High-level Power Modeling of CC430 SoC

Strahinja Janković, *Student Member, IEEE*, and Lazar Saranovac, *Member, IEEE*

***Abstract*—Power management is an important aspect of modern embedded system design. In order to evaluate effectiveness of power management techniques power profiling is used. Most model-based power profiling for low power systems so far has been focused on cycle-accurate models, which are not useful for online use. We present high-level model-based power profiling, for CC430 System on Chip (SoC) as an example. Three SoC components are modeled: CPU, AD converter and RF core. Models are constructed using finite state machines and linear regression. Presented model has less than 4% mean error on run tests.**

***Index Terms* — Power model; Power profiling; System on Chip; Embedded system.**

I. Introduction

Embedded systems are present in great number in everyday life. Internet of Things (IoT) [1] has been a major influence on increase in number of deployed embedded systems. It is expected that number of devices connected in IoT will be between 20 and 30 billion by the year 2020 [2]. This huge number of deployed devices makes power management of these devices an important topic.

In order to quantify the quality of power management policy its effect on the power consumption of the system must be known. Power profiling is used to obtain that information. Power profiling can be measurement-based and model-based.

Measurement-based power profiling provides precise consumption data, but hardware support must be present. Also if higher measurement precision is required, the amount of data that is generated while measuring power consumption can be too complex to process in real time.

Model-based power profiling uses prebuilt model to estimate system's power consumption. Power consumption of the system is represented using power consumption of each component. Power consumption of a component is expressed as a function of relevant component parameters. Model-based power profiling is less accurate than measurement-based, but it does not require additional measurement hardware.

Depending on the amount of details that they incorporate and their purpose power models can be categorized into two classes: low-level power models and high-level power models.

Low-level power models are based on architecture-level details. Models created this way are not portable and require full characterization if they are to be used on different platform. They can be used to estimate power consumption of an embedded system based on a given workload, but they cannot be used to estimate power consumption in real time.

High-level power models can be used to estimate power consumption of the system using data that is available at run time. That data can come from hardware components like performance counters or operating system (OS) interfaces. High-level models can be updated or even constructed from scratch during execution. They are usually targeted at high-performance embedded systems running an operating system.

We present high-level power model for the CC430 SoC. Power consumption of CPU, AD converter and RF core are modeled and their utilization is used as model parameters.

The paper is organized as follows. In section 2 related work on low-level and high-level power modeling is presented. CC430 SoC peripherals are introduced in section 3. Model construction and models for CC430 SoC peripherals are presented in section 4. In section 5 constructed model is tested in two use cases. Conclusion is in section 6.

II. Related Work

Various low-level models have been previously presented [3]-[5]. They use instruction type and memory and peripheral access information to estimate power consumption for a given workload. In [3] authors have presented instruction-level energy/power estimation model for embedded systems. Authors in [4] have created energy/power consumption model for a microcontroller. In [5] authors have created cycle-accurate energy/power simulator that uses cycle-accurate models of components.

High-level models [6]-[8] use available hardware and software interfaces to relate events during execution to power consumption. In [6] authors are using available performance counters for online power estimation. Authors in [7] have created tools for online power model creation for mobile systems. Data for model parameterization is based on utilization of system components. OS System calls have been used in [8] as data for model parameterization.

III. CC430 System on Chip

CC430 [9] is System on Chip which integrates MSP430X processing core and CC1101 RF core. Three components of the system are analyzed: CPU, AD converter and RF core.

CPU is a MSP430X architecture RISC core. Operating frequency can be adjusted using internal digitally controlled oscillator (DCO) up to 20 MHz. Depending on selected operating frequency, core voltage can be adjusted to one of four supported levels. CPU supports several different low power modes, with power consumption going down to under

Strahinja Janković and Lazar Saranovac are with the School of Electrical Engineering, University of Belgrade, 73 Bulevar kralja Aleksandra, 11020 Belgrade, Serbia (e-mail: {jankovics,laza}@etf.bg.ac.rs).

1 µA at 3.3 V supply voltage.

AD converter is up to 12 bit Successive Approximation register type with software selectable sample and hold time. Sampling frequency can also be configured. Internal references of 1.5V, 2.0V and 2.5V are present and external reference can also be used.

RF core is CC1101 compatible. It operates in the sub-1GHz range and supports data bitrate up to 500kBaud. Different output power level settings can be configured.

IV. Model Construction

Power models are usually represented as finite state machines (FSM) or through mathematical equations obtained using regression. FSM approach uses state for each different configuration of parameters. It is better suited when parameters have discrete values. The problem is that complexity of state machine grows exponentially with number of parameters that are used and with increasing estimated power consumption granularity.

In regression based approach parameters are selected and regression is performed. Various types of regression have be used so far for power modeling [10]. Regression based approach is easy to implement, but accuracy can depend on the regression type.

A. Measurement setup

Power model is built based on measured power consumption of system in specific scenarios. Scenarios are selected so only one component is active while power consumption is measured. Measurement is done by keeping supply voltage constant and measuring current consumption. After measurement is finished, power model for a component is constructed.

High-side potentiostat technique [11] is used for power consumption measurement (Fig. 1). Voltage V_{CC430} = 3.3 V for all measurements.

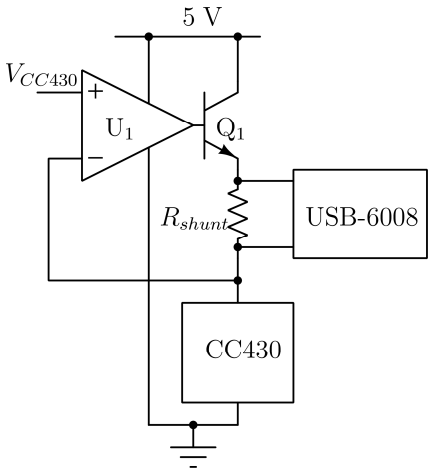

Fig. 1. Power consumption measurement system

Operational amplifier U_1 is used to keep voltage of CC430 SoC constant and transistor Q_1 is used to extend output current range of operational amplifier. Input bias current for used operational amplifier is negligible.

NI USB-6008 [12] data acquisition card with 12-bit analog inputs is used to measure voltage drop on shunt resistor R_{shunt}. Shunt resistor value is selected according to measured current value, so voltage drop on shunt stays under 1 V in order to use maximum accuracy for USB-6008.

B. CPU

CPU power consumption depends on selected operating frequency, core voltage level, mode of operation, utilization and workload boundness. Maximum supported operating frequency is 20 Mhz. Four core voltage levels are supported and they can be configured independently from SoC supply voltage. CC430 supports several low power modes, out of which we have modeled four.

CPU utilization is an important parameter for power consumption estimation. When CPU is in low power mode interrupts can wake up the CPU. Since power consumption in active mode is 10 to 1000 times higher than power consumption in low power mode, interrupt handling increases power consumption substantially, but it cannot be seen in the model unless utilization is modeled. CPU utilization is measured at runtime using hardware timers. Impact of hardware timers on power consumption is negligible. If RTOS is used CPU utilization can be obtained without additional code.

Whether workload is memory-bound or CPU-bound affects performance of model. However, boundness needs to be determined based on code analysis. If that data is available, it can be incorporated into model.

Three different CPU models have been created for both CPU-bound and memory-bound workloads: FSM, linear regression and hybrid.

FSM model for memory-bound workload is shown in Fig 2.

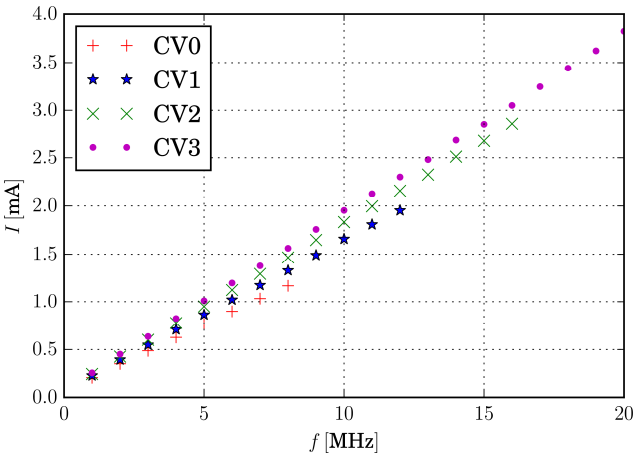

Fig. 2. CPU power consumption for different frequencies and core voltage levels for memory-bound workload

Low power mode power consumption is measured with all peripherals turned off. Power consumption for each low power mode is presented in Table I.

TABLE I
CC430 LPM POWER CONSUMPTION

I_{LPM} [μA]	CV0	CV1	CV2	CV3
LPM0	94	96	98	99
LPM1	93	95	97	98
LPM2	8	8	9	9
LPM3	5	5	5	5

Total CPU power consumption is calculated as (1)

$$P_{CPU} = (u \cdot I_{AM} + (1-u) \cdot I_{LPM}) \cdot V_{CC430} \quad (1)$$

where u is CPU utilization.

Linear regression and hybrid models have been created with only low power mode 3 taken into account.

Linear regression models for memory-bound (2) and CPU-bound (3) workload are shown in following equations:

$$I_{CPU} = 0.041 \cdot (cv == 1) + 0.135 \cdot (cv == 2) + 0.25 \cdot (cv == 3) + 0.178 \cdot f - 0.097 \text{ [mA]} \quad (2)$$

$$I_{CPU} = 0.05 \cdot (cv == 1) + 0.161 \cdot (cv == 2) + 0.298 \cdot (cv == 3) + 0.214 \cdot f - 0.128 \text{ [mA]} \quad (3)$$

In (2) and (3) cv is current core voltage and f is current CPU frequency.

Hybrid model consists of FSM where power consumption in a state is modeled by linear regression. It is constructed so it has four states for each core voltage level. Hybrid model for memory-bound workload is presented in table II.

TABLE II
CPU HYBRID MODEL

Core voltage	I_{CPU} [mA]
CV0	$0.17 \cdot f + 0.05$
CV1	$0.192 \cdot f + 0.057$
CV2	$0.211 \cdot f + 0.062$
CV3	$0.225 \cdot f + 0.063$

For linear regression and hybrid models CPU utilization u impacts CPU power consumption as in (4)

$$P_{CPU} = (u \cdot I_{CPU}(f = f_{AM}) + (1-u) \cdot I_{CPU}(f = 0)) \cdot V_{CC430} \quad (4)$$

where f_{AM} is operating frequency in active mode.

C. ADC

Power consumption of AD converter has been measured with different lengths of sampling time and different operating frequencies. Based on measurement results, ADC power consumption can only be expressed using linear regression (5).

$$I_{ADC} = on \cdot \frac{sht \cdot 0.302 + 13 \cdot (0.027 \cdot f + 0.206)}{sht + 13} + ref \cdot 0.093 \text{ [mA]} \quad (5)$$

In (5) on is 1 if ADC is turned on and 0 if it is turned off, f is ADC operating frequency, sht is selected value for sampling time which can take values from $4/f$ to $1024/f$, and ref indicates if voltage reference is used ($ref = 1$ if it is used, $ref = 0$ if not).

D. RF Core

RF Core is the biggest power consumer in CC430 SoC. Since all parameters have discrete values, power model is based on FSM. Table III holds power consumption values.

TABLE III
RF CORE POWER CONSUMPTION

State	I_{RF} [mA]
SPWD	0.018
SIDLE	1.689
STXON	9.498
STX(-30dBm)	12.559
STX(-12dBm)	14.755
STX(-6dBm)	18.216
STX(0dBm)	18.046
STX(10dBm)	31.542
STX(max power)	33.574
SRX	18.020

RF core is initially powered down (SPWD state). After it is powered on, it is in idle state (SIDLE). When transmission should occur, RF core first goes to STXON state and then transitions into STX(#) state, where # represents transmission power. When RF core should receive data, it goes into SRX mode.

E. CC430 SoC

Power consumption of CC430 SoC is estimated as sum of power consumption of all modeled components (6).

$$P_{CC430} = \sum_i P_i = P_{CPU} + P_{ADC} + P_{RF} \quad (6)$$

V. RESULTS AND DISCUSSION

Two scenarios have been used to evaluate performance of generated model. Power consumption is measured using setup in Fig. 1. Estimated power consumption is obtained using (6). Relative error between measured and estimated power consumption is calculated.

A. ADC averaging

In first scenario ADC is used to acquire 128 samples. After

they have been acquired, their value is averaged and new sampling period starts. Measured and estimated power consumption, as well as estimation error are shown in Fig. 3.

CPU utilization while ADC is acquiring data is 4%. Mean estimation error is 3.15%.

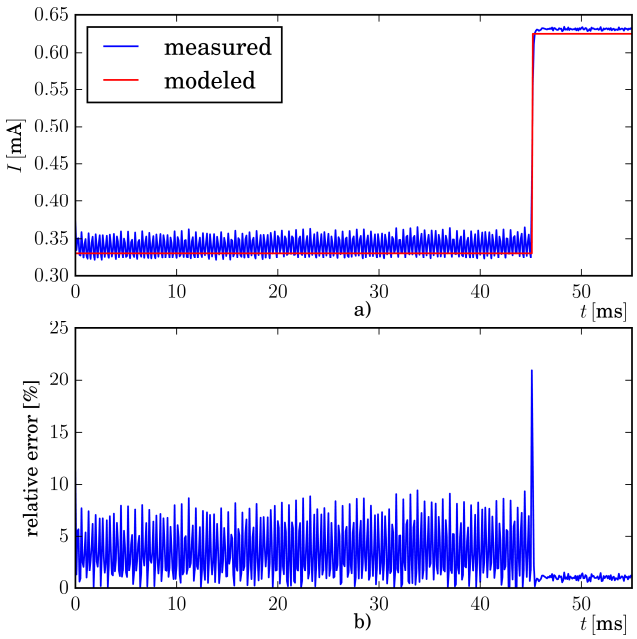

Fig. 3. ADC averaging demo: a) measured and modeled power consumption; b) relative error

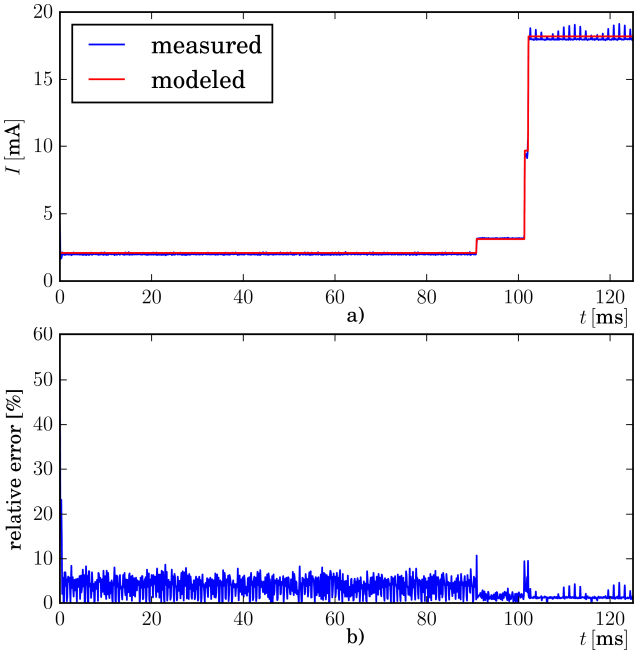

Fig. 4. RF transmission demo: a) measured and modeled power consumption; b) relative error

B. *RF transmission*

In second scenario ADC acquires 256 samples and they are averaged. After averaging, all samples are transmitted using RF core and new sampling period starts. RF core is configured with 0 dBm output power.

CPU utilization while ADC is acquiring data is 4% and transmitting data 6%. Mean estimation error is 3.35%.

VI. Conclusion

High-level power model of CC430 SoC has been presented. This universal modeling methodology can be applied to existing RTOS-based systems without modification since utilization measurement is integrated. It can also be adapted to other SoC.

Presented model can be used by the SoC at runtime to achieve better power savings since it is not computationally intensive. It can also be integrated into different power management algorithms to provide estimate of consumed power/energy and energy left in power source.

Acknowledgment

This work was supported by the Serbian Ministry of Education, Science and Technological Development through the project TR32043.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Computer Networks, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] A. Nordrum, "Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated," IEEE Spectrum: Technology, Engineering, and Science News, 18-Aug-2016. [Online]. Available: http://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated.

[3] M. Bazzaz, M. Salehi, and A. Ejlali, "An Accurate Instruction-Level Energy Estimation Model and Tool for Embedded Systems," IEEE Transactions on Instrumentation and Measurement, vol. 62, no. 7, pp. 1927–1934, Jul. 2013.

[4] V. Konstantakos, A. Chatzigeorgiou, S. Nikolaidis, and T. Laopoulos, "Energy Consumption Estimation in Embedded Systems," IEEE Transactions on Instrumentation and Measurement, vol. 57, no. 4, pp. 797–804, Apr. 2008.

[5] O. Celebican, T. S. Rosing, and V. J. Mooney III, "Energy Estimation of Peripheral Devices in Embedded Systems," in Proceedings of the 14th ACM Great Lakes Symposium on VLSI, New York, NY, USA, 2004, pp. 430–435.

[6] W. L. Bircher and L. K. John, "Complete System Power Estimation: A Trickle-Down Approach Based on Performance Events," in 2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Los Alamitos, CA, USA, 2007, vol. 0, pp. 158–168.

[7] L. Zhang et al., "Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones," in Proceedings of the Eighth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, New York, NY, USA, 2010, pp. 105–114.

[8] A. Pathak, Y. C. Hu, M. Zhang, P. Bahl, and Y.-M. Wang, "Fine-grained Power Modeling for Smartphones Using System Call Tracing," in Proceedings of the Sixth Conference on Computer Systems, New York, NY, USA, 2011, pp. 153–168.

[9] "*CC430 Family User's Guide*", Texas Instruments Inc., Dallas, TX, 2013.

[10] J. C. McCullough, Y. Agarwal, J. Chandrashekar, S. Kuppuswamy, A. C. Snoeren, and R. K. Gupta, "Evaluating the Effectiveness of Model-based Power Characterization," in Proceedings of the 2011 USENIX Conference on USENIX Annual Technical Conference, Berkeley, CA, USA, 2011, pp. 12–12.

[11] Janković, S.P. and Drndarević V.R, "Microcontroller Power Consumption Measurement Based on PSoC," Telfor Journal, vol. 8, no. 1, pp. 44–49, Jul. 2016.

[12] "*NI USB-6008/6009 User Guide*", National Instruments, TX, 2015.