Energy Proportional Management of Residential Gateways

Mirela Simonović, Vojin Živojnović, Davorin Mista, Strahinja Janković, and Lazar Saranovac

Abstract — **We are presenting a novel methodology for energy proportional management of electronic devices on the example of a triple-play broadband residential gateway. Using the newly developed formal abstraction of hardware and software, combined with the associated design tools, it was possible to automate the generation of the energy management software and meet the regulatory energy requirements. The resulting energy savings amounted to 31%, 38% and 71% in the active, idle and sleep modes, respectively. Based on the expected 2014 worldwide gateway production, the achieved savings are predicted to reach half of the annual output of a 500MW coal-fired power plant.**

Keywords — **Energy Proportional Computing, Power Management, Energy Design and Management Automation, Unified Hardware Abstraction, Residential Gateway.**

I. Introduction

The users of the newest residential communication devices expect new exciting services from the electronic equipment in their homes, like high resolution video streaming or home security management. Such energy hungry features require highly efficient power management not only to keep the devices small and fan-less, but also to meet increasingly stringent national regulations to save energy.

Energy management for electronic devices is based on controlling the clock frequency, supply voltage and operational state of the individual device components without impacting end-user quality of service. Although the basic principle is simple, the implementation in real life devices has proven to be a major challenge. The complexity stems from the large number of hardware components and their interdependencies, intricate hardware-software relationships, and unavoidable latencies due to the natural inertia of hardware and software.

These factors complicate the collaboration between hardware developers, device integrators and software developers working on the same device. As a consequence, the energy savings potential in the hardware is either underutilized by the software or not utilized at all, in both cases delivering suboptimal products.

The technology presented here leverages a formal description of the device hardware and application specific software to generate a target-specific and optimized run-time energy management software. The automatically generated software centralizes the information collection and decision making into one process, and thus has full control over all aspects of the energy requirements and consumption.

In this paper we have focused on the application of the new technology to a production level triple-play (phone, TV and Ethernet) residential gateway device and the analysis of the achieved energy savings.

II. Energy Proportional Systems

The standard approach to electronic design results in devices that guarantee optimal energy efficiency at highest levels of performance. In actual use, however, such devices spend most of their time below peak performance levels, thereby operating at points which are not optimized for energy efficiency. In mobile applications this leads to shorter battery life, in server farms to higher infrastructure and maintenance costs, and in plug load consumer electronics to higher home energy bills and environmental impact.

Electronic devices are energy converters. They convert the electrical energy from a battery or electrical plug into thermal and electro-magnetic energy dissipated by the circuitry and thereby generate some useful side effects, like changing the state of local or remote memory locations. These side effects represent the useful work expected by the users. The goal of energy proportional systems is to keep direct proportionality between the useful work and the energy consumed across the widest range of device operations.

Early methods to achieve energy proportionality relied on dynamic voltage and frequency scaling (DVFS) [1] to adjust to the work to be performed by the device. With the smaller transistor geometries and the growing complexity of the digital circuitry DVFS was gradually replaced by other low-power techniques, such as power and clock gating. The higher integration of semiconductors and the sharp drop in the price per transistor opened the opportunity to add additional circuitry to the electronic devices in order to reach energy proportionality. Similar to memory and bus hierarchies, such electronic devices

Mirela Simonović is a Ph.D. candidate at the School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia.

Mirela Simonović (mirela.simonovic@aggios.com), Dr Vojin Živojnović and Davorin Mista are with AGGIOS Europe d.o.o, Ruzveltova 1A, 11120 Belgrade, Serbia.

Strahinja Janković and Dr Lazar Saranovac are with the School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia.

The authors would like to thank Mindspeed Technologies, Newport Beach, CA (USA) for their support and collaboration on this project and National Instruments (USA), Agilent Technologies (USA) and ARM (UK) for their hardware and software donations. The project is financed by the Innovation Fund, within Innovation Serbia Project funded by EU IPA I 2011 and administered by the World Bank.

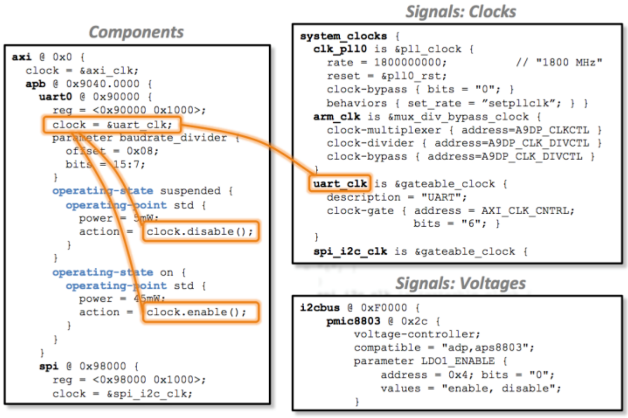
contain processor and subsystem hierarchies aimed at reducing the energy consumption with no or minimal impact on the quality of service. A well known example is the audio playback on the iPhone, which is executed on a dedicated, essentially redundant DSP circuitry that offers up to 40 hours of play time. If executed on the main application subsystem, the play time would drop to 8 hours or less. Other examples are the auxiliary cores found in most modern systems on a chip (SoC) [2], [3] and the big.LITTLE concept of ARM [4].

The main challenge with subsystem hierarchies is the trade-off between energy proportionality and system responsiveness caused by latencies while transitioning the execution to the alternative subsystem. This challenge is putting a significant burden on the hardware and software design methodology for such devices. In particular, there are immense challenges on the hardware-software divide, the traditionally weak point of electronic devices.

III. The Unified Hardware Abstraction

In order to cope with the above challenges, we are proposing a new formal and automated approach to the design of energy proportional systems. In the essence of our approach is the Unified Hardware Abstraction (UHA), which allows us to formally describe the hardware system, the application software, as well as the anticipated application scenarios and thus the useful work to be performed by the device. Such UHA descriptions are then used to automatically generate energy proportional power management software for the target using design-automation tools [5].

The UHA description of the hardware is based on a programmer’s view of components, representing actual hardware blocks. Components have operating states and points representing different power/performance configurations of the hardware block. Transitions between those states are mostly performed through changes of signals, such as clocks, reset-lines and voltages. Those are described explicitly in UHA as well, as shown on Fig. 1.



Components

```
axi @ 0x0 {
  clock = &axi_clk;
  apb @ 0x9040.0000 {
    uart0 @ 0x90000 {
      reg = <0x90000 0x1000>;
      clock = &uart_clk;
      parameter baudrate_divider {
        offset = 0x08;
        bits = 15:7;
      }
      operating-state suspended {
        operating-point std {
          power = 5mW;
          action = clock.disable();
        }
      }
      operating-state on {
        operating-point std {
          power = 45mW;
          action = clock.enable();
        }
      }
    }
    spi @ 0x98000 {
      reg = <0x98000 0x1000>;
      clock = &spi_i2c_clk;
```

Signals: Clocks

```
system_clocks {
  clk_pll0 is &pll_clock {
    rate = 1800000000;          // "1800 MHz"
    reset = &pll0_rst;
    clock-bypass { bits = "0"; }
    behaviors { set_rate = ”setpllclk”; } }
  arm_clk is &mux_div_bypass_clock {
    clock-multiplexer { address=A9DP_CLKCTL }
    clock-divider { address=A9DP_CLK_DIVCTL }
    clock-bypass { address=A9DP_CLK_DIVCTL }
  }
  uart_clk is &gateable_clock {
    description = "UART";
    clock-gate { address = AXI_CLK_CNTRL;
                 bits = "6"; }
  }
  spi_i2c_clk is &gateable_clock {
```

Signals: Voltages

```
i2cbus @ 0xF0000 {
    pmic8803 @ 0x2c {
        voltage-controller;
        compatible = "adp,aps8803";
        parameter LDO1_ENABLE {
            address = 0x4; bits = "0";
            values = "enable, disable";
        }
```

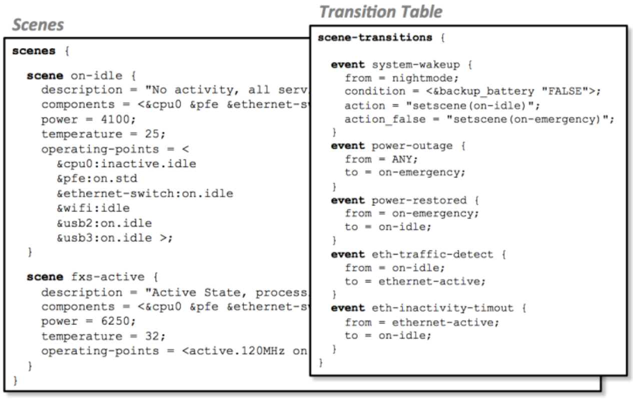
Fig. 1. UHA Components and Signals.

The software components and their performance requirements are described using UHA Tasks. This allows the resulting power management control software to be more aware of the work that is to be performed and to adjust the component power states accordingly.

Finally, the concept of UHA Scenes and Transitions shown on Fig. 2 is used to capture application scenarios and the optimum component configurations for the work to be performed in those scenarios, essentially formalizing the work requirements for the device and directly relating the work to the energy consumption. Scene transitions are initiated by events resulting from external impacts or from running software tasks.

The operation of a device is represented by a series of discrete Scenes which can occasionally overlap. Scenes specify the hardware components needed to deliver the work and are transitioned based on triggering events using state transition tables.

In this paper we have applied the new methodology of UHA and the tools to the broadband residential gateway.



Scenes

```
scenes {

  scene on-idle {
    description = "No activity, all serv
    components = <&cpu0 &pfe &ethernet-s
    power = 4100;
    temperature = 25;
    operating-points = <
      &cpu0:inactive.idle
      &pfe:on.std
      &ethernet-switch:on.idle
      &wifi:idle
      &usb2:on.idle
      &usb3:on.idle >;
  }

  scene fxs-active {
    description = "Active State, process
    components = <&cpu0 &pfe &ethernet-s
    power = 6250;
    temperature = 32;
    operating-points = <active.120MHz on
  }
}
```

Transition Table

```
scene-transitions {

  event system-wakeup {
    from = nightmode;
    condition = <&backup_battery "FALSE">;
    action = "setscene(on-idle)";
    action_false = "setscene(on-emergency)";
  }
  event power-outage {
    from = ANY;
    to = on-emergency;
  }
  event power-restored {
    from = on-emergency;
    to = on-idle;
  }
  event eth-traffic-detect {
    from = on-idle;
    to = ethernet-active;
  }
  event eth-inactivity-timout {
    from = ethernet-active;
    to = on-idle;
  }
}
```

Fig. 2. UHA Scenes and Transitions.

IV. The EU Code of Conduct

Information and communication technology (ICT) equipment is increasingly responsible for a significant portion of the total energy consumption. Broadband equipment accounts for around 15% of the ICT's overall energy consumption and is predicted to equal 50 TWh annually by 2015 [6]. Consequently, standardization and regulation activities have been initiated to curb the energy consumption of widely deployed communication devices. These activities promote close cooperation with broadband equipment vendors, who take on the task to voluntarily meet the standards and maximize equipments’ energy efficiency. In order to ensure consistency between different approaches in measuring energy efficiency, equipment vendors were given guidelines and methodology principles in the form of the EU Code of Conduct (EU-CoC) [6] and EU-CoC compliant Home Gateway Initiative requirements [7].

The EU-CoC specifies power consumption limits for different operating modes of a broadband device while providing specific services. For each type of service, such as USB data transfers and WAN downstream traffic, a set of hardware components cooperates to provide the service. Each of the components can be configured in various operating states, corresponding to different levels of capability and power consumption. The goal is to configure the operating state so that the performance requirements of the service can be met with the lowest possible power consumption. However, the EU-CoC does not specify implementation details, just the power constraints and response times to the service requests.

V. Applying UHA to Meet EU Power Guidelines for Residential Gateways

The operating modes of a residential gateway defined by EU-CoC are “active”, “idle” and “off”. In the idle mode, all components are in their individual idle states, not processing or transmitting significant amounts of data, but able to detect activity and to switch to the active state automatically. The device’s active mode is defined as all components in active state, processing and transmitting data at a specific rate. Most of the time, provided services do not require all device components to be in their active state, therefore many device states are possible, not just idle or active. Additionally, we introduced a “sleep” mode which is similar to “off”, but does not require manual wakeup and reboot. Component states are determined by inter-component state dependencies, maximum allowed response time and resource requirements, all of which depend on services the device should provide. We are using UHA and the formal concept of Scenes to describe these requirements and dependencies in a formal manner.

An example of a Scene for a residential gateway can be presented as follows. During the periods while the user is not at home, the device should be in energy-saving mode. The Wi-Fi access point should only send beacons in order to check for the presence of Wi-Fi clients. Therefore, the Wi-Fi access point is configured to the beacon-only operating state, in which the energy consumption scales with useful work done: short periodic, instead of continuous activity. The scene which describes user absence can be expanded with all services which the user cannot request outside of home, such as LAN access. In general, a Scene consists of components in their optimal operating state for the given use case.

Our target was the Mindspeed Technologies EVM evaluation board with the Comcerto 2200 (C2200) SoC [8]. The C2200 relies on a dual ARM Cortex-A9 processor in its CPU subsystem which is running the OpenWRT, a Linux distribution optimized for networking applications. A simplified block diagram of the system is presented on Fig. 3 (not all components are shown).

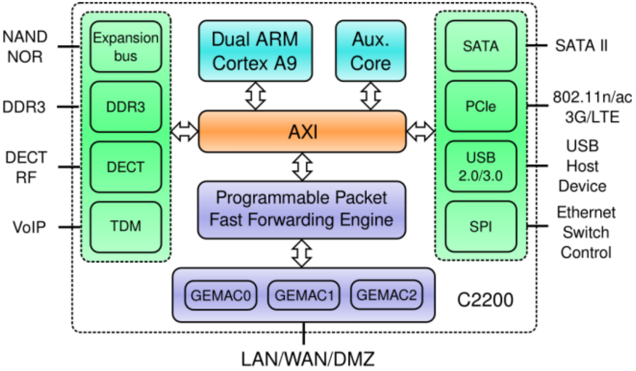


Fig. 3. Mindspeed Comcerto 2200 block diagram.

The C2200 contains an auxiliary core executing the energy management software. This enables a wider range of energy proportionality as the power hungry Cortex-A9 and its subsystem are only woken up and active when necessary. Similar solutions are deployed in SoCs for other applications, in particular in the mobile space [9].

VI. Measurement Setup

The measurement setup used to verify the results consists of the Mindspeed Technologies EVM evaluation board with the C2200, a high precision data acquisition (DAQ) unit and a PC. To obtain fine grained power consumption data, each voltage rail on the EVM board is instrumented with high accuracy 50 mΩ shunt resistors with low temperature drift, allowing us to measure not only overall power, but also the power for each of the six different voltage domains of the device.

The DAQ is used to measure and log the voltage drop over the shunts. We used the National Instruments NI-DAQ USB-6289 and the Agilent U2500, although a wide variety of alternative high precision DAQs can be used as well. Both of the selected units support simultaneous sampling of multiple analog inputs and are capable of sending sampled values via USB to the PC for post-processing and visualization.

To coordinate the measurements with the events in the hardware, the digital input of the data acquisition unit is connected to a GPIO pin of the board. The auxiliary core generates synchronization signals by toggling the output value on that GPIO pin when a transition in hardware is initiated. GPIO output values are sampled on the digital input of the DAQ. The approach found in [10] is using the serial connection for the same purpose.

VII. Measurement Results and Analysis

In order to evaluate the effectiveness of the newly implemented methods, the power consumption is measured in the active, idle and sleep modes. Fig. 4 shows the measured power graphs for each voltage rail of the board with the Energy Proportional Management (EPM) enabled, as well as the total power consumption measured at the DC input of the board with and without EPM enabled. In Table 1, the resulting power values refer to the sum of the average power consumptions only on the voltage rails of the EVM board and the power savings percentages are presented in the last column.

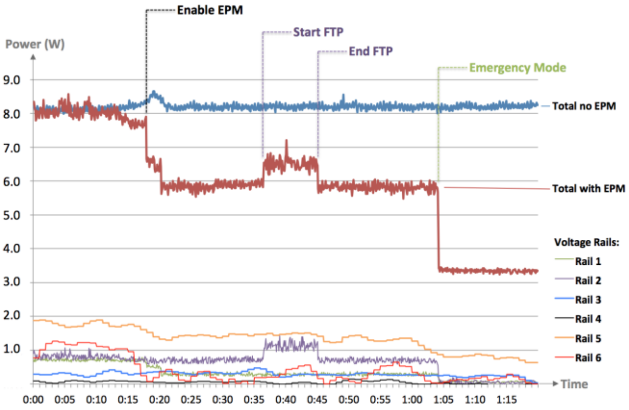


Fig. 4. Measured power with and without EPM.

To measure the active state power consumption, a PC is connected to an Ethernet LAN port of the residential gateway and Ethernet traffic is generated by transferring files using file transfer protocol from the OpenWRT file system to the PC. Starting a transfer through the LAN port causes Energy Efficient Ethernet (EEE) to automatically switch from Low-Power Idle (LPI) to active state. Power consumption is measured during the file transfer, while

CPU, DDR, AXI were on. Other components are either not used at the time and are turned off, or have no significant influence on the power consumption.

TABLE 1: BOARD RAIL POWER SAVINGS IN DIFFERENT MODES.

Mode	Component State				Power [W]	Saving [%]
	CPU	EEE	DDR	AXI		
Active	On	Active	On	On	3.482	31
Idle	Idle	LPI	On	LP	2.985	38
Sleep	Off	Off	SR	LP	1.385	71

The idle mode refers to the configuration when there is no traffic or processing, and all components are in their individual idle or on states. To save more energy during idle, the AXI bus frequency is lowered accordingly. By contrast to idle, in the sleep state most components are in their off state, except the wake-up logic and the DDR memory, configured to operate in self-refresh (SR) mode in order to save the memory context.

The sleep mode brings the largest power savings and can be leveraged in multiple scenarios. In the case of a power outage the device will be powered by its backup battery, so it is important to maximize the operating time of the device. While battery powered, the device performs a limited set of functions, such as periodic communication with sensors, reaction to an external event and sending messages to the base station. Therefore, the sleep mode is the most appropriate to be used: the device remains in sleep mode most of the time in order to preserve available energy and wakes up only to perform critical functions. The sleep mode can also be applied at night time, while the functionalities of the residential gateway are not used.

In summary, under the EPM control the device was able to meet the EU-CoC power levels and latency requirements with a reasonable margin. Assuming uniform mode usage and 85% conversion efficiency [6], the energy savings based on the expected 2014 worldwide residential gateway production [11] are equivalent to the half of the output of an average (500 MW) coal-fired power plant [12].

The generated EPM software consumed around 5% of the auxiliary core's cycles allowing the EPM to run simultaneously with other applications. Alternatively, a less powerful core can be used. The code image size was 32-48 kB, depending on the configuration, and 24 kB was used for the data. Further optimizations of the code size are possible, but were not required in this case.

VIII. RELATED WORK

In the past, numerous research activities have been conducted to optimize energy consumption using dynamic power management [13], [14], [15]. However, the majority of the research in recent years was based on improving energy efficiency through implementation and selection of different policies [16], [17] using operating system (OS) based power management (PM), such as the ACPI [18] and Linux PM [19], [20], [21].

To achieve energy-proportionality, the latest generations of SoCs have heterogeneous multiprocessor architectures [8], [9], [22] with a hierarchy of processors in terms of capabilities, power consumption and purpose. Similar to [21], we have used the auxiliary core of the residential gateway platform to execute the EPM software.

The EPM presented here is OS agnostic. As in [23], it can coordinate power management of multiple OSs running simultaneously, but contrary to [23], does not require the power-hungry and latency-intensive main processors to stay active. Also, we can selectively manage the energy of all on-chip and on-board components not only the basic on and off states. In this context our solution can be also seen as a system level hypervisor running on the multiprocessor system with main and auxiliary cores.

Because of their widespread use and high total energy consumption, the broadband devices and in particular residential gateways are a topic of strong interest in the energy community and were extensively analyzed in [11] and [12]. To the best of our knowledge, no systematic methodology solution to the problem similar to ours has been deployed or published so far.

REFERENCES

[1] L. Barroso, and U. Hözle, "The Case for Energy-Proportional Computing," *IEEE Comp. Soc. Press*, vol. 40, issue 12, Dec. 2007.

[2] S. Gunther et al., "Energy-Efficient Computing: Power Management System on the Nehalem Family of Processors," *Intel Technology Journal*, vol. 14, issue 3, 2010.

[3] Texas Instruments, "AM335x ARM Cortex-A8 Microprocessors (MPUs)," *Technical Reference Manual*, April 2013.

[4] P. Greenhalgh, "Big.LITTLE Processing with ARM Cortex-A15 & Cortex-A7," *ARM White Paper,* Sept. 2011.

[5] AGGIOS, *http://www.aggios.com/technology.htm*

[6] European Commission, "Code of Conduct on Energy Consumption of Broadband Equipment," version 4.1, Jan. 2013. *http://iet.jrc.ec.europa.eu/energyefficiency/sites/energyefficiency/files/code_of_conduct_broadband_equipment_v4_1_final.pdf*

[7] Home Gateway Initiative, "Requirements for an Energy Efficient Home Gateway," *HGI Publications*, Oct. 2010.

[8] Mindspeed, "M86xxx Comcerto 2000 Family Application and Content Aware Communication Processors," *www.mindspeeed.com*

[9] NVIDIA, "Variable SMP – A Multi-Core CPU Architecture for Low Power and High Performance," *Whitepaper*, 2011.

[10] A. Carroll, and G. Heiser, "An Analysis of Power Consumption in a Smartphone," *Proceedings of the USENIXATC '10*, 2010.

[11] B. Sappington, "Residential Gateway Trends: Bringing Value Home," *Park Associates Whitepaper*, March 2013.

[12] N. Horowitz, "Better Viewing, Lower Energy Bills, and Less Pollution: Improving the Efficiency of Television Set-Top Boxes," *Natural Resources Defense Council*, June 2011.

[13] L. Benini, A. Bogliolo, and G. Micheli, "A Survey of Design Techniques for System-Level Dynamic Power Management," *IEEE Transactions on VLSI Systems*, vol. 8, no. 3, June 2000.

[14] B. Brock, and K. Rajamani, "Dynamic Power Management for Embedded Systems," *Proceedings of the IEEE Int'l SOCC*, 2003.

[15] IBM, and MontaVista Software, "Dynamic Power Management for Embedded Systems," Nov. 2002.

[16] Y. H. Lu, and G. D. Micheli, "Comparing System-Level Power Management Policies," *IEEE Design & Test of Computers*, vol. 18, issue 2, March 2001.

[17] N. Pettis, and Y. H. Lu, "A Homogeneous Architecture for Power Policy Integration in Operating Systems," *IEEE Transaction on Computers*, vol. 58, no. 7, July 2009.

[18] Advanced Configuration and Power Interface, rev. 5.0, Dec. 2011. *http://www.acpi.info/spec50.htm*

[19] V. Pallipadi, and A. Belay, "Cpuidle – Do Nothing Efficiently," *Proceedings of the Linux Symposium*, vol. 2, June 2007.

[20] L. Brown, and R. Wysocki, "Suspend-to-RAM in Linux," *Proceedings of Linux Symposium*, vol. 2, July 2007.

[21] R. Muralidhar et al., "Experiences with Power Management Enabling on the Intel Medfield Phone," *Proceedings of the Linux Symposium*, July 2012.

[22] R. Kumar et al., "Heterogeneous Chip Multiprocessors," *IEEE Computer Society Press*, vol. 38, issue 11, Nov. 2005.

[23] ARM, "Power State Coordination Interface," *Whitepaper*, Aug. 2013.