core1on} * core1_{on}$$

DVFS capabilities are represented by 4 different operational frequencies that are derived from the processor's user defined frequency using the following pattern:

- Frequency_Top = Processor's Top Frequency (Defined by User)
- Frequency_High = Frequency_Top – 200 MHz
- Frequency_Medium = Frequency_Top – 400 MHz
- Frequency_Low = Frequency_Top – 700 MHz

Each of the two cores can operate in a different frequency as shown by the $freq_{core0}$ and $freq_{core1}$ parameters of the model. Also each core has its own utilization ratio.

Some minor modifications were done in the model of [3] so that it can be applied to a broad range of CPUs (various top frequencies). Only requirement is that the Processor's Top Frequency, set by the user, needs to be above 800 MHz, which is the case for the CPU frequency of every modern smartphone available. Finally the current power model can be easily expanded to support any number of cores that may be found on mobile processors in the future.

A description of each of the model's coefficients together with its value can be seen in the table below:

Coefficient	**Description**	**Value**
β_{freq}(Frequency_Top)	Coefficient when at Top Frequency	8.41*Frequency_Top*10^{-3}
β_{freq}(Frequency_High)	Coefficient when at High Frequency	6.33*Frequency_Top*10^{-3}
β_{freq}(Frequency_Medium)	Coefficient when at Medium Frequency	4.61*Frequency_Top*10^{-3}
β_{freq}(Frequency_Low)	Coefficient when at Low Frequency	1.95*Frequency_Top*10^{-3}

$Util_{core0}$	Utilization of Core 0	0 - 100%
$Util_{core1}$	Utilization of Core 1	0 - 100%
$\beta_{core0on}$	Static Power of Core 0	Set by User – (Pre. Def.: 133.25 mW)
$\beta_{core1on}$	Static Power of Core 1	Set by User – (Pre. Def.: 150.61 mW)
$core0_{on}$	Core 0 Always On	0 , 1
$core1_{on}$	Core 1 Always On	0 , 1

Table 1: Application Processor Power Model Parameters

4.2 Graphics Processor (GPU)

With the intensive graphic requirements of modern mobile applications, latest mobile games and HD Videos, graphics processing requirements in smartphones are increasing in a fast pace. Previous generation smartphones let the CPU take care of these calculations which end up taking a significant amount of time to complete and consume most of the CPU's resources, resulting in slower response times and poor user interface experience. These reasons led to the insertion of a graphics processor module in modern smartphones, dedicated in handling all the graphic calculations required. This module helps relieve CPU of extra workload and also offers superior graphics capabilities able to support the requirements of modern graphic intensive mobile applications. Previous work on smartphone power modeling found in [1] and [2] doesn't model the GPU at all, possibly because such a module was not available at the time in the smartphone devices they measured. Work done in [3], even though

it presents a more accurate and generic smartphone power model than previous attempts, it still ignores the GPU consumption.

The GPU power model used in this thesis is based on the work done in [5], where the power behavior of a smartphone's GPU is analyzed while running four applications, each with a different GPU utilization ratio. As shown by the results at [5], GPU power consumption accounts for up to 20% of the total system's power consumption. Thus, GPU power consumption needs to be modeled to guarantee sufficient accuracy. As demonstrated in [5], GPU power consumption does not show much deviation across the four applications, though the GPU utilization ratios for all four applications vary significantly. In addition, most recent smartphones do not support DVFS capabilities for the GPU.

Hence, the power model used in this thesis is the same as described in [5]:

$$\text{GPU_active_power} = \beta_{GPU} * GPU_{on}$$

Two different states are used for β_{GPU} in my work.

A description of each of the model's coefficients together with its value can be seen in the table below:

Coefficient	Description	Value
β_{GPU}_High_power	Coefficient when doing Graphic intensive Calculations	Set by User – (Pre. Def.: 660 mW)
β_{GPU}_Low_power	Coefficient when doing Graphic non-intensive Calculations	Set by User – (Pre. Def.: 500 mW)
GPU_{on}	GPU Module ON	0 , 1

Table 2: GPU Power Model Parameters

Moreover, zero consumption is assumption when the GPU in not in use since it can be powered off.

4.3 3G Modem – 4G/LTE Modem

The modem is one of the most important features of mobile phones. It includes the transmitter, receiver, power amplifiers and the baseband processor, which handles all incoming and outgoing cellular packets, whether they include voice, messages or data. The power models proposed in [1] and [2] are very abstract and device specific to be used in this generic approach. Work done in [3] only models the power dissipation of a 3G modem for voice transmission (calling), ignoring its major operation of data transferring. Finally, the models proposed for a 3G Modem in [9] and [10] are accurate but hard to use in a simulator like the one built here due to the big number of parameters that needs to be set by the user.

The power model used in this work is taken from [11] where analytical power models - similar as the one shown in [10] - are constructed for a 4G/LTE Modem, a 3G Modem and the WiFi module. Using these models authors then derive a unique power equation for all of these modules. Setting different parameter values for each module (4G/LTE, 3G, WiFi), they are able to show that this user-friendly power model can accurately describe the behavior of each of these modules. Due to its accuracy and user-friendly characteristics (small number of parameters), this power model proposed in [11] is adopted in my work to describe the behavior of both the 3G Modem and the 4G/LTE Modem.

The power model is the following:

$$\text{Modem_active_power} = \text{Power_per_Mbps} * \text{Throughput} + \text{Base_power}$$

Also a parameter is used to define the power dissipated when in Idle Mode.

A description of each of the model's coefficients together with its value can be seen in the table below:

Coefficient	**Description**	**Value**
Power_per_Mbps	Power consumed for each Mbit Transmitted	Set by User – (Pre. Def. for 3G: 122.12 mW/Mbps) (Pre. Def. for 4G/LTE: 51.97 mw/Mbps)

Throughput	Number of Mbit Transmitted per Second	Set by User – (Pre. Def. for 3G: 0.5 Mbps) (Pre. Def. for 4G/LTE: 5 Mbps)
Base_power	Power consumed when on Transmission Mode	Set by User – (Pre. Def. for 3G: 817.88 mW) (Pre. Def. for 4G/LTE: 1288 mW)
Idle_power	Power consumed when on Idle Mode	Set by User – (Pre. Def. for 3G: 10 mW) (Pre. Def. for 4G/LTE: 14 mW)

Table 3: 3G - 4G/LTE Modem Power Model Parameters

4.4 Wifi

Wifi's operation is quite similar to the modem one. It also contains a transmitter, receiver, and a power amplifier, with an important difference being the fact that the handling of transmitted/received packets is usually done by the phone's CPU. Due to it's functional and power behavior similarities with the modem, -seen in all the power models composed in [9], [10] and [11]-, the same power model used in my simulator for the modems, –described in detail above-, is also used for the wifi module.

Wifi module's power model:

```
Wifi_active_power = Power_per_Mbps * Throughput  + Base_power
```

Also a parameter is used to define the power dissipated when in Idle Mode.

A description of each of the model's coefficients together with its value can be seen in the table below:

Coefficient	Description	Value
Power_per_Mbps	Power consumed for each Mbit Transmitted	Set by User – (Pre. Def.: 137.01 mW/Mbps)
Throughput	Number of Mbit Transmitted per Second	Set by User – (Pre. Def.: 2 Mbps)
Base_power	Power consumed when on Transmission Mode	Set by User – (Pre. Def.: 132.86 mW)
Idle_power	Power consumed when on Idle Mode	Set by User – (Pre. Def.: 20 mW)

Table 4: Wifi Power Model Parameters

4.5 Low Power DDR2 Memory

This low power version of the DDR2 memory used in computers is the internal memory chip found in most of the modern smartphones. It serves as the DRAM memory of the smartphone and is accessed much more frequently than the SD memory card. Also its throughput is much higher, since it is built in the phone. [1] and [3] both ignore the power dissipated by memory while [2] only models the power of an SD memory card.

The selection of the specific memory type and its power model used in this work, were made based on the analysis performed in [6], [7] and [8].

Low Power DDR2 Memory power model:

```
Memory_active_power = Power_per_GBps * Bandwidth
```

Also a parameter is used to define the power dissipated when in Idle Mode.

A description of each of the model's coefficients together with its value can be seen in the table below:

Coefficient	Description	Value
Power_per_GBps	Power consumed for each Giga-Byte Transmitted	Set by User – (Pre. Def.: 45 mW/GBps)
Bandwidth	Number of Giga-Bytes Transmitted per Second	Set by User – (Pre. Def.: 6.5 GBps)
Idle_power	Power consumed when on Idle Mode	Set by User – (Pre. Def.: 27 mW)

Table 5: Low Power DDR2 Memory Power Model Parameters

4.6 Audio Codec

The audio codec module is responsible for any sound playback done by the smartphone. It consists of the codec, power amplifier and the digital signal processor (DSP) responsible for processing all the sound signals from the microphone and to the speakers of the phone. The model adopted for the audio codec in this simulator is shown below and it's in compliance with the power models used in [1], [2] and [3]. In all these papers the audio codec power is modeled with a single value that refers to the power dissipated by the module when it's on.

Audio Codec power model:

$$Audio_Codec_active_power = \beta_{audio} * audio_{on}$$

A description of each of the model's coefficients together with its value can be seen in the table below:

Coefficient	Description	Value
β_{audio}	Power consumed when module in ON	Set by User – (Pre. Def.: 390 mW)
$audio_{on}$	Audio Codec is ON	0 , 1

Table 6: Audio Codec Power Model Parameters

4.7 GPS

The GPS module is an essential part for every modern smartphone. Navigation, security, social and numerous other mobile applications make use of the Geo-location features it offers. Work in [2] doesn't model the power dissipation of the GPS module at all, probably because the device for which it tries to create power models doesn't have a GPS chip integrated. The authors of [3] model the power of the GPS module with a single value that refers to the power dissipated by the module when it's on, while work done in [1] uses two different power values to represent the active and sleep mode accordingly. This last approach seen in [1] is also used in this proposed simulator.

GPS power model:

$$GPS_active_power = \beta_{GPS} * GPS_{on}$$

A description of each of the model's coefficients together with its value can be seen in the table below:

Coefficient	Description	Value
β_{GPS}_Active	Power consumed when in Active mode	Set by User – (Pre. Def.: 450 mW)
β_{GPS}_Sleep	Power consumed when in Sleep mode	Set by User – (Pre. Def.: 150 mW)
GPS_{on}	GPS is ON	0 , 1

Table 7: GPS Power Model Parameters

4.8 LCD Screen

The screen is a vital module for every phone. LCD screens in particular offer a wide variety of colors and sharp resolution, features that make them suitable candidates for every smartphone.

Both [1] and [2] use linear equations to describe LCD screen's power, based on brightness level and step power values. But as it was proved in [3], advance LCD screens have non-linear power consumption characteristics that cannot be accurately modeled by

using a single simple linear equation. Their solution includes five different brightness level areas with a power coefficient corresponding to each one of them. Also a power value for the screen being on is added. This model, being the most accurate power model for advance LCD screens yet, is also used in this work with the addition of one extra brightness area to further increase accuracy.

LCD Screen power model:

$$LCD_Screen_active_power = \beta_{br}(brightness) * brightness * LCD_{on} + \beta_LCD_{on} * LCD_{on}$$

A description of each of the model's coefficients together with its value can be seen in the table below:

Coefficient	Description	Value
Brightness	Brightness level used	0 - 255
$\beta_{br}1\sim136$	Power Coefficient when brightness level $1 \leq br < 136$	0
$\beta_{br}136\sim166$	Power Coefficient when brightness level $136 \leq br < 166$	0.59
$\beta_{br}167\sim186$	Power Coefficient when brightness level $167 \leq br < 186$	1.03

β_{br}187~196	Power Coefficient when brightness level 187≤br<196	1.79
β_{br}197~224	Power Coefficient when brightness level 197≤br<225	2.69
β_{br}225~255	Power Coefficient when brightness level 225≤br≤255	3.23
β_LCD_{on}	Power Coefficient when Screen is ON	Set by User – (Pre. Def.: 351.07 mW)
LCD_{on}	LCD Screen is ON	0 , 1

Table 8: LCD Screen Power Model Parameters

Also when the LCD Screen is off, zero power dissipation is assumed.

Chapter 5: Application Power Profiles

Millions of applications are available for smartphones these days and the number is only expected to increase in the future. Each of these applications utilizes a number of hardware modules available in the phone in order to deliver its content. As part of the simulator presented in this work, ten different applications were created, each with its own power profile. These applications were selected as they are some of the most common ones to be executed regularly in the majority of smartphones. They also effectively demonstrate the use of different hardware modules and the transition between various power states during their runtime.

These ten applications together with their unique characteristics, power states and hardware modules usage are described in detail below:

5.1 Standby Application

This application demonstrates the power usage when the phone is on but not operating. Its power profile consists of only one state, which adds the idle or static power of all of its components (hardware modules) that are left on.

The operating modes of each of the smartphone's hardware modules for all the power states of this specific application are seen below:

H/w Modules	Power State 1
CPU	Static power of each of the Processor's Cores that is On
GPU	Off
3G Modem or 4G/LTE Modem	Idle Power
Wifi	Off
LPDDR2 Memory	Idle Power
Audio Codec	Off
GPS	Off
LCD Screen	Off

Table 9: Standby Application's Hw Modules Operation Mode per Power State

Display of the power profile of the Standby application after running for 5 minutes:

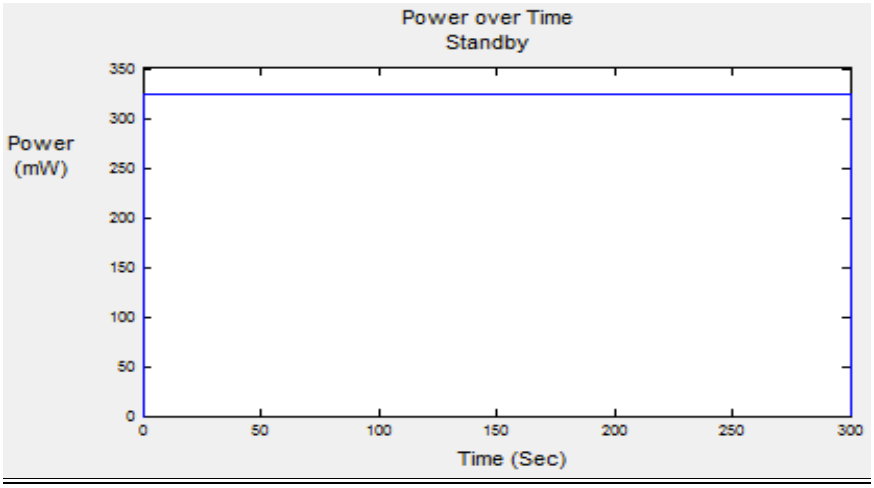

Figure 2: Power Profile of Standby Application

5.2 Interaction Application

This application demonstrates the power usage when the user is interacting with the phone, browsing through applications, checking stored information or changing settings. Its power profile consists of only one state.

The operating modes of each of the smartphone's hardware modules for all the power states of this specific application are seen below:

H/w Modules	Power State 1
CPU	State 1 frequency and utilization ratio + Static power of each of the Processor's Cores that is On
GPU	Power Low
3G Modem or 4G/LTE Modem	Idle Power
Wifi	Off
LPDDR2 Memory	Active Power
Audio Codec	Off
GPS	Off
LCD Screen	On

Table 10: Interaction Application's Hw Modules Operation Modes per Power State

Display of the power profile of the Interaction application after running for 5 minutes:

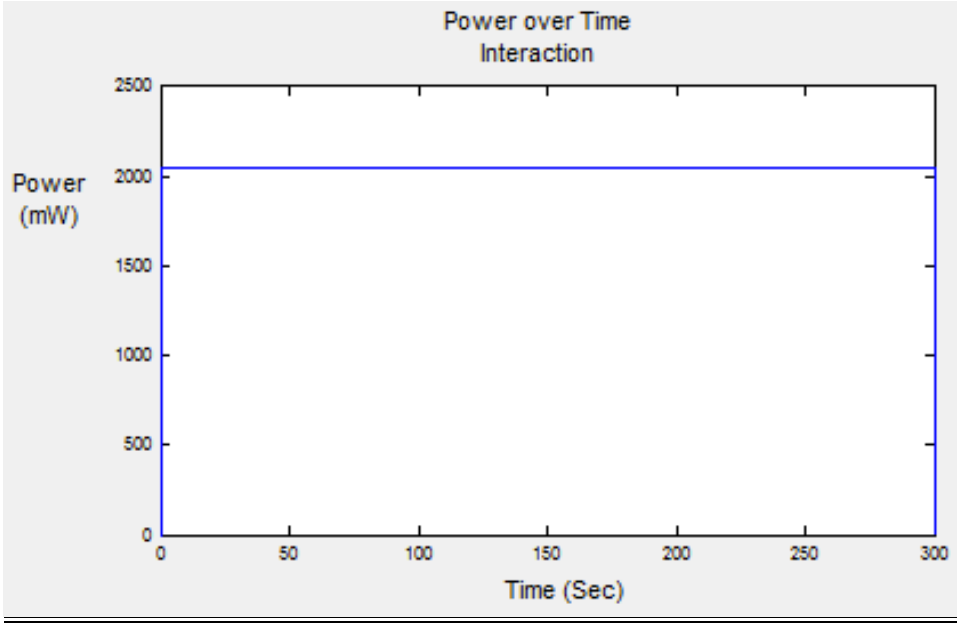

Figure 3: Powel Profile of Interaction Application

5.3 Music Playing Application

This application demonstrates the power being dissipated over time when the user is listening to music selected from a list of tracks stored in the phone's memory. This application's power profile consists of the following three power states:

- 1st state: Selection of music tracks to be played (Lasts for 60 seconds)
- 2nd state: Music Playing
- 3rd State: Power overhead (Lasts for 5 seconds)

The operating modes of each of the smartphone's hardware modules for all the power states of this specific application are seen below:

H/w Modules	Power State 1	Power State 2	Power State 3
CPU	State 1 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 2 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 3 frequency and utilization ratio + Static power of each of the Processor's Cores that is On
GPU	Power Low	Off	Power Low
3G Modem or 4G/LTE Modem	Idle Power	Idle Power	Idle Power
Wifi	Off	Off	Off
LPDDR2 Memory	Active Power	Active Power	Active Power
Audio Codec	Off	ON	Off
GPS	Off	Off	Off
LCD Screen	On	Off	On

Table 11: Music Playing Application's Hw Modules Operation Mode per Power State

Display of the power profile of the Music Playing application after running for 5 minutes:

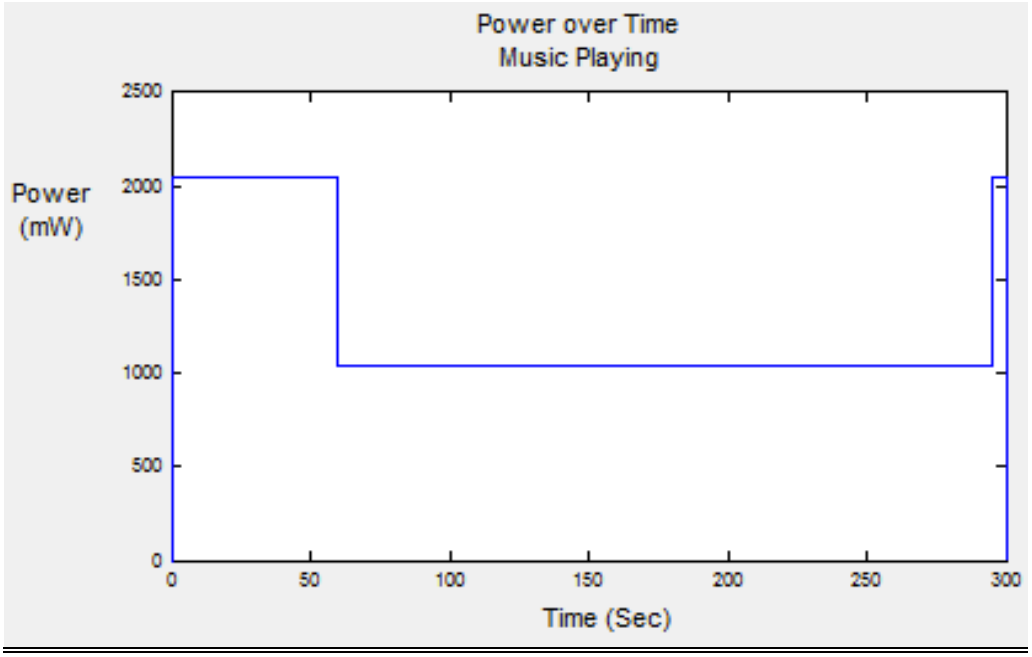

Figure 4: Power Profile of Music Playing Application

5.4 Video Streaming (Wifi) Application

This application demonstrates the power being dissipated over time when the user is streaming a video online and watches it on his/her smartphone. Uncompressed High Definition video requires really high throughput rates which would be impossible to achieve using today's modem and wifi technology. This is the reason many compression formats were created the last 5 years to handle mobile HD video. The format selected in the proposed simulator is the **H.264,** which is the most popular compression format being used today for High Definition Video. This format greatly reduces the demand for throughput to **1.5 Mbps**,

which is an acceptable rate for today's Wifi and Modem technologies.

The video streaming power profile consists of the following four power states:

- 1st state: Launch of the video playback –Beginning of Streaming
- (20 seconds duration)
- 2nd state: Video playback – streaming
- 3rd state: Video playback – streaming is done.
- 4th state: Power overhead (10 seconds duration).

The video's size is found by multiplying the video runtime selected with the rate required for HD Video by the H.264 format (1.5 Mbps).

The duration of the 2nd state is calculated by dividing the video size with the predefined throughput of the Wifi module. Then the duration of the 3rd state is simply the remaining time after subtracting the 2nd state duration from the video playback time selected.

The operating modes of each of the smartphone's hardware modules for all the power states of this specific application are seen below:

H/w Modules	**Power State 1**	**Power State 2**	**Power State 3**	**Power State 4**
CPU	State 1 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 2 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 3 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 4 frequency and utilization ratio + Static power of each of the Processor's Cores that is On

GPU	Power High	Power High	Power High	Power Low
3G Modem or 4G/LTE Modem	Idle Power	Idle Power	Idle Power	Idle Power
Wifi	Active Power at predefined throughput	Active Power at predefined throughput	Idle Power	Idle Power
LPDDR2 Memory	Active Power	Active Power	Idle Power	Active Power
Audio Codec	On	On	On	Off
GPS	Off	Off	Off	Off
LCD Screen	On	On	On	On

Table 12: Video Streaming Application's Hw Modules Operation Mode per Power State

Display of the power profile of the Video Streaming application after running for 5 minutes:

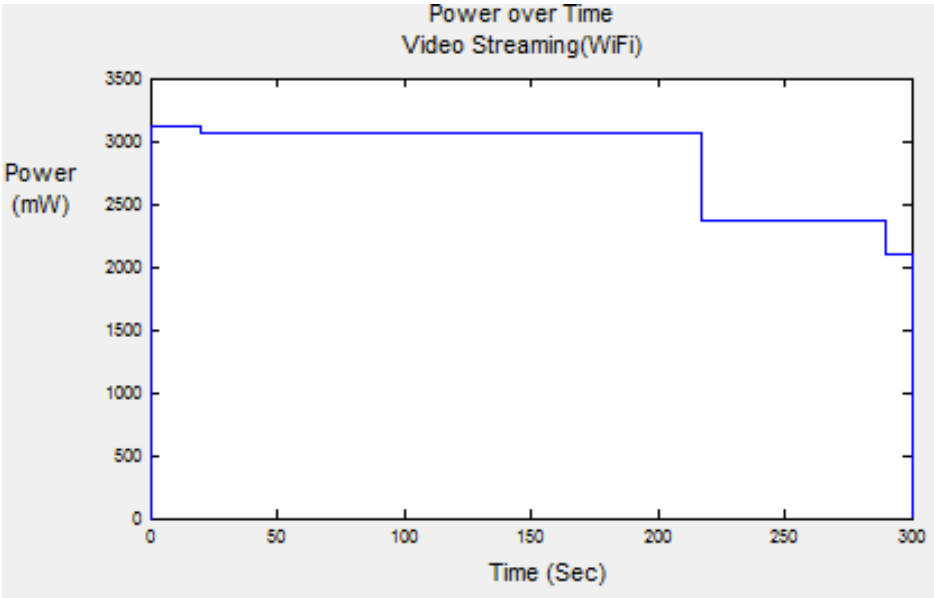

Figure 5: Power Profile of Video Streaming Application

5.5 Calling Application

This application demonstrates the power being dissipated over time when the user is making a call. Voice packets require a transmit/receive rate of 12,65 Kbps in order for the user to experience a high quality call. This rate is much lower that the available max throughput of all 3G or 4G/LTE Modems. So when the modem is active during the call, it automatically readjusts its throughput to 12,65 Kbps to reduce its power dissipation while ensuring a high quality call. The algorithm developed here for the calling application follows the same trend.

The calling power profile consists of the following three power states:

- 1st state: Interaction–Search for contact to call (10 seconds duration)
- 2nd state: Calling (specified duration)
- 3rd state: Power overhead (5 seconds duration)

The operating modes of each of the smartphone's hardware modules for all the power states of this specific application are seen below:

H/w Modules	**Power State 1**	**Power State 2**	**Power State 3**
CPU	State 1 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 2 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 3 frequency and utilization ratio + Static power of each of the Processor's Cores that is On
GPU	Power Low	Off	Off
3G Modem or 4G/LTE Modem	Idle Power	Active Power at calling specific throughput	Active Power at calling specific throughput
Wifi	Off	Off	Off
LPDDR2 Memory	Active Power	Idle Power	Active Power
Audio Codec	Off	Off	Off
GPS	Off	Off	Off

LCD Screen	On	Off	On

Table 13: Calling Application's Hw Modules Operation Mode per Power State

Display of the power profile of the Calling application after running for 5 minutes:

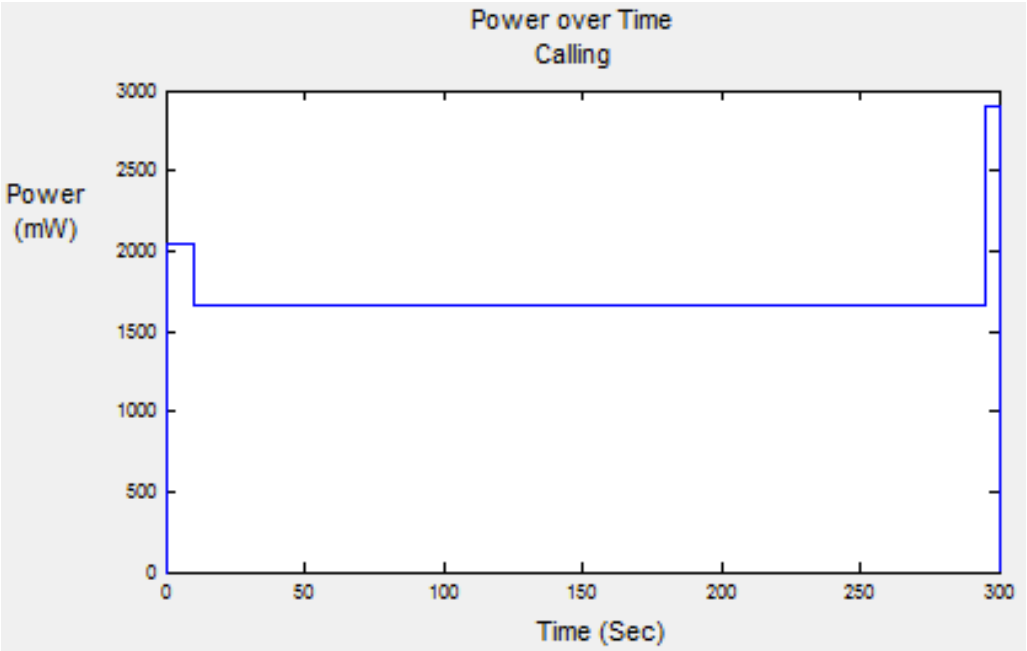

Figure 6: Power Profile of Calling Application

5.6 Messaging Application

This application demonstrates the power being dissipated over time when the user is sending text messages. The size of a single text message,-based on Wikipedia-, is 140 Bytes and together with headers that are attached to it before it is sent is reaches almost 200 Bytes. Assuming that the modem should support the transmission of one message per second, a throughput of 1.6 Kbps is required. Modern modems automatically readjust their throughput

to the one mentioned above while sending messages in order to reduce the dissipated power. The algorithm developed here for the calling application follows the same trend.

The power profile of each text message consists of the following three power states:

- 1st state: Typing text message – Selecting recipient (55 seconds duration)
- 2nd state: Sending text message (2 seconds duration)
- 3rd state: Power Overhead (3 seconds duration)

To calculate the amount of simulated text messages sent, the proposed simulator divides the total selected messaging time with the overall time of each text message (60 seconds). The power profile of a single message is then repeated the number of times found above. The messaging application ends after the last message's power profile.

The operating modes of each of the smartphone's hardware modules for all the power states of this specific application are seen below:

H/w Modules	**Power State 1**	**Power State 2**	**Power State 3**
CPU	State 1 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 2 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 3 frequency and utilization ratio + Static power of each of the Processor's Cores that is On
GPU	Power Low	Off	Off

3G Modem or 4G/LTE Modem	Idle Power	Active Power at messaging specific throughput	Active Power at messaging specific throughput
Wifi	Off	Off	Off
LPDDR2 Memory	Active Power	Idle Power	Active Power
Audio Codec	Off	Off	Off
GPS	Off	Off	Off
LCD Screen	On	On	On

Table 14: Messaging Application's Hw Modules Operation Mode per Power State

Display of the power profile of the Messaging application after running for 5 minutes:

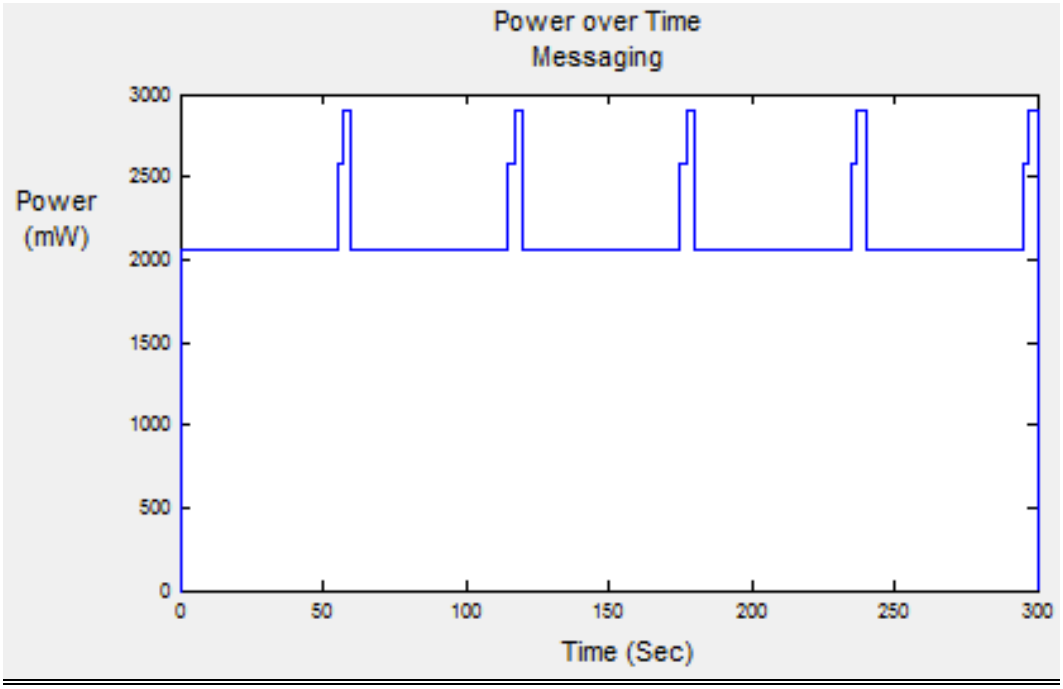

Figure 7: Power Profile of Messaging Application

5.7 Browsing (Wifi) Application

This application demonstrates the power being dissipated over time when the user is browsing through the web using the Wifi module to download data. The average size of a modern web page is 1.5 Mbytes (source: webperformancetoday.com), while a new page is approximately downloaded every 40 seconds during browsing.

The power profile of browsing through each web page consists of the following three power states:

- 1st state: Page download– Navigation through page
- 2nd state: Power overhead of download – Navigation through page (5 seconds duration)
- 3rd state: Download done - Navigation through page

To calculate the number of web pages browsed through during the selected runtime, the proposed simulator divides the total selected browsing time with the overall time of a single page browsing (40 seconds). The power profile of a single web page browsing is then repeated the number of times found above. After the last page to be browsed a power overhead state for the entire application is added:

- 4th state: Power overhead of entire application (10 seconds duration)

The browsing application ends after the overall power overhead power state is applied.

The time needed to download a page (1st state duration) is found by dividing the web page's size (1.5MBytes) with the Wifi module's predefined throughput, -divided by 8 to turn into

MBps from mbps -. The duration of the 3rd state is simply the result of the subtraction of (download time + time of power overhead of download) from the overall browsing time that was selected.

The operating modes of each of the smartphone's hardware modules for all the power states of this specific application are seen below:

H/w Modules	**Power State 1**	**Power State 2**	**Power State 3**	**Power State 4**
CPU	State 1 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 2 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 3 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 4 frequency and utilization ratio + Static power of each of the Processor's Cores that is On
GPU	Off	Off	Off	Off
3G Modem or 4G/LTE Modem	Idle Power	Idle Power	Idle Power	Idle Power
Wifi	Active Power at predefined throughput	Active Power at predefined throughput	Idle Power	Idle Power
LPDDR2 Memory	Active Power	Idle Power	Idle Power	Active Power
Audio Codec	Off	Off	Off	Off
GPS	Off	Off	Off	Off

LCD Screen	On	On	On	On

Table 15: Browsing (Wifi) Application's Hw Modules Operation Mode per Power State

Display of the power profile of the Browsing (Wifi) application after running for 5 minutes:

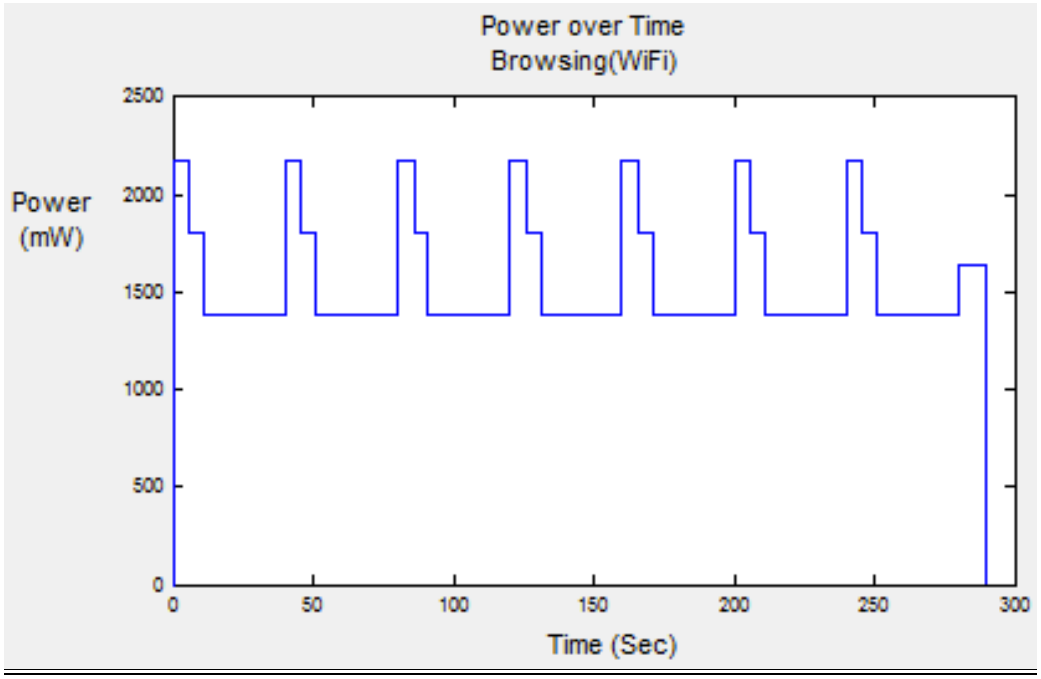

Figure 8: Power Profile of Browsing (Wifi) Application

5.8 Browsing (Cellular) Application

This application demonstrates the power being dissipated over time when the user is browsing through the web using the Cellular module (3G or 4G/LTE Modem) to download data. The average size of a modern web page is 1.5 Mbytes (source:

webperformancetoday.com while a new page is approximately downloaded every 40 seconds during browsing.

Similar to the browsing (Wifi) application, the power profile of the browsing through each web page consists of the following three power states:

- 1st state: Page download– Navigation through page
- 2nd state: Power overhead of download – Navigation through page
- 3rd state: Download done - Navigation through page

To calculate the number of web pages browsed through, the proposed simulator divides the total selected browsing time with the overall time of a single page browsing (40 seconds). The power profile of a single web page browsing is then repeated the number of times found above. After the last page to be browsed a power overhead state for the entire application is added:

- 4th state: Power overhead of entire application (10 seconds duration)

The browsing application ends after the overall power overhead power state is applied.

Same as for the browsing (Wifi) application, the time needed to download a page (1st state duration) is found by dividing the web page's size (1.5MBytes) with the Modem's (3G or 4G/LTE) predefined throughput, -divided by 8 to turn into MBps from mbps -. The duration of the 3rd state is simply the result of the subtraction of (download time + time of power overhead of download) from the overall browsing time that was selected.

The operating modes of each of the smartphone's hardware modules for all the power states of this specific application are seen below:

H/w Modules	Power State 1	Power State 2	Power State 3	Power State 4
CPU	State 1 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 2 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 3 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 4 frequency and utilization ratio + Static power of each of the Processor's Cores that is On
GPU	Off	Off	Off	Off
3G Modem or 4G/LTE Modem	Active Power at predefined throughput	Active Power at predefined throughput	Idle Power	Idle Power
Wifi	Off	Off	Off	Off
LPDDR2 Memory	Active Power	Idle Power	Idle Power	Active Power
Audio Codec	Off	Off	Off	Off
GPS	Off	Off	Off	Off
LCD Screen	On	On	On	On

Table 16: Browsing (Cellular) Application's Hw Modules Operation Mode per Power State

Display of the power profile of the Browsing (Cellular) application after running for 5 minutes:

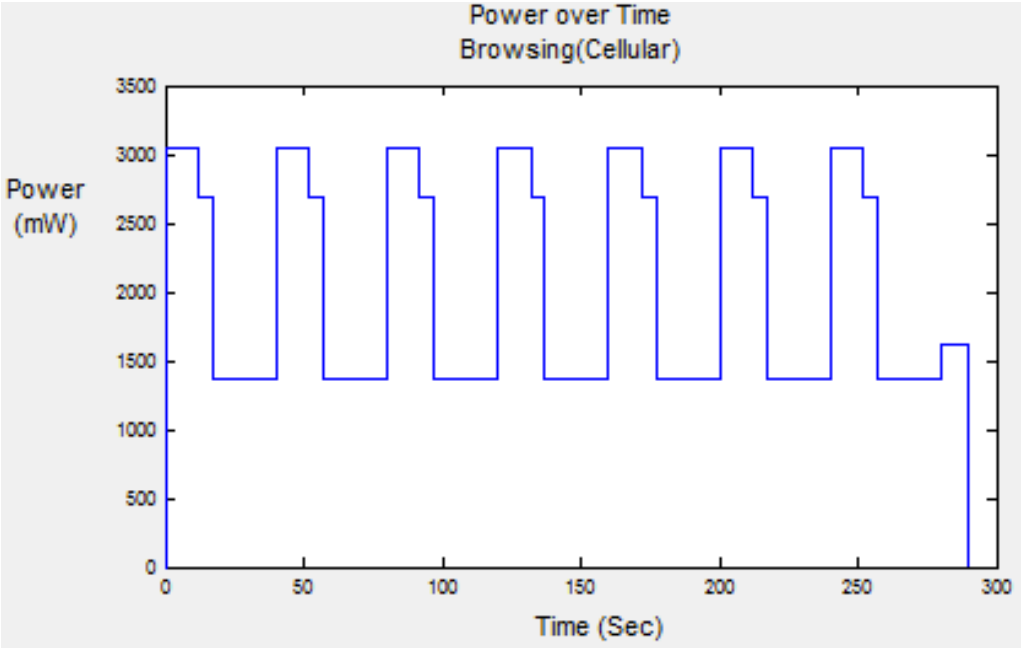

Figure 9: Power Profile of Browsing (Cellular) Application

5.9 Navigation (Cellular) Application

This application demonstrates the power being dissipated over time when the user is utilizing navigation software, using the smartphone's Cellular module (3G or 4G/LTE Modem) to download data. Every navigation application has to go through an initial satellite tracking and location identification phase which also includes the download of a substantial amount of the surrounding area's map data (500 KBytes).

The power profile of this initialization phase consists of the following power states:

- 1st state: GPS tracking – Downloading of initial map data.
- 2nd state: GPS tracking – Downloading is done.

The initialization phase lasts for 20 seconds and the time of the 1st state is found by dividing the amount of initial data to be downloaded (500KBytes) with the Cellular's module (Modem 3G or 4G/LTE) predefined throughput -divided by 8 to turn into MBps from mbps -. The duration of the 2nd state is simply the result of the subtraction of the calculated download time above from the total initialization time (20 seconds).

After this initial phase is done the actual navigation period starts. During navigation it was assumed that GPS tracking needs to be enabled and 100Kbytes of data need to be downloaded once every 5 seconds.

The power profile of each navigation cycle consists of the following power states:

- 3rd state: Navigation – GPS tracking – Downloading of additional map data.
- 4th state: Navigation – No GPS tracking or downloading of map data.

The duration of the 3rd state is found by dividing the amount of additional data to be downloaded (100KBytes) with the Cellular's module (Modem 3G or 4G/LTE) predefined throughput, -divided by 8 to turn into MBps from mbps -. The duration of the 4th state is simply the result of the subtraction of the calculated 3rd state time above from the total navigation cycle time (5 seconds).

The number of navigation cycles is found by dividing the (total navigation time selected – initialization time) with the time of a single navigation cycle (5 seconds).

After the last navigation cycle a power overhead state for the entire application is added:

- 4th state: Power overhead of entire application (10 seconds duration)

The navigation (cellular) application ends after the overall power overhead power state is applied.

The operating modes of each of the smartphone's hardware modules for all the power states of this specific application are seen below:

H/w Modules	**Power State 1**	**Power State 2**	**Power State 3**	**Power State 4**	**Power State5**
CPU	State 1 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 2 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 3 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 4 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 5 frequency and utilization ratio + Static power of each of the Processor's Cores that is On
GPU	Power Low	Power Low	Power Low	Power Low	Off
3G Modem or 4G/LTE Modem	Active Power at predefined throughput	Idle Power	Active Power at predefined throughput	Idle Power	Idle Power
Wifi	Off	Off	Off	Off	Off
LPDDR2 Memory	Active Power	Idle Power	Active Power	Idle Power	Active Power

Audio Codec	Off	Off	Off	Off	Off
GPS	Active Power	Active Power	Active Power	Sleep Power	Active Power
LCD Screen	On	On	On	On	On

Table 17: Navigation (Cellular) Application's Hw Modules Operation Mode per Power State

Display of the power profile of the Navigation (Cellular) application after running for 2 minutes:

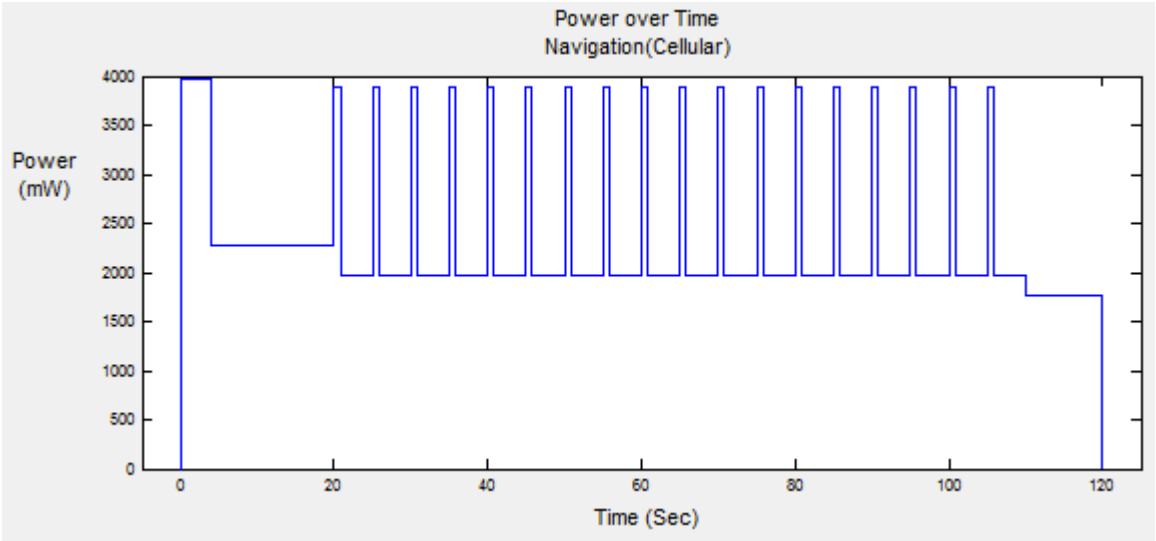

Figure 10: Power Profile of Navigation (Cellular) Application

5.10 Gaming (Offline) Application

This application demonstrates the power being dissipated over time when the user is paying an offline game on his/her smartphone.

This application's power profile consists of the following three power states:

- 1st state: Launch of game– Loading of features (Lasts for 40 seconds)
- 2nd state: Game Playing
- 3rd State: Power overhead (Lasts for 5 seconds)

The operating modes of each of the smartphone's hardware modules for all the power states of this specific application are seen below:

H/w Modules	Power State 1	Power State 2	Power State 3
CPU	State 1 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 2 frequency and utilization ratio + Static power of each of the Processor's Cores that is On	State 3 frequency and utilization ratio + Static power of each of the Processor's Cores that is On
GPU	Power Low	Power High	Power Low
3G Modem or 4G/LTE Modem	Idle Power	Idle Power	Idle Power
Wifi	Off	Off	Off

LPDDR2 Memory	Active Power	Active Power	Active Power
Audio Codec	Off	On	Off
GPS	Off	Off	Off
LCD Screen	On	On	On

Table 18: Gaming (Offline) Application's Hw Modules Operation Mode per Power State

Display of the power profile of the Gaming (Offline) application after running for 5 minutes:

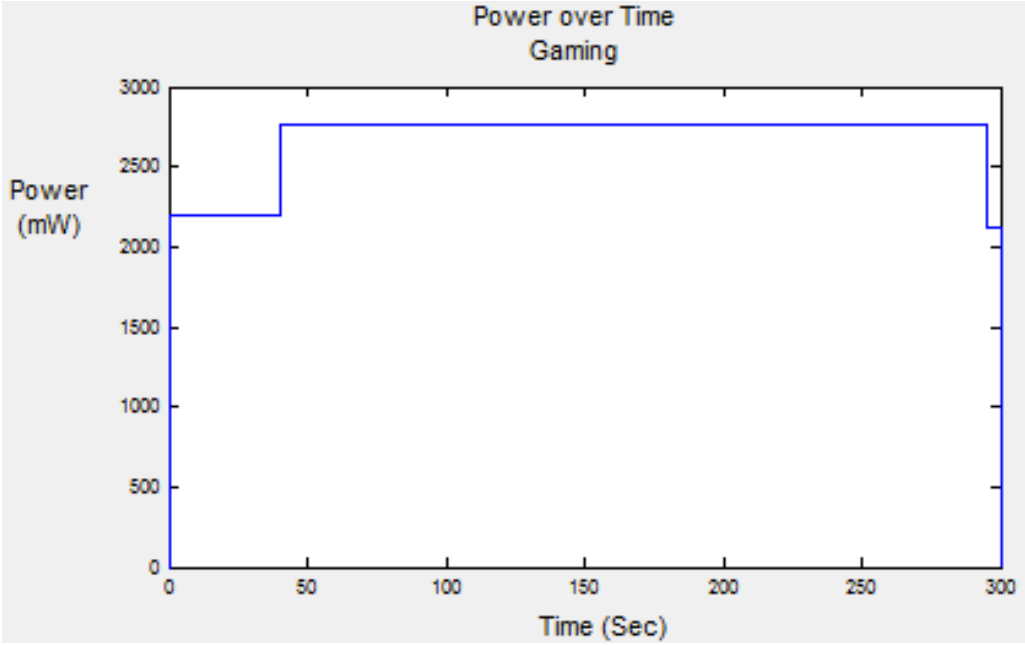

Figure 11: Power Profile of Gaming (Offline) Application

Notes:

- The 3rd available hardware configuration, -presented in chapter 7-, was selected and used to run the power simulations of the above applications. The GPU module included above is only present in this configuration. For the other two configurations an off state can be assumed for the GPU for every application.
- The processor's frequency and utilization rate for the power states of each application are different between the three available hardware configurations displayed at chapter 7. These values were selected based on previous papers and real time measurements using PowerTutor in various smartphone devices.
- The 4G/LTE Modem bandwidth was reduced to 2 Mbps (from 5Mpbs that is its predefined value), when a simulation of the Browsing (Cellular) and the Navigation (Cellular) applications was ran, so that the transitions between different power states can be seen more clearly.

Chapter 6: Parallel Execution Algorithm

6.1 Motivation for Algorithm Development

As mentioned earlier, a big contribution of the current work is the development of an algorithm that allows for accurate estimation of the power dissipated over time when two applications are executed in parallel. This important feature is not present in any of the previous smartphone power estimation attempts or tools developed shown in chapter 2. Parallel execution power can be calculated using real time power measuring tool like PowerTutor by simply calculating the power of each of the smartphone's hardware modules when two applications are running simultaneously. But real time tools have many disadvantages and inaccuracies as described in the Previous Work chapter.

6.2 Parallel Execution Capabilities of Modeled Applications

The application models that are included in this work differ from each other in their parallel execution capabilities, with some of them only able to be executed sequentially, while others can be executed on parallel with another application.

These capabilities for all available applications are demonstrated on the table below:

Reference Application	**Applications that can run in parallel with the reference application**
Standby	-
Interaction	Calling Music Playing
Calling	Interaction Messaging Browsing (Wifi or Cellular) Navigation (Cellular)
Messaging	Calling Music Playing
Music Playing	Interaction Messaging Browsing (Wifi or Cellular) Navigation (Cellular)
Video Streaming	-
Browsing (Wifi or Cellular)	Calling Music Playing
Navigation (Cellular)	Calling Music Playing
Gaming (Offline)	-

Table 14: Parallel Execution Capabilities of Modeled Applications

During each time unit (second) only two applications are allowed to be executed in parallel, same as when these applications are running in real smartphones.

6.3 Description of Algorithmic Steps

Below is a description of each of the steps of the parallel execution algorithm proposed in this work:

1. When simulation starts the current algorithm analysis the entire time period of the simulation to identify the areas where parallel execution may exist.
2. For each one of the identified parallel execution time windows the algorithm then creates a time matrix and a power states matrix of time areas for each of the two parallel applications according to the power states that are present in this timing window for each of them.
3. The two time matrixes are then connected into one matrix, containing all the time areas that need to be examined.
4. The power states matrixes of the parallel applications are then increased as needed to include the power state of each of the two applications under each of the time windows defined in the unified time matrix created in the previous step.
5. For each of the unified time matrix's time areas the power state of each of the applications is read and the corresponding power for each hardware module - except the CPU - for this application in the specific time area is reported. This power is then compared with the power of the same module reported for the second application under the same time area. The algorithm picks the bigger of the two values, which is added to the final parallel power. The higher value is assumed to be the same as the parallel value as most of the modules have only one active power value.
6. To calculate the CPU's parallel power the frequency and utilization ratio of

each core for both applications at each time area is reported. The product of frequency* utilization of each time area is then computed for each application it order to find the number of instructions that are executed per second for each application. The two numbers are added and the total value is divided with the higher of the two frequencies. This gives the utilization ratio that the processor must execute at – when operating at the higher of the two frequencies -, in order to support the same amount of instructions executed per second as required by the two applications together in each time area. These values are finally applied to the CPU power model and the power reported is added to the parallel power for the specific time area.

7. Last but not least the algorithm reports the calculated parallel power over time of the entire parallel execution time window.

The functionality of the steps of the parallel execution algorithm shown above is further demonstrated in the second example presented in chapter 8 - "Demonstration of Power Simulator" -.

Notes:

- The CPU frequency – utilization recalculation is really important since it avoids overestimating power, which would be the result if the two CPU powers were just added. Also it's closer to how the processor operation is recalibrated in real time operating systems.
- The steps 2-7 are repeated for each parallel execution time window found after the initial simulation time analysis performed in step 1.

Chapter 7: Power Simulator's User Interface

The smartphone simulator developed here offers the following features:

- Universal hardware modules' power models with small number of parameters so that they can be user friendly yet still accurate.

- Display of max power dissipated for each hardware module based on its selected parameters.

- Ten different popular applications' power profiles with clear distinction between different power states within the application.

- Parallel application execution capabilities with display of the total (parallel) power over time and power over time of each of the parallel applications.

- Total energy consumption for the simulated time window and energy consumption of each of the applications under test.

7.1 Available Configurations

The initial screen of the graphical user interface (GUI) of the developed simulator offers a selection between three predefined hardware configurations for the potential smartphone and a custom configuration option. An area is also included where a list of all the hardware modules contained in the smartphone configuration selected can be seen (blue area on the right of the initial window).

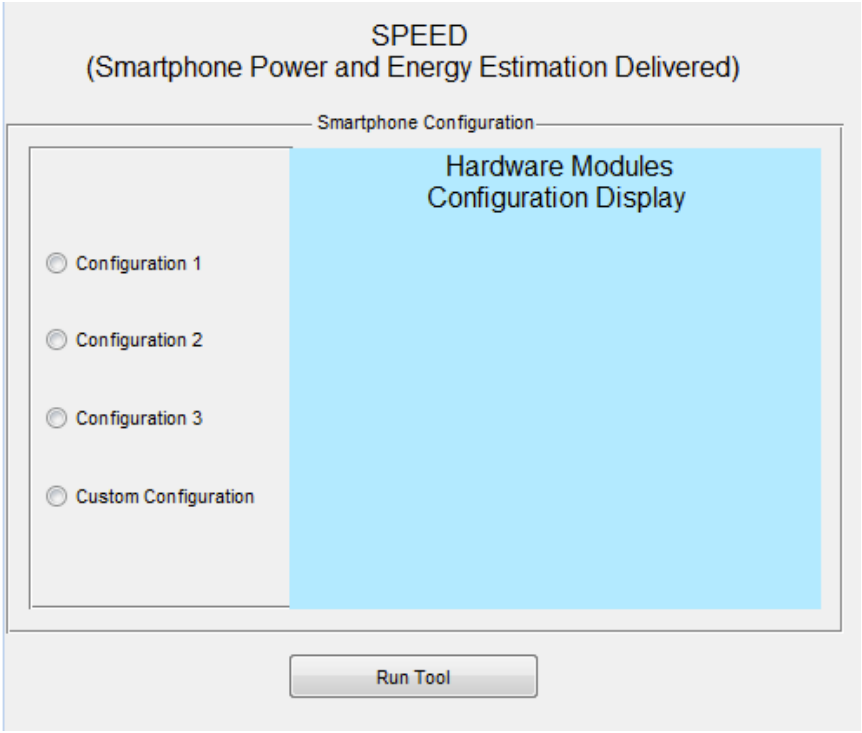

Figure 12: Demonstration of Hardware Configuration Selection Window

The configurations available are the following:

- Configuration 1

Applications Processor (Single Core)
3G Modem
WiFi
LCD Screen
LPDRAM
GPS
Audio Codec

Figure 13: Modules of 1st Hardware Configuration

- Configuration 2

Applications Processor (Dual Core)
4G/LTE Modem
WiFi
LCD Screen
LPDRAM
GPS
Audio Codec

Figure 14:Modules of 2nd Hardware Configuration

- Configuration 3

Applications Processor (Dual Core)
GPU
4G/LTE Modem
WiFi
LCD Screen
LPDRAM
GPS
Audio Codec

Figure 15: Modules of 3rd Hardware Configuration

When one of the above mobile hardware configurations is selected, tables containing the user defined parameters for each of the modules available under the current configuration appear on the GUI. The default value for each of these parameters, – displayed in chapter 4 "Power Models of Smartphone's Hardware Modules"-, automatically appear in each value box, but the user is available to change them according to the hardware modules' specifications that he wishes to test.

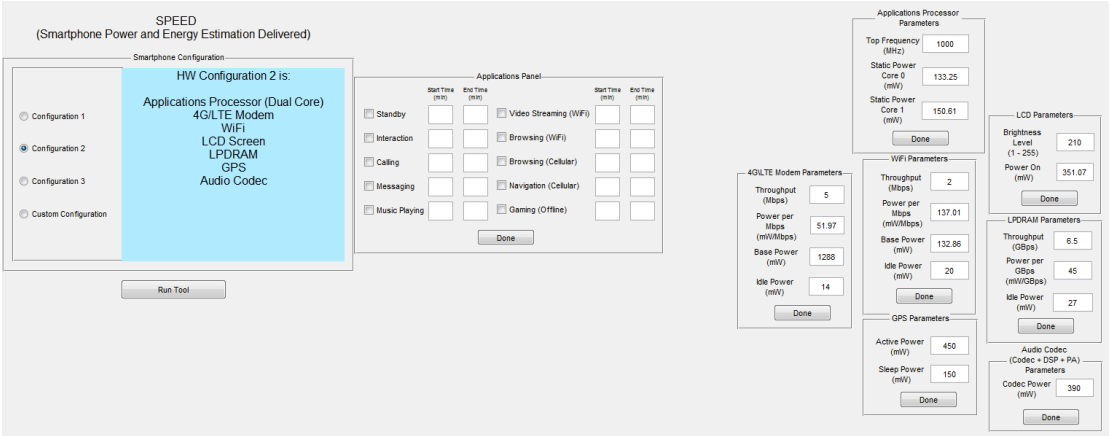

Figure 16:Demonstration of GUI Window after Hardware Configuration Selection

The simulator also offers a 'Custom Configuration' option, where the user is able to manually select the hardware modules that he wishes to include in the hardware configuration of his featured smartphone, from a menu of all available hardware modules that appears on the initial GUI window. Parameter setting then becomes available for each module that is selected.

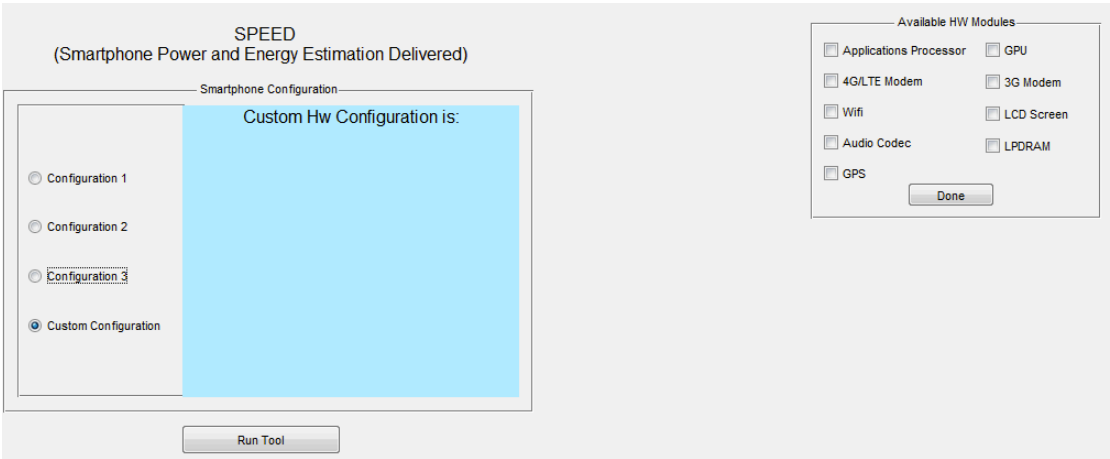

Figure 17: Demonstration of GUI Window after the Custom Configuration Option is Selected

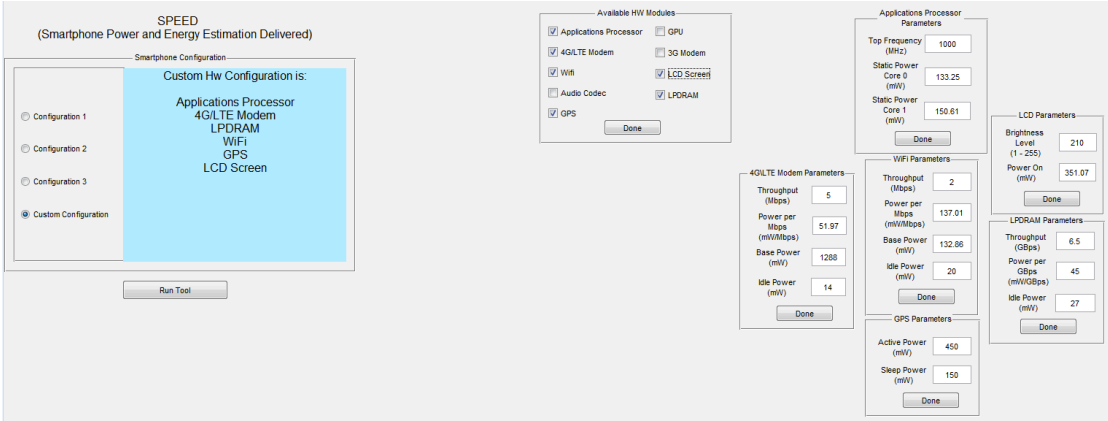

Figure 18: Demonstration of Selection of Various Hardware Modules Under the Custom Configuration Option

This configuration can only be used to test the maximum active power of each of the hardware modules selected based on the parameter values set. The rest of the tool capabilities demonstrated below only apply for Configurations 1, 2 and 3.

7.2 Application Selection

When one of the 3 available predefined hardware configurations is selected, a window containing all the available mobile applications appears. Each of these applications, together with its power profile, is described in detail in chapter 5 “Application Power Profiles”. The user then needs to select the applications that he/she wants to simulate running. For each application selected a start and an end time must be declared. These values are in minutes and refer to the time window the application will be executed at during the simulation phase. The user can also select the start and end time for different applications in a way that results in parallel execution for a period of the applications’ execution time or even their entire runtime. For example, if the music playing application is selected to start at the 5th minute and end at

the 15th minute, while the browsing (Wifi) application is selected to start at the 10th minute and finish at the 20th minute, that will lead to the parallel execution of the two applications mentioned above from the 10th till the 15th minute.

Figure 19: Demonstration of Applications Selection and Runtime Period Setup

If one of the applications that are only allowed to run sequentially, –found in the table seen in chapter 6 “Parallel Execution Algorithm” -, is selected to run in parallel with another application for a period of time or for its entire execution time, the tool notifies the user for the error and the simulation halts for the user to redefine the time periods of the applications.

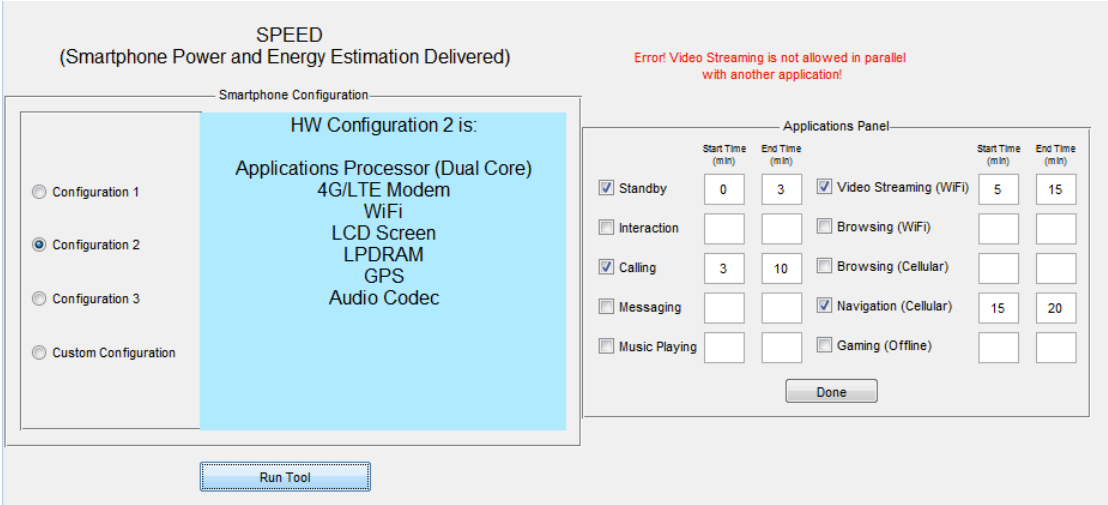

Figure 20: Demonstration of Error Message Display for Non-Parallel Allowed Application Execution Attempt

Also if three applications' time windows are selected in a way that end up in time periods where all three are to be executed in parallel, the simulator again displays an error message notifying the user while halting the simulation.

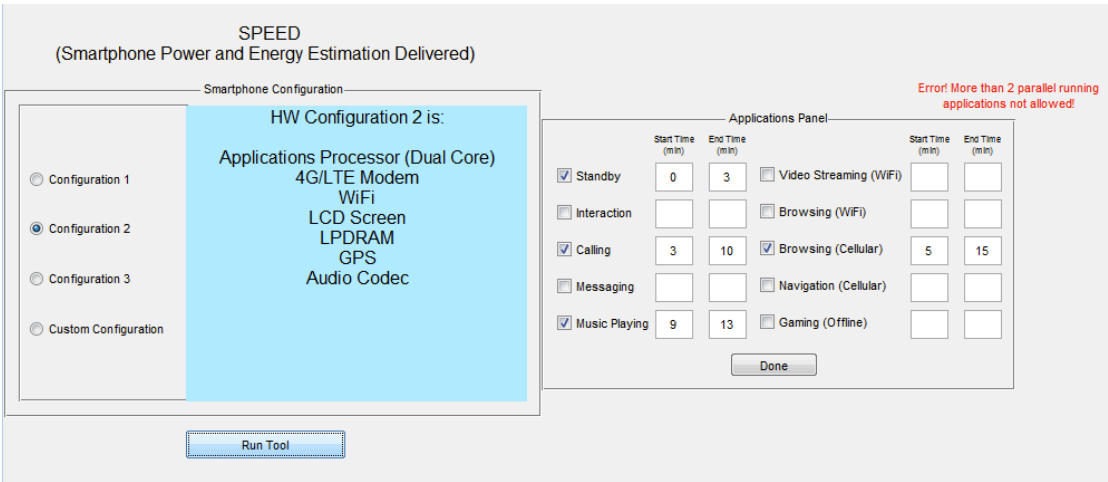

Figure 21: Demonstration of Error Message Display for 3 Parallel Applications Execution Attempt

7.3 Simulation Phase

After selecting a hardware configuration, setting its modules' user defined parameter and selecting applications to be executed, the simulation is ready to start. When the user hits the run button a time line for the entire simulation is created based on the start and end times of the applications selected. The time line is divided into sequential execution time areas, – where the power of the single application executed at that time window is calculated based on the application's power profile -, and parallel execution time areas (if any exists), – where the power is calculated for each time widow with a different power state, based on the parallel execution algorithm described in chapter 6 "Parallel Execution Algorithm".

7.4 Reported Results

After the simulation is done three result windows appear in the GUI of the simulator. The first one, - located on the bottom left side of the GUI –, displays the maximum active power of each of the hardware modules of the selected configuration based on the parameter values that were set by the user. A graphical representation using a bar for each module is included, together with an exact power number reported on the blue area on the left of the result window.

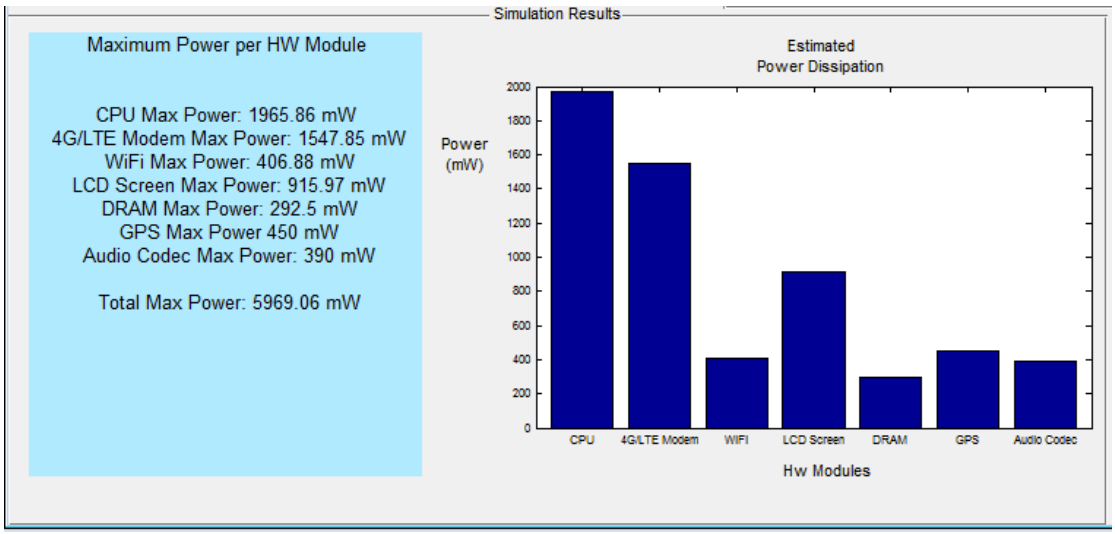

Figure 22: Result Window – Demonstration of Maximum Power per Hardware Module

The second window, – located on the right bottom side of the GUI –, displays the energy being consumed by each of the applications that were selected. Again a graphical representation using a bar for each application's energy is included, together with an exact energy number reported on the blue area on the left of the result window.

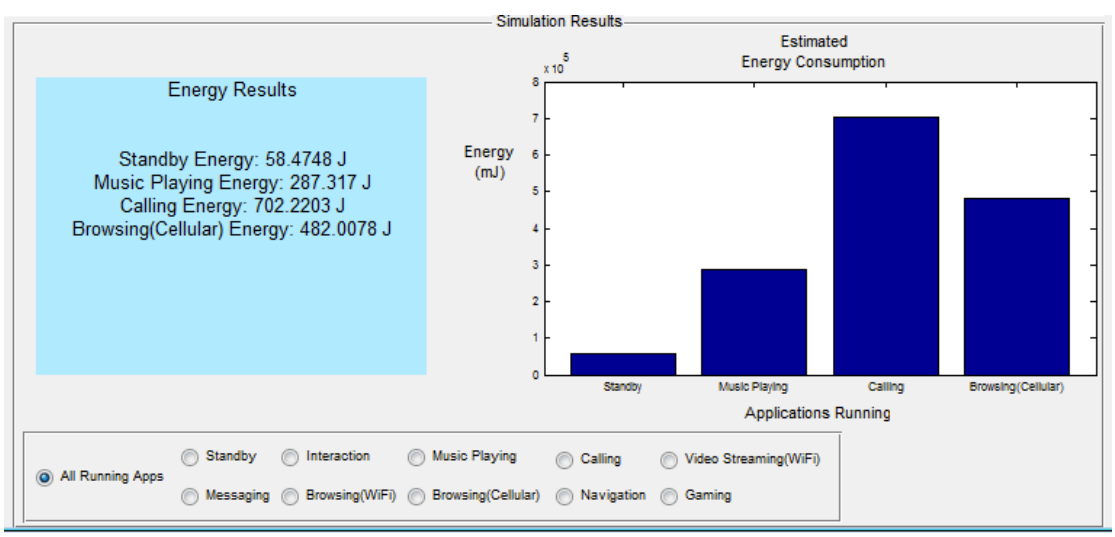

Figure 23: result Window – Demonstration of Energy Consumed per Application

The third window, – located on the right center side of the GUI -, displays a graph of the total power being dissipated over time for the duration of the simulation. Parallel applications' power can also be distinguished here if parallel execution time windows exist. On the left of the result window an exact number of the overall energy consumed by the simulated scenario is reported on the blue area, together with the overall simulation's start and end times.

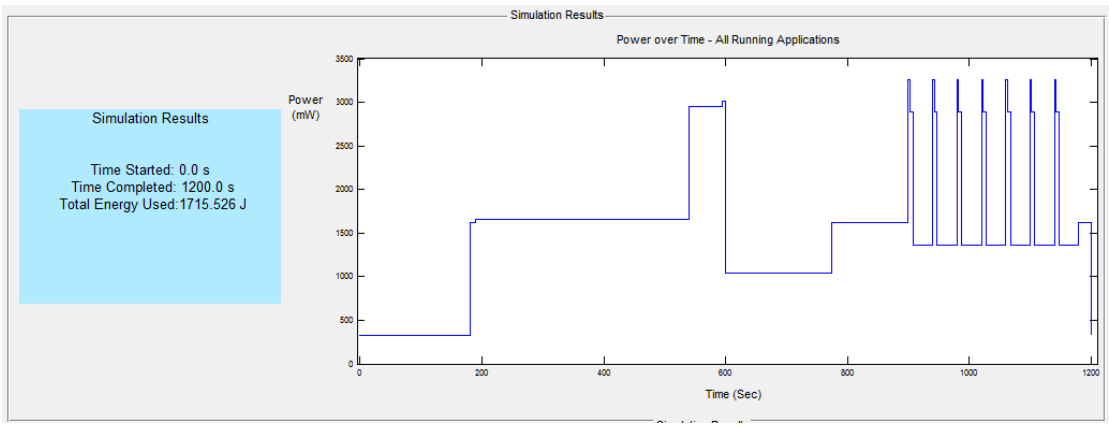

Figure 24: Result Window – Demonstration of Overall Simulation Power Over Time Plot

As it can be seen on the bottom of the second results window, ratio buttons exist for each of the available applications, as well as for the overall simulation (All running applications). When one of the selected applications' ratio buttons is checked and the run button is pressed again, the third window changes, now reporting the average power of each of the hardware modules used during the specified application's execution time (bars on the left side). Also the power profile – power over time – for the selected application appears, demonstrating the different power states of the application (graph on the right side).

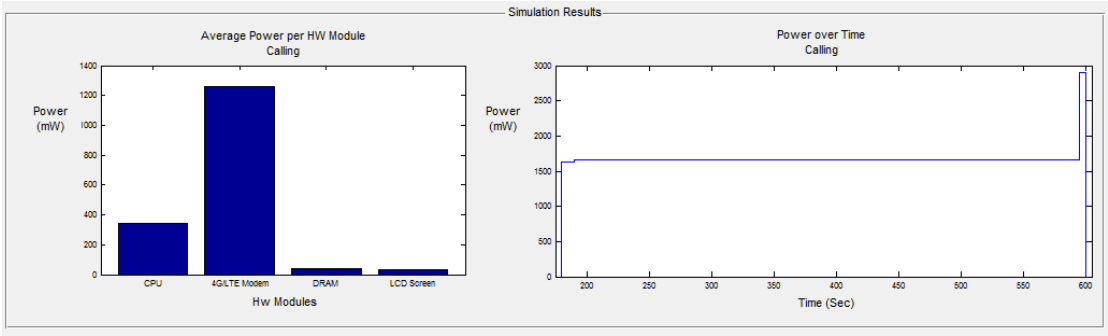

Figure 25: Result Window - Demonstration of Application's Power Over Time Plot

Notes:

- An important thing to mention here is that for parallel execution, the energy reported for each of the two parallel applications in the second results window and the power over time graph shown in the third one – if the specific application is selected using the ratio button -, is the same as the one that would be reported if the application was the only one running during that time frame. This is an overestimation of each of the applications' energy consumption when executed on parallel but it is shown for the user to realize the power dissipation and contribution in total power of each of the two parallel applications. This feature – although important in order to examine potential power gains of parallel execution over sequential execution of same number of applications-, can't be offered by real time power estimation, which base their calculations on the collection of the hardware modules' parameter values.

- The overall energy consumption calculation and the power over time graph of the entire simulation, – shown in the third results window if the overall simulation (All running applications) ratio button is selected –, takes into account the parallel power of the two applications - calculated using the parallel algorithm presented in chapter 6 “Parallel Execution Algorithm”-, to display an accurate result.

Chapter 8: Demonstration of Power Simulator

Two different usage profiles were selected and simulated using SPEED to demonstrate the various features of the tool described in the previous section.

8.1 1st Usage Profile – Sequential Execution

The simulated profile shown below demonstrates the sequential execution of applications, as well as the effect of the hardware modules' parameter values on the power and energy dissipation of each application.

For this example the 1st available hardware configuration was selected. The hardware modules included can be seen below:

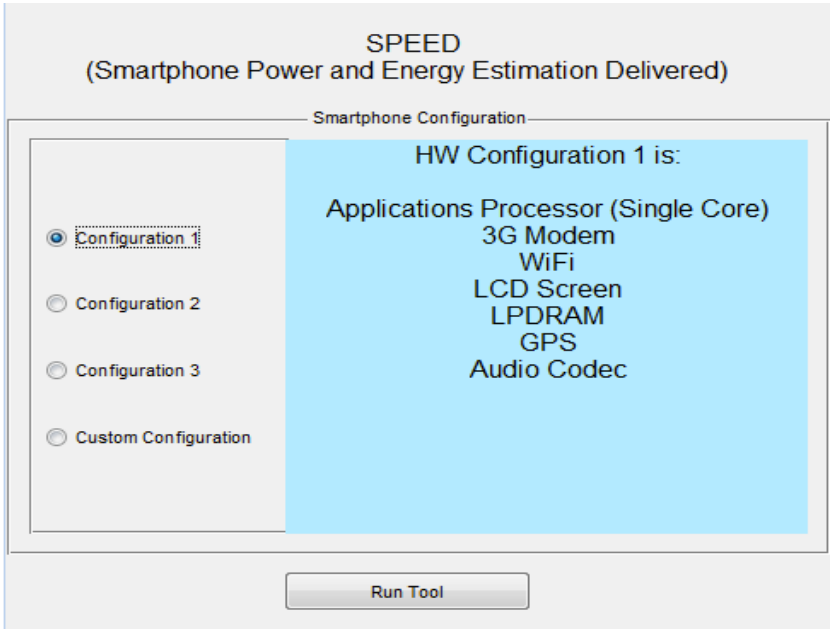

Figure 26: Configuration Selection for the 1st Usage Example

The application profiles to be simulated are then selected together with their start - end times. Also the parameter values of all the hardware modules listed above are set to their predefined values.

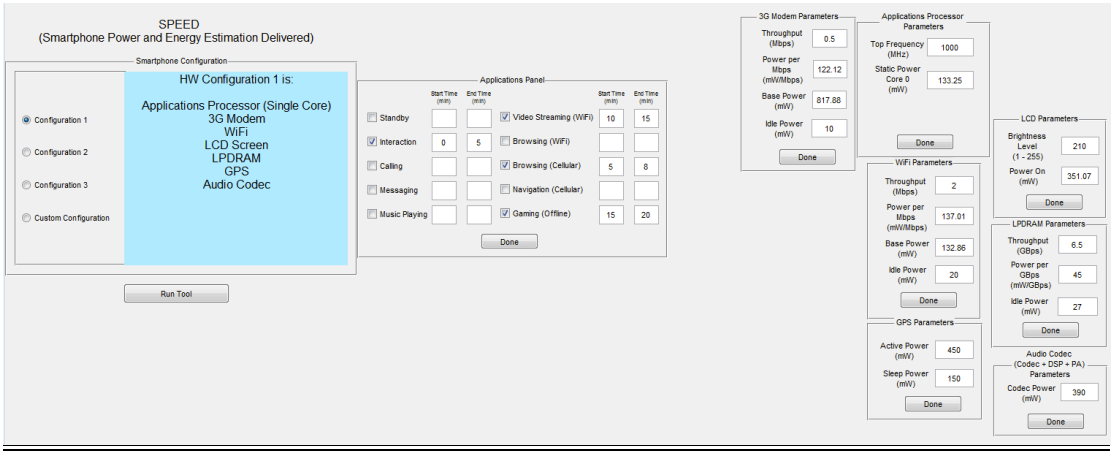

Figure 27: Application Selection and Parameter Setting for the 1st Usage Example

When the simulation starts the tool first sorts the selected applications from the first to the last one to be executed based on their start times. For this example the following sequence is produced:

- The Interaction application is executed from the 0 till the 5th minute.
- The Browsing (Cellular) application is executed from the 5th till the 8th minute.
- The Video Streaming application is executed from the 10th till the 15th minute.
- The Gaming application is executed from the 15th till the 20th minute.

Each of the applications' execution time windows is then divided into smaller time areas based on the power states of each application and their duration.

The time areas for two of the four applications simulated in this example are shown below:

Browsing (Cellular)

Time Start (sec)	Time End (sec)	Power State
300	324	1st
324	329	2nd
329	340	3rd
340	364	1st
364	369	2nd
369	380	3rd
380	404	1st
404	409	2nd
409	420	3rd
420	446	1st
446	451	2nd
451	460	3rd
460	470	4th
470	480	Interaction Power

Table 15: Execution Time Window Division per Power State of a Sequential Application of the 1st Usage Example

Video Streaming (Wifi)

Time Start (sec)	Time End (sec)	Power State
600	620	1st
620	817.5	2nd
817.5	890	3rd
890	900	4th

Table 16: Execution Time Window Division per Power State of a Sequential Application of the 1st Usage Example

After the simulation is done the following results are produced:

Maximum active power of each of the hardware modules contained in the selected configuration:

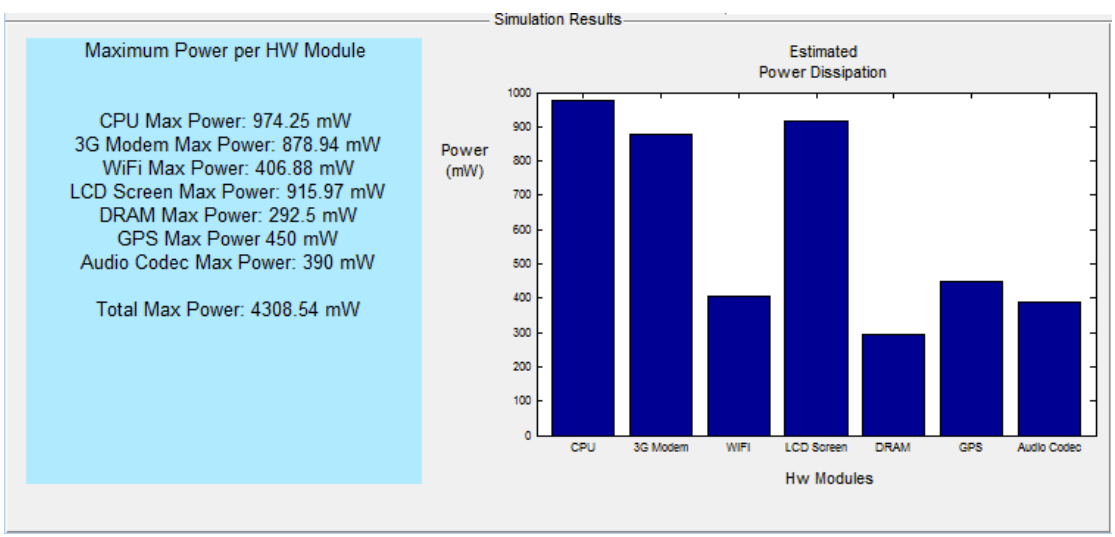

Figure 28: Maximum Power per Hardware Module for the 1st Usage Example

Overall power over time plot and total energy used for the entire simulated usage profile:

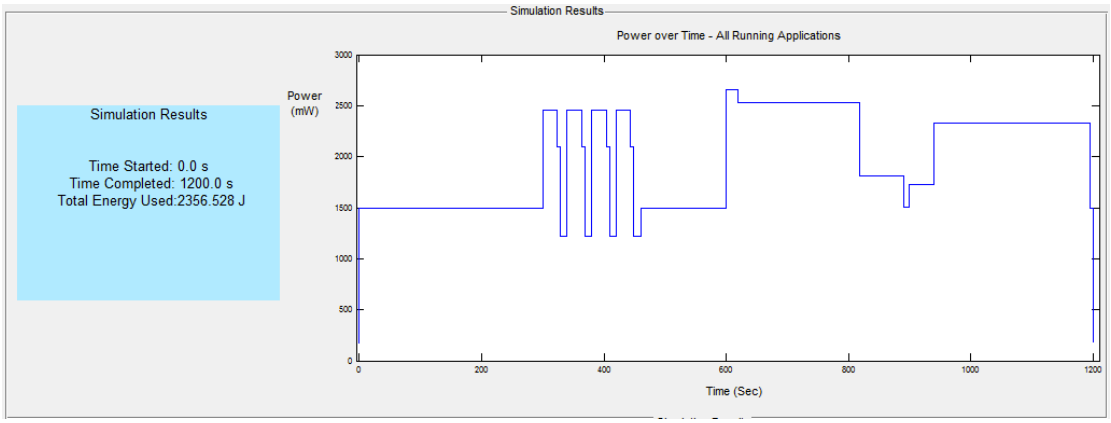

Figure 29: Overall Simulation Power Over Time Plot for the 1st Usage Example

Total energy consumed by each of the applications simulated:

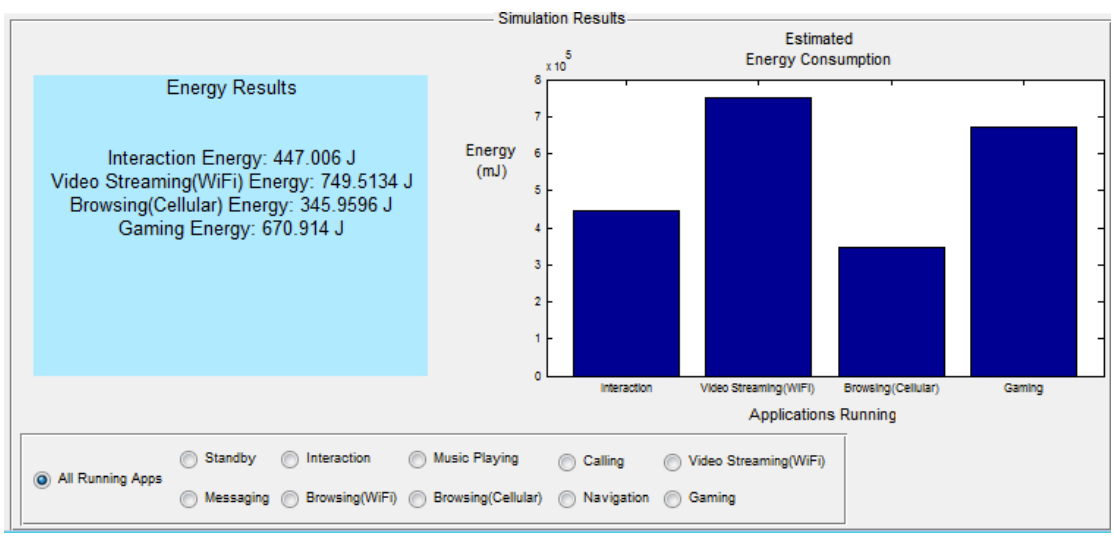

Figure 30: Energy Dissipation per Application for the 1st Usage Example

Power over time plots of two of the applications selected:

Browsing (Cellular)

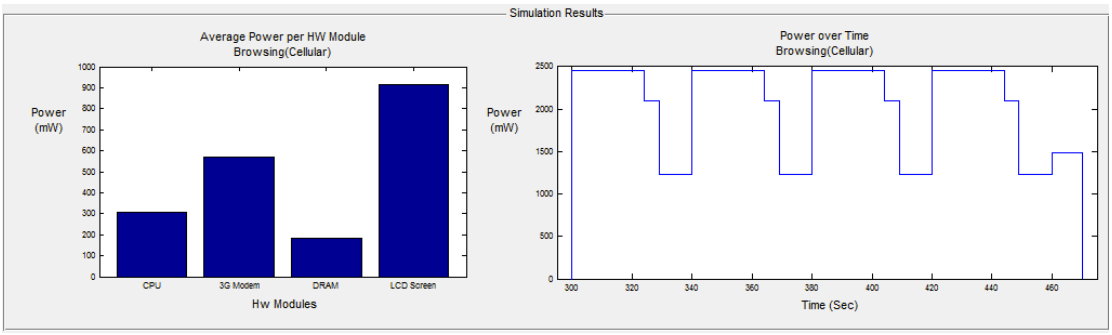

Figure 31: Power Over Time and Average Hw Modules Power for a unique application for the 1st Usage Example

Video Streaming (Wifi)

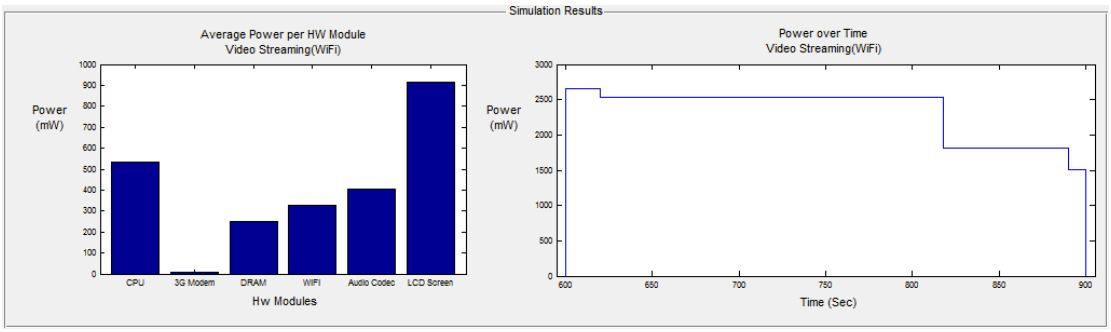

Figure 32: Power Over Time and Average Hw Modules Power for Selected Application for the 1st Usage Example

Some of the hardware modules' parameter values are then changed:

- CPU Frequency set to 1.2 GHz (from 1 GHz that is the predefined value).
- Modem 3G throughput set to 1Mbps (from 0.5 that is the predefined value).
- LCD screen brightness level set to 255 (from 210 that is the predefined value).

The effect of the parameters values can be seen on the new simulation results, some of them reported below:

New maximum active power of each of the hardware modules contained in the selected configuration:

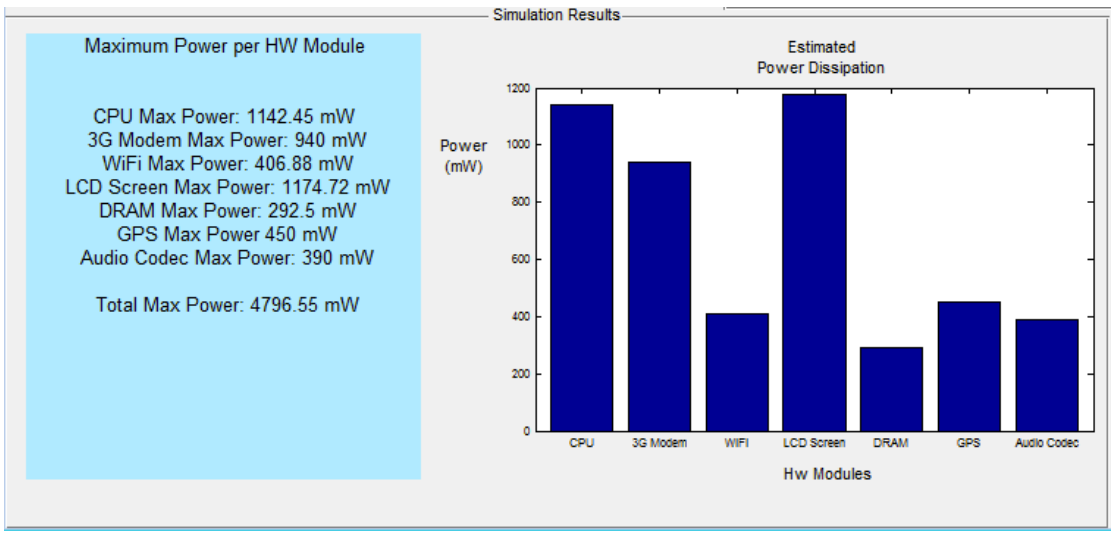

Figure 33: New Maximum Power per Hw Module for the 1st Usage Example

New Overall power over time plot and total energy used for the entire simulated usage profile:

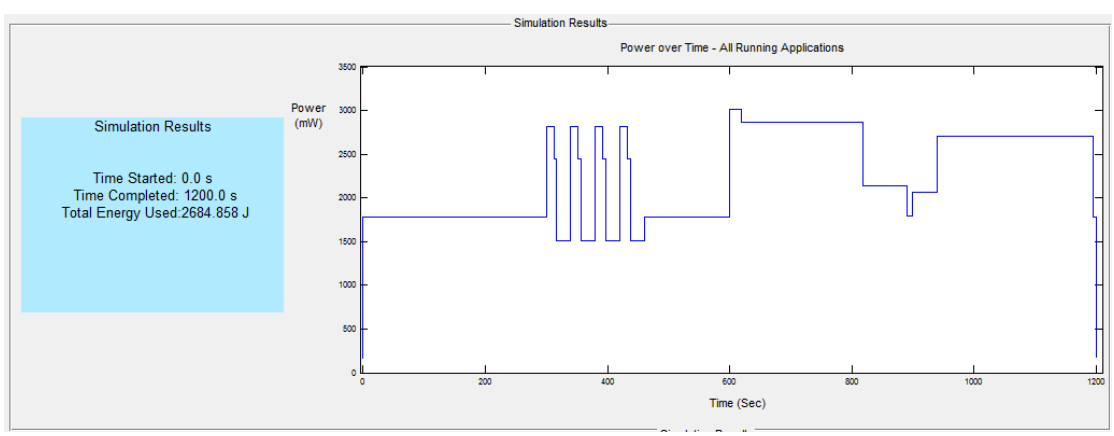

Figure 34: New Overall Simulation Power Over Time Plot for the 1st Usage Example

Also the periods of the power states of many applications are affected by the new parameter values, as seen by the new power profile of the application demonstrated below:

Browsing (Cellular)

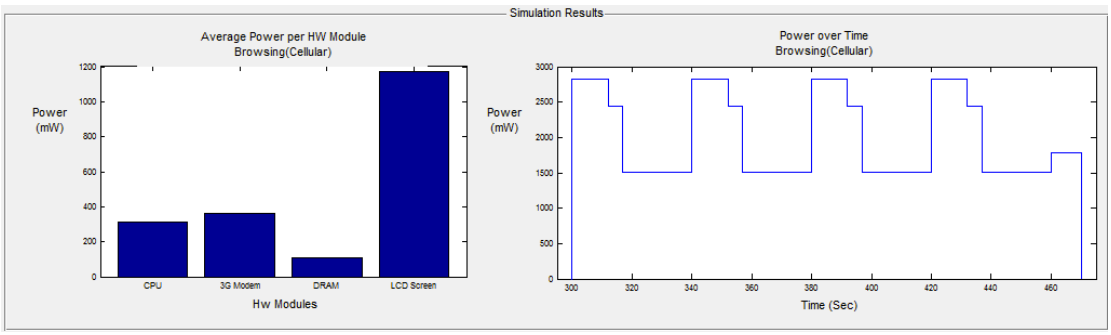

Figure 35: New Power over Time and Average Hw Modules Power Results for Selected Application for the 1st Usage Example

<u>Note:</u>

- As seen on the application selection phase of the above usage profile, no application is selected to run from the 8th till the 10th minute when the video streaming application starts. For any time gap detected on the simulation time window the tool adds the power of the interaction application, since it is assumed that the user keeps interacting with the phone during this time as he/she would have when using a real device.

8.2 2nd Usage Profile – Parallel Execution

The simulated profile shown below demonstrates the parallel execution algorithm, as well as the DVFS capabilities of the CPU's power model.

For this example the 3rd available hardware configuration was selected. The hardware modules included can be seen below:

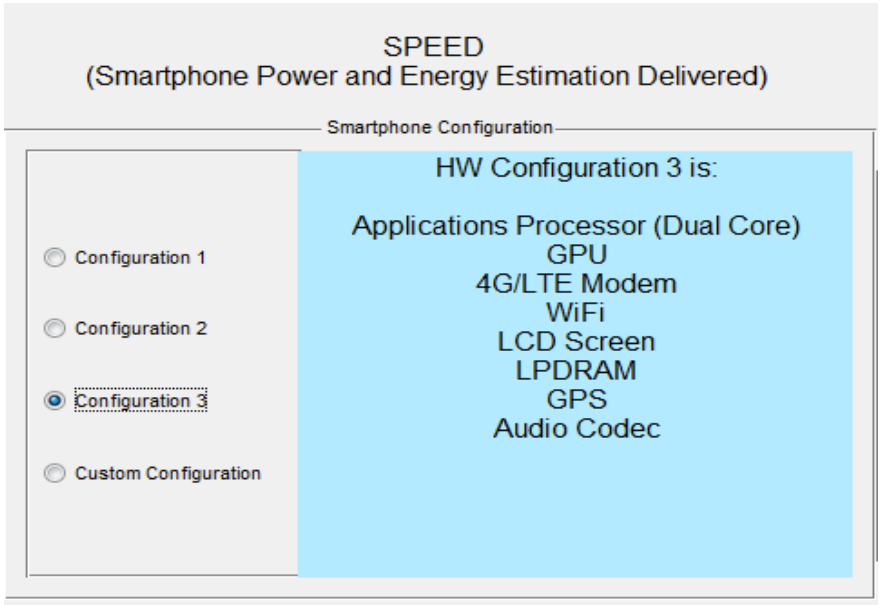

Figure 36: Configuration Selection for the 2nd Usage Example

The application profiles to be simulated are then selected together with their start - end times. Also the parameter values of all the hardware modules listed above are set to their predefined values.

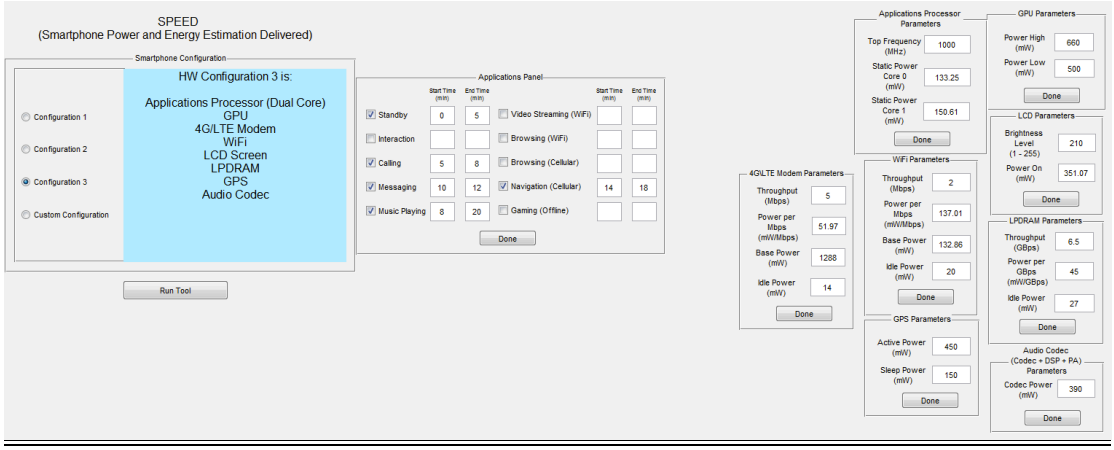

Figure 37: Application Selection and Parameter Setting for the 2nd Example

When the simulation starts the tool first sorts the selected applications from the first to the last one to be executed based on their start times. For this example the following sequence is produced:

- The Standby application is executed from the 0 till the 5th minute.
- The Calling application is executed from the 5th till the 8th minute.
- The Music Playing application is executed from the 8th till the 20th minute.
- The Messaging application is executed from the 10th till the 12th minute.
- The Navigation application is executed from the 15th till the 20th minute.

After the initial sorting the parallel execution algorithm starts. Its first task is to analyze the simulation time window and identify each time area where sequential or parallel execution exists. For the demonstrated usage profile the following time windows are created:

Time Start (sec)	Time End (sec)	Running App 1	Running App 2
0	300	Standby	-
300	480	Calling	-
480	600	Music Playing	-
600	720	Music Playing	Messaging
720	900	Music Playing	-
900	1200	Music Playing	Navigation

Table 17: Division of Simulation Time in Sequential and Parallel Execution Time Windows for the 2nd Usage Example

Each of the above time windows is then further divided into power states for each application, based on the applications' power profiles shown in detail in chapter 5 "Application Power Profiles].

The sequential execution time areas are created similarly as shown in the previous demonstration example.

The time matrix and power states matrix for each of the two parallel applications in both parallel regions are shown below:

<u>Parallel region 1 (600 – 720s)</u>

Music Playing

Time Start (sec)	Time End (sec)	Power State
600	720	2nd

Table 18: Time and Power State Matrix of one Parallel Executed Application in the 1st Parallel Region of the 2nd Usage Example

Messaging

Time Start (sec)	Time End (sec)	Power State
600	655	1st
655	657	2nd
657	660	3rd
660	715	1st
715	717	2nd
717	720	3rd

Table 19: Time and Power State Matrix of one Parallel Executed Application in the 1st Parallel Region of the 2nd Usage Example

The algorithm then creates unified time and power states matrixes for both applications, combining the time areas of each of the two parallel applications.

The unified time and power states matrixes for this example are shown below:

Time Start (sec)	Time End (sec)	Music Playing Power State	Messaging Power State
600	655	2nd	1st
655	657	2nd	2nd
657	660	2nd	3rd
660	715	2nd	1st
715	717	2nd	2nd
717	720	2nd	3rd

Table 20: Unified Time and Power State Matrix of the 1st Parallel Region of the 2nd Usage Example

The same analysis and matrix creation procedure takes place for the second parallel execution window:

<u>Parallel region 2 (900 – 1200s)</u>

Music Playing

Time Start (sec)	Time End (sec)	Power State
900	1195	2nd
1195	1200	3rd

Table 21: Time and Power State Matrix of a Parallel Executed Application in the 2nd Parallel Region of the 2nd Usage Example

Navigation

Time Start (sec)	Time End (sec)	Power State
900	900.8	1st
900.8	920	2nd
920	920.16	3rd
920.16	925	4th
925	925.16	3rd
925.16	930	4th
. . .	. . .	3rd state (lasts for 0.16s) and 4th state (lasts for 4.84 s) alternated

1190	1200	5th

Table 22: Time and Power State Matrix of a Parallel Executed Application in the 2nd Parallel Region of the 2nd Usage Example

The unified time and power states matrixes for this example are shown below:

Time Start (sec)	Time End (sec)	Music Playing Power State	Navigation Power State
900	900.8	2nd	1st
900.8	920	2nd	2nd
920	920.16	2nd	3rd
920.16	925	2nd	4th
925	925.16	2nd	3rd
925.16	930	2nd	4th
. . .	. . .	2nd	3rd state (lasts for 0.16s) and 4th state (lasts for 4.84 s) alternated
1190	1195	2nd	5th
1195	1200	3rd	5th

Table 23: Unified Time and Power State Matrix of the 2nd Parallel Region of the 2nd Usage Example

Finally the power is calculated for each of the time areas with sequential and parallel execution. For the sequential time windows the power estimation is based on the applications' power profiles and is similar to the demonstrated one in the previous example.

For the parallel time windows the total power for each time area shown above is calculated for all hardware modules, – except the CPU –, following the procedure described in chapter 6 “Parallel Execution Algorithm”.

To take into account the dynamic voltage and frequency scaling capabilities of the CPU, – supported by the power model used in this work- , the frequency and utilization ratio of each core for both applications at each time area is reported. Using the process described in chapter 6, the best fitted frequency and utilization rate that can support both applications running in parallel is selected for each core for this time area. These values are then applied to the CPU power model and the power reported is added to the final parallel power for the specific time area.

Below the CPU frequency and utilization ratio selection is demonstrated for two time areas of each parallel time windows existing in this example:

Time areas of 1st parallel time window:

Time Start (sec)	**Time End (sec)**	**Music Playing CPU Frequency& Utilization**		**Messaging CPU Frequency & Utilization**		**Parallel Execution CPU Frequency & Utilization**	
		Core 0	**Core 1**	**Core 0**	**Core 1**	**Core 0**	**Core 1**
655	657	300MHz - 30%	300MHz - 0%	300MHz - 30%	300MHz - 0%	300 MHz - 60%	300 MHz - 0 %
657	660	300MHz - 30%	300MHz - 0%	300MHz - 30%	300MHz - 30%	300 MHz - 60%	300 MHz - 30 %

Table 24: Frequency and Utilization Ratio Results of Parallel Execution for Time Areas of the 1st Parallel Region of the 2nd Usage Example

Time areas of 2nd parallel time window:

Time Start (sec)	Time End (sec)	Music Playing CPU Frequency & Utilization		Navigation CPU Frequency & Utilization		Parallel Execution CPU Frequency & Utilization	
		Core 0	Core 1	Core 0	Core 1	Core 0	Core 1
900	900.8	300MHz - 30%	300MHz - 0%	600MHz - 30%	300MHz - 20%	600 MHz - 45%	300 MHz - 20 %
920.16	925	300MHz - 30%	300MHz - 0%	300MHz - 20%	300MHz - 20%	300 MHz - 50%	300 MHz - 20 %

Table 25: Frequency and Utilization Ratio Results of Parallel Execution for Time Areas of the 2nd Parallel Region of the 2nd Usage Example

After the execution is done the following results are produced:

Maximum active power of each of the hardware modules contained in the selected configuration:

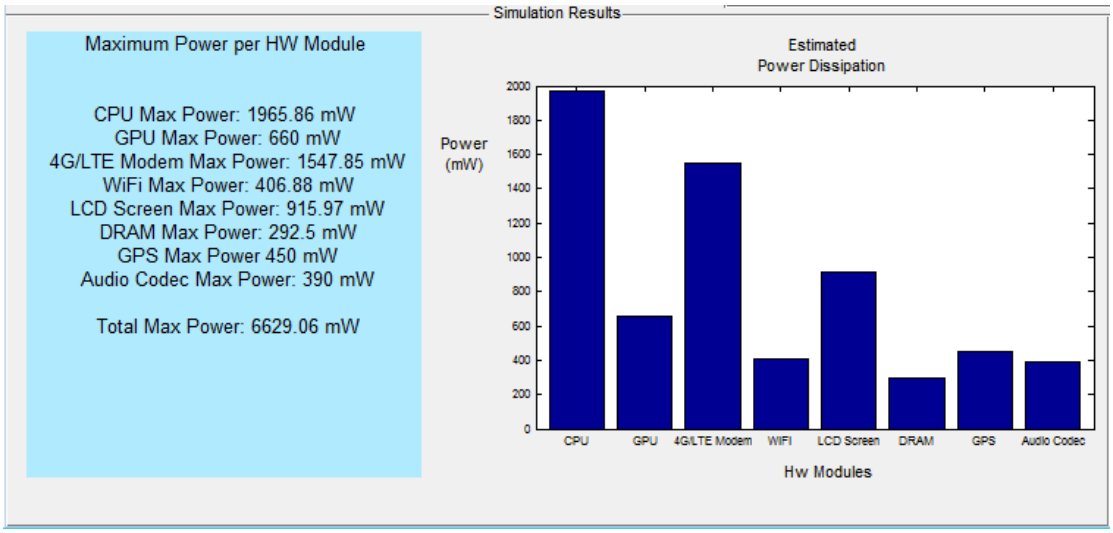

Figure 38: Maximum Power per Hw Module for the 2nd Usage Example

Overall power over time plot and total energy used for the entire simulated usage profile:

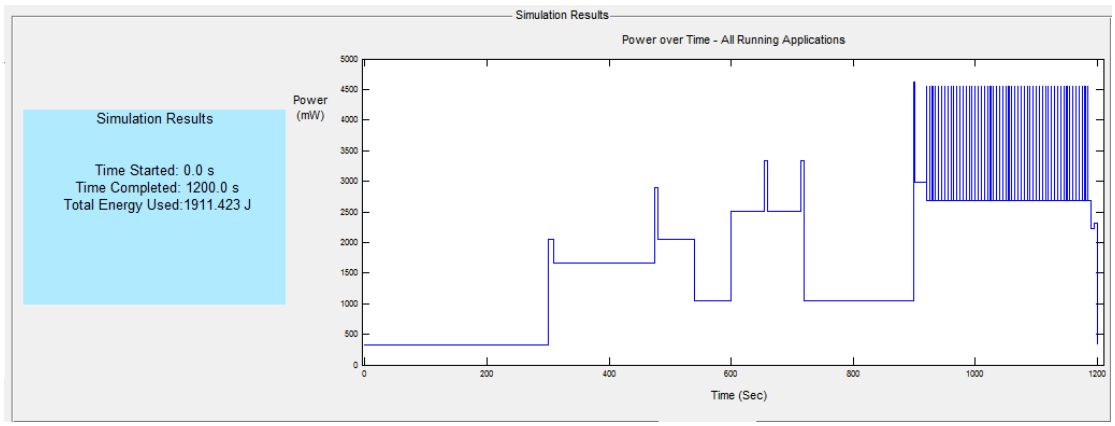

Figure 39: Overall Simulation Power Over Time Plot for the 2nd Usage Example

Total energy consumed by each of the applications selected in the simulation:

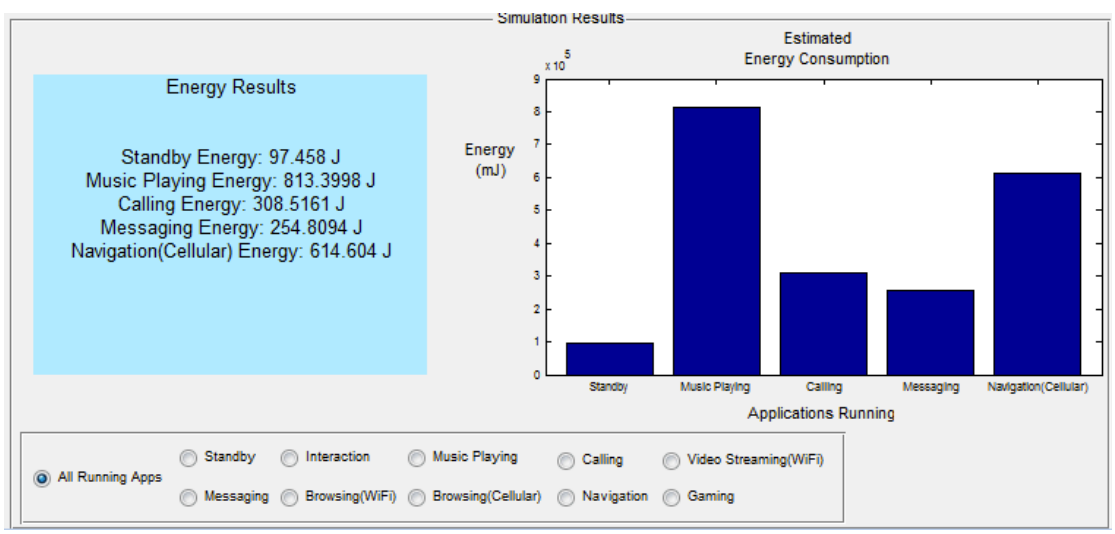

Figure 40: Energy Dissipation per Application for the 2nd Usage Example

Power over time plots of three of the selected applications:

Music Playing

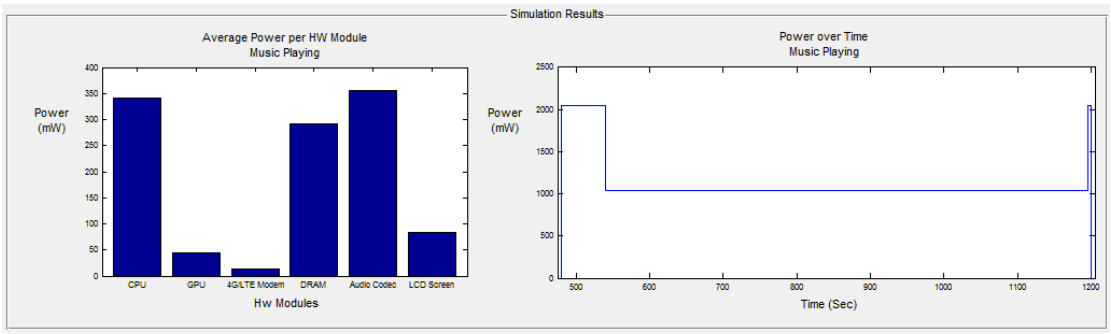

Figure 41: Power Over Time and Average Hw Module Power for a Selected Application for the 2nd Usage Example

Messaging

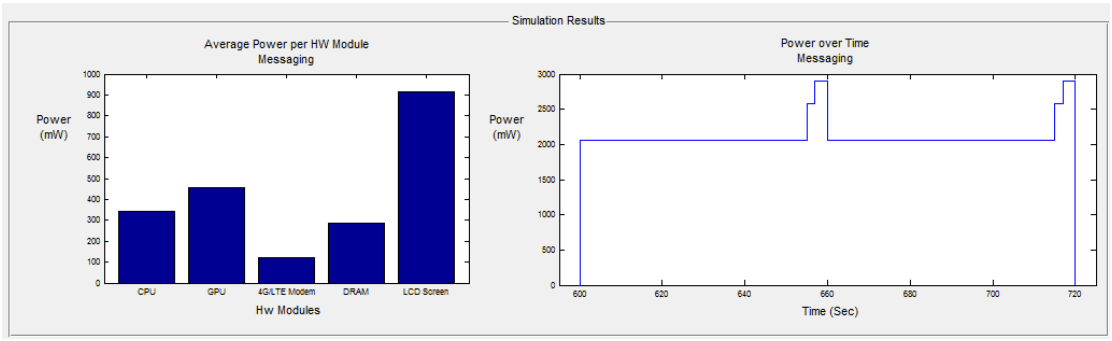

Figure 42: Power Over Time and Average Hw Modules Power for a Selected Application for the 2nd Usage Example

Navigation (Cellular)

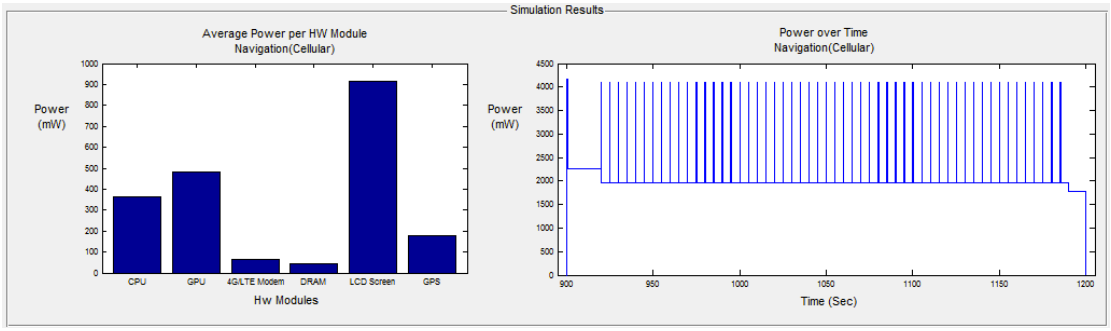

Figure 43: Power Over Time and Average Hw Modules Power for a Selected Application for the 2nd Usage Example

Notes:

- As seen above, the total energy consumed is smaller that the sum of all individual applications' reported energy, as parallel execution power consumption is less than if the two applications where running sequentially. This is explained in more detail in chapter 7 "Power Simulator's User Interface".

- The frequency of the CPU in this example was set to 1 GHz. The Low Frequency and Medium frequency states of the CPU are used on the time areas demonstrated above.

Chapter 9: Conclusion and Future Work

9.1 Conclusion

The work of this thesis presented in detail above, aimed on the creation of a smartphone simulator that can accurately estimate power and energy dissipation based on selected usage profiles. This tool tried to deal with all of the disadvantages of previously proposed solutions while ensuring ease of use and expandability.

First described was the power modeling of every major smartphone's hardware module. The models selected are as generalized as possible, each consisting of a set of user-defined parameters that can be chosen to represent the behavior of any similar hardware module available on the market. The number of user defined parameters for each module was kept small to ensure ease of use. Also the models were further tuned based on previously acquired results and measurements conducted on real devices using Google's PowerTutor, to further improve accuracy.

The created applications' profiles where then presented. Ten of the most frequently used mobile applications were modeled, each one consisting of a number of power states. These states help resemble the differences in power dissipation over time when an application is running with increased accuracy compared with flat application power dissipation models created in previous work.

The parallel execution algorithm created as part of this thesis was then described. This algorithm can distinguish the time windows where two applications are executed on parallel and can calculate the total power dissipation of this area based on the power state of each of the parallel applications. At the same time it avoids overestimating power by computing the required frequency and utilization rate for the CPU to run both applications. These values are then used to calculate the new CPU power, effectively modeling the DVFS capabilities of modern CPUs. This novel algorithm implementation allows for the creation of realistic usage profiles for simulation, where applications are executed in parallel or sequentially, same as in real smartphones.

Finally the graphical user interface (GUI) of the smartphone simulator was presented. The GUI was created to be as user friendly and simple as possible both for the hardware modules' parameter setting as well as for the usage profile creation with the selection of applications to be executed and the setup of their execution time windows. Also a variety of power and energy related results are presented after the simulation (Max active power of each hardware module, energy dissipation and power over time for each selected application, total energy dissipation and total power consumed over time for the entire simulated time window).

In chapter 8 "Demonstration of Power Simulator" two different usage profiles were simulated using the developed tool. The first one demonstrated the sequential execution of applications and the effect of hardware modules' parameter values in the power dissipation and time period of each of the application's power states. The second one presented the parallel execution capabilities of the created simulator and the effect of parallel execution on the total power dissipation over time.

9.2 Future Work

Work on this thesis was focused on every part needed in the creation of a smartphone power and energy simulator. The power models selected for each of the hardware modules were based on extensive previous work presented on chapter 2 “Previous Work”, with small modifications/additions made wherever needed. Power models were also further tuned based on measurements on real smartphones using PowerTutor.

In the future physical measuring equipment can be used to measure power in a variety of smartphone devices while running different applications. This can help collect precise data that can be used to further tune the above described power models. Also more hardware modules and application profiles can be added to the simulator to support future smartphone implementations and popular applications. The existing applications can also be executed in smartphone devices connected with physical measuring equipment to help adjust the number power states of each application and their period with greater accuracy. Finally the tool’s graphical user interface can be further enhanced to include more results, as for example the power dissipated over time for each of the smartphone’s hardware modules for the entire simulation window.

Another future goal is the development of a web-based smartphone simulator that can use the same power models, application profiles and execution algorithms developed here, but it will run online, making it easier for the user to run simulations using any device with internet access capabilities, without having to install Matlab on their desktop or laptop in order to execute the tool.

References

[1] Zhang L., Tiwana B., Qian Z., Wang Z., Dick R. P., Mao Z. M., Yang L., "Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones", CODES+ISSS, (Oct 2010),http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5751489

[2] Shye A., Scholbrock B., Memik G., "Into the Wild: Studying Real User Activity Patterns to Guide Power Optimizations for Mobile Architectures", MICRO (DEC 2009),http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5375354

[3] Kim M., Kong J., Woo S., "Enhancing Online Power Estimation Accuracy for Smartphones", IEEE - TCE (May 2012),http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6227431

[4] Carroll A., Heiser G., "An Analysis of Power Consumption in a Smartphone", USENIX (2010), http://static.usenix.org/event/usenix10/tech/full_papers/Carroll.pdf

[5] Kim M., Woo S., "Accurate GPU Power Estimation for Mobile Device Power Profiling", IEEE ICCE (2013),http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6486850

[6] Kahng A. B., Srinivas V., "Mobile System Considerations for SDRAM Interface Trends", SLIP (June 2011), http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6135437

[7] Joshi A., Sambamurthy S., Kumar S., John L., "Power Modeling of SDRAMs", University of Texas (2004), http://lca.ece.utexas.edu/pubs/TR-040126-02.pdf

[8] Malladi K. T., Nothaft F. A., Periyathambi K., Lee B. C., Kozyrakis C., Horowitz M., "Towards Energy-Proportional Datacenter Memory with Mobile DRAM", IEEE – ISCA (June 2012), http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6237004

[9] Deng S., Balakrishnan H., "Traffic-Aware Techniques to Reduce 3G/LTE Wireless Energy Consumption", CoNEXT (Dec 2012),http://conferences.sigcomm.org/co-next/2012/eproceedings/conext/p181.pdf

[10] Balasubramanian N., Balasubramanian A., Venkataramani A., "Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications", IMC (Nov 2009), http://ciir-publications.cs.umass.edu/getpdf.php?id=904

[11] Huang J., Qian F., Gerber A., "A Close Examination of Performance and Power Characteristics of 4G LTE Networks", MobiSys (June 2012),http://dl.acm.org/citation.cfm?id=2307658

[12] Qian F., Wang Z., Gerber A., Mao Z. M. Sen S., Spatscheck O., "Profiling Resource Usage for Mobile Applications: A Cross-layer Approach", MobiSys (June 2011), http://dl.acm.org/citation.cfm?id=2000026

[13] Ning K., "System-level memory power and performance optimization for system-on-a-chip embedded systems", (Aug 2008),http://hdl.handle.net/2047/d10017707

[14] Sklavos N., Touliou K., "A System-Level Analysis of Power Consumption & Optimizations in 3G Mobile Devices", NTMS (May 2007),http://link.springer.com/chapter/10.1007%2F978-1-4020-6270-4_19

[15] Im J., Cho M., Jung Y., Kim J., "Low-Power Low-Complexity MIMO-OFDM Baseband Processor for Wireless LANs", IEEE (2009),http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5117820