

Scopus - Document details

https://www.scopus.com/record/display.url?eid=2-s2.0-85059985644&origin=resultslist&sort=pdf-f&src=s&st1=Yakimov&st2=Peter&no=1&nlr=20&nls=alprfrm-t&aff=

Search Sources Lists

Create account Sign in

Document details

[Back to results](#) | [Previous](#) 5 of 15 [Next](#) | [Export](#) [Download](#) [Print](#) [E-mail](#) [Save to PDF](#) [Add to List](#) [More...](#) | [View at Publisher](#)

2018 IEEE 27th International Scientific Conference Electronics, ET 2018 - Proceedings
 27 November 2018, Article number 8549674
 27th IEEE International Scientific Conference Electronics, ET 2018; Szopopol; Bulgaria; 13 September 2018 through 15 September 2018; Category numberCFP18H39-ART; Code 143063

DAEDALED: A GUI Tool for the Optimization of Smart City LED Street lighting Networks (Conference Paper)

Bogdanov, L.^a, Polstra, S.^b, Yakimov, P.^c, Marinov, M.^c

^aDepartment of Electronics, Faculty of Electronic Engineering and Technologies, Technical University of Sofia, 8 Kliment Ohridski Blvd., Sofia, 1000, Bulgaria
^bFaculty of Science, Informatics Institute, University of Amsterdam, Netherlands
^cEuropean Polytechnical University, Bulgaria

Abstract
 This paper describes an experiment that uses software tools for Multi-Processor Systems-On-Chip (MPSoC) design to create networks of smart street lights. We first discuss the main features of the DAEDALUS design flow and the optimization framework SESAME. Then we present a graphical user interface (GUI) application that converts the street light network design parameters into a format suitable for the DAEDALUS framework. Finally we outline the benefits of this approach to designing smart street light networks. © 2018 IEEE.

SciVal Topic Prominence
 Topic: Energy resources | Scheduling | Optimal scheduling
 Prominence percentile: 85.343

Author keywords
 DAEDALUS design flow | GUI | Smart City | Street lights

Metrics [View all metrics >](#)
 PlumX Metrics
 Usage, Captures, Mentions, Social Media and Citations beyond Scopus.

Cited by 0 documents
 Inform me when this document is cited in Scopus:
[Set citation alert >](#) [Set citation feed >](#)

Related documents
 Smart street lighting system: A platform for innovative smart city applications and a new frontier for cyber-security
 Jin, D., Hannon, C., Li, Z. (2016) *Electricity Journal*
 A Microgrid Architecture for Integrating EV Charging System and Public Street Lighting
 Bruno, S., Giannocaro, G., La Scala, M. (2019) *Proceedings - 2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe*.

19:23 ENG 4.12.2019 r.

DAEDALUS: A GUI Tool for the Optimization of Smart City LED Street lighting Networks

Lubomir Bogdanov¹, Simon Polstra², Peter Yakimov¹ and Marin Marinov³

¹Department of Electronics, Faculty of Electronic Engineering and Technologies, Technical University of Sofia
8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria, {lbogdanov, pij}@tu-sofia.bg

²Faculty of Science, Informatics Institute, University of Amsterdam, the Netherlands, s.polstra@uva.nl

³European Polytechnical University, Bulgaria, mmarinov@ecad.tu-sofia.bg

Abstract – This paper describes an experiment that uses software tools for Multi-Processor Systems-On-Chip (MPSoC) design to create networks of smart street lights.

We first discuss the main features of the DAEDALUS design flow and the optimization framework SESAME. Then we present a graphical user interface (GUI) application that converts the street light network design parameters into a format suitable for the DAEDALUS framework. Finally we outline the benefits of this approach to designing smart street light networks.

Keywords – Street lights, DAEDALUS design flow, Smart City, GUI

I. INTRODUCTION

According to Directive 2005/32/EC of the European Parliament and the Council of the European Union from 06.07.2005, the energy efficiency is of great importance for evaluating the operation of every system. The accepted goal by the European Commission in 2007 is the strategy 20-20-20 which aims for a 20% reduction of energy consumption by 2020 (20% increase of energy efficiency, 20% reduction of CO₂ emissions, and 20% renewables by 2020). However in order to ensure good living conditions the street lighting is developed quickly and has become a large energy consumer. To respond to the above-mentioned initiative the introduction of energy saving solutions is obligatory. LEDs that produce the same luminous flux as the traditional fixtures while consuming two times less energy have the advantage over the other light sources and are suitable for use in the street lights. Their full electronic control allows for the introduction of smart technologies and networking. The benefits of the networked LED street lights include: up to 60% reduction of energy and operating costs; about 20 years long life; possibility for dimming during low traffic situations; flexible on/off scheduling in connection with events; remote failure detection; measuring the energy consumption [1].

Because of the precise positioning of the street lights and their ubiquitous spreading, they can operate as nodes of a complex networking infrastructure, e.g. for video surveillance or for communications.

Smart street lights can contain multiple microprocessors and memory components and therefore the street light network in a city is comparable to a distributed embedded system. And here a typical problem of an embedded system arises – it is difficult to achieve a perfect symmetry between the hardware platform and the software applications. This means that finding an optimal mapping of software processes to hardware components is of a great concern.

The goal of the presented research is to integrate embedded system development tools in deployment of smart street light networks in order to achieve savings in terms of hardware (which means less cost) or software execution time (which means better application performance).

This paper is organized as follows: section 2 presents the essentials of the DAEDALUS design flow. The development of the graphical user interface (GUI) application “Daedalus Design Studio” for optimization of smart LED street light networks is considered in section 3. The developed GUI application and its interface are discussed in section 4. The conclusions and the future work are presented in the last section.

II. DAEDALUS TOOL DESIGN FLOW

For this project we selected the DAEDALUS framework developed by Leiden University, the Netherlands (<http://daedalus.liacs.nl>) for Multi-Processor Systems-On-Chip (MPSoC) design [2]. The whole MPSoC design process includes several complex steps (e.g., application specification, synthesis, programming, and implementation). The methodology is based on the Platform-Based Design (PBD) concept, which is a promising new approach to tame the permanently growing complexity of the modern embedded systems.

The system-level MPSoC design process starts with the specification of the application that is to be executed on the MPSoC as a sequential program in C. A definite set of rules, the Daedalus rules for static affine nested loop programs, must be followed when writing the application in C. These rules ensure that the conversion of the sequential program into a parallel one can be done automatically later on using the PNgen tool from DAEDALUS [3]. Hence, there are no synchronization primitives that have to be used, which greatly reduces the development time. A file describing the data flow of the parallel program is generated and it is referred to as the Kahn Process Network (KPN) parallel application specification. In the next step untimed and timed software simulations of the parallel application specification are performed to verify the final result. The observed behavior of the parallel program has to be the same as that of the initial sequential C program. If the parallel application specification is correct, a high-level block diagram of the MPSoC is created using the DAEDALUS graphical user interface (GUI). After generating the block diagram, also referred to as the platform specification of the MPSoC, an application mapping needs to be made. A mapping is created by binding each process of the parallel application

specification to a microprocessor (or an IP block) of the platform specification. Again, this is easy to do using the DAEDALUS GUI. After the generation of the application, platform and mapping specifications, the ESPAM tool (Embedded System-level Platform synthesis and Application Mapping) accepts these as inputs and automatically generates the firmware and the hardware specification of the MPSoC system. Then the commercial Xilinx tools are used to produce the final low-level synthesis of the netlist and/or to generate a bitstream file in order to implement an MPSoC system prototype, which can be downloaded and run on an FPGA-based prototyping board [4]. The DAEDALUS design flow includes an optimization framework called the SESAME Design Space Exploration framework for finding optimal mappings in embedded systems (Fig. 1).

III. GUI DEVELOPMENT

A graphical user interface (GUI) application “Daedalus Design Studio” that aims at optimizing smart LED street light networks in cities has been developed. In theory the SESAME tools could be used directly. In practice however this is a daunting task due to the high number of parameters that have to be entered. Such parameters include the number of processes, their names, their C++ application models,

hardware components, input/output ports, component costs, execution times, default mappings, etc. If this is done for one, or even ten nodes, it is doable within an hour with the help of the SESAME graphical editor. The problem is that a city consists of thousands of LED street lamps, intermediate components (called Concentrators) and possibly multiple main stations. The development time would increase drastically. There is a missing link between SESAME and this practical problem, and the goal of this research project is to automate the generation of the input for the SESAME framework. This means that an application with a simple GUI must be developed that requires the user to input only the important parameters of the city network and automatically fill in the rest of them with sensible default values. The proposed tool flow is shown in Fig. 2.

The program has been implemented and its internal structure is shown in Fig. 3. Using a graphical interface, the user inputs basic information about the LED street lamps, the intermediate devices and the main stations. Afterwards the process is launched. This invokes four basic methods of the program that: generate a project template, generate application XML (with its corresponding C++ code), generate XML of the platform (that is actually a model of the city LED network) and generate a default XML mapping. The top-level directory contains the application and architecture directories.

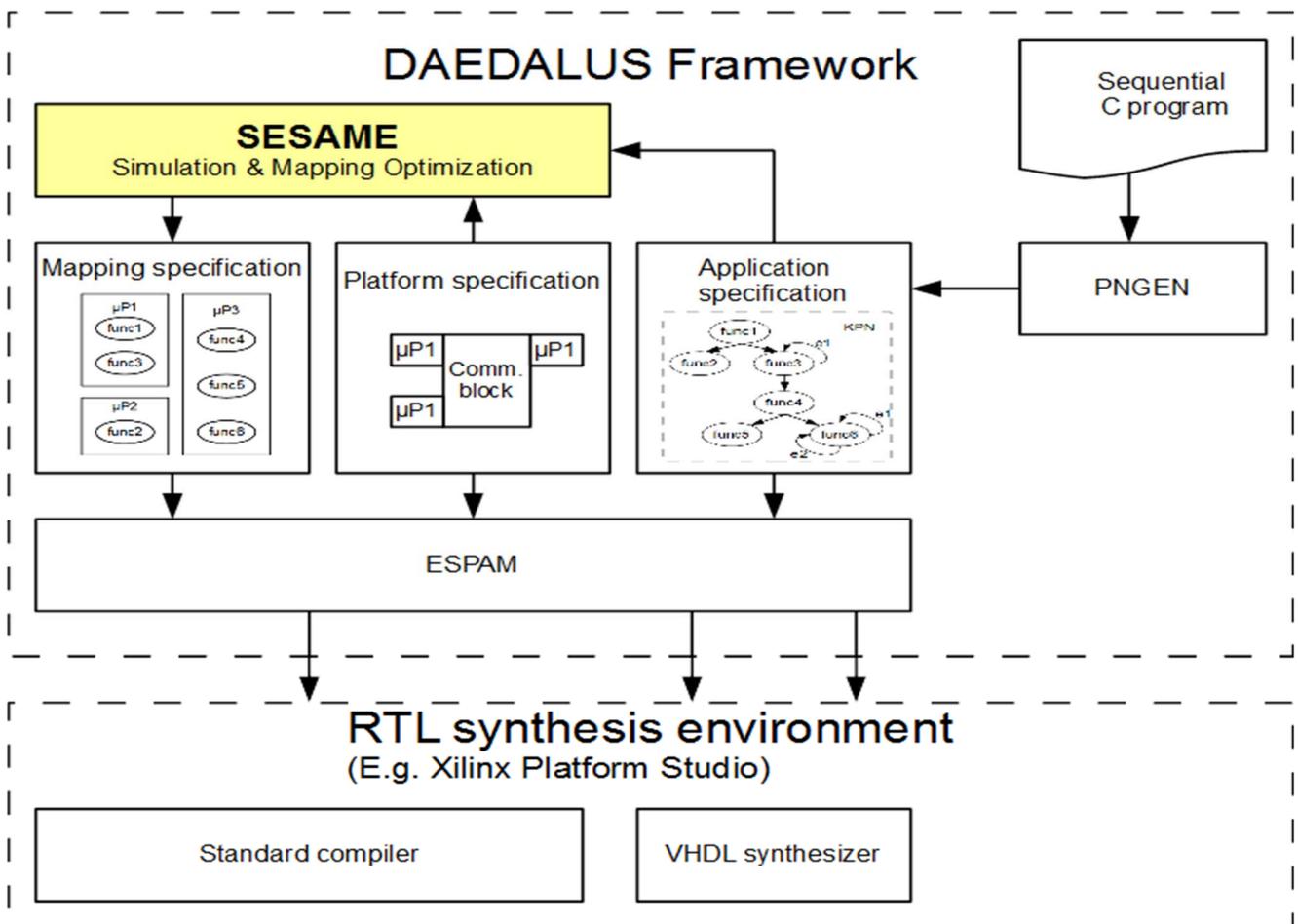


Fig. 1. DAEDALUS high-level synthesis and optimization tools

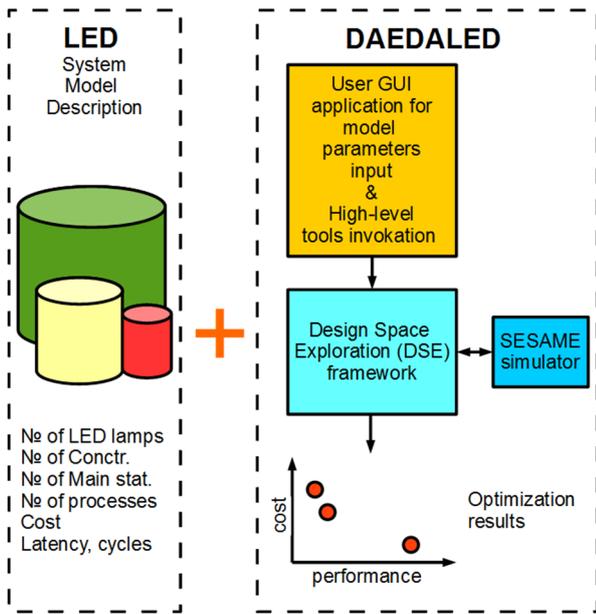


Fig. 2. Combining existing software and smart LED models to form the DAEDALED concept

The application directory includes an XML description of a Kahn Process Network (KPN) that reflects the city structure. Each street lamp and concentrator have a single KPN node assigned to it. The main station however has more than one node that map to it. This reflects the user applications that can be executed on different main stations and possibly on concentrators. Links between nodes are substituted with FIFO buffers in the architecture layer which is required for simulating the system. In reality no such devices exist. The links between devices are simple sets of wires. This is the reason the latency of FIFO buffers is set to 1 cycle by default. All the generated files are gathered in a directory that is structured to conform to the input requirements of the SESAME framework. With this structure, the project can be processed with all the existing SESAME tools. One of

them is the Design Space Exploration framework that is an extensible set of programs to optimize the mapping of applications to hardware. Starting with a population of randomly initialized mappings it uses a genetic algorithm to evolve these mappings according to different objectives. Parameters such as the genetic operators and the selection algorithms are all configurable. The SESAME simulation is used to evaluate the fitness of each new mapping. When the optimization process terminates, the Pareto front of the best mappings according to the chosen objectives is plotted in a chart and a summary of the results is displayed in the console. These Pareto optimal mappings in terms of the chosen objectives such as cost, power usage and execution time are shown allowing the designer to optimize the network before deploying it to the customer. The designer needs to perform a small set of experiments in order to model each type of device correctly – information about the execution time (referred to as “latency” in the tool) and data token generation rate are needed for the LED smart street lamp, concentrator and main station. With the help of the proposed software tool, the designer will benefit in terms of LED network components cost and in terms of reduced application execution time.

IV. GUI APPLICATION

The data is entered using the main window shown in Fig. 4. The smart city LED street lights network may be divided into four components: LED lamps, intermediate devices (concentrators), main stations and user applications running on the processing elements in the network. Note that those applications are not the firmware of the controllers. This software is the front-end that displays and processes the gathered data. Usually there will be only one main station that gathers all the data from the LED network but for flexibility this parameter is also editable with the help of the Quantity field.

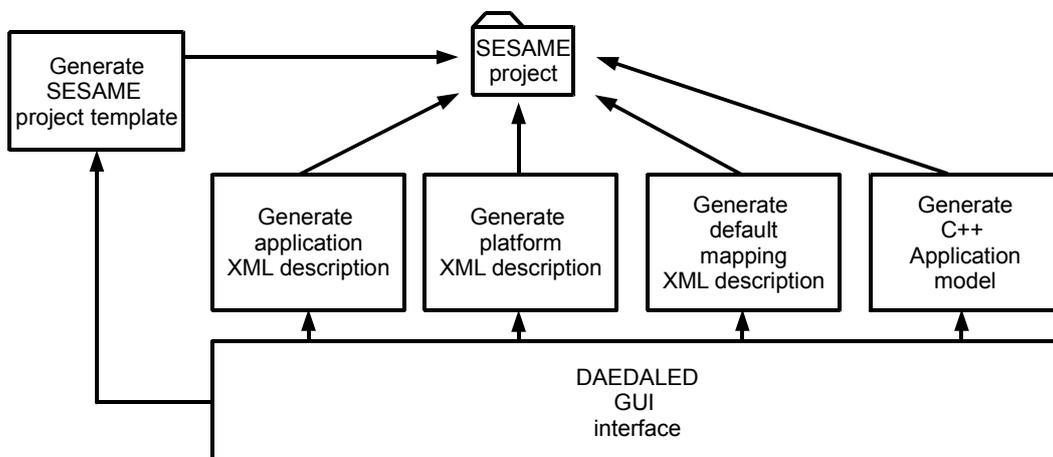


Fig. 3. DaedaLED Design Center – a tool for optimizing mapping of applications in a smart LED street light network

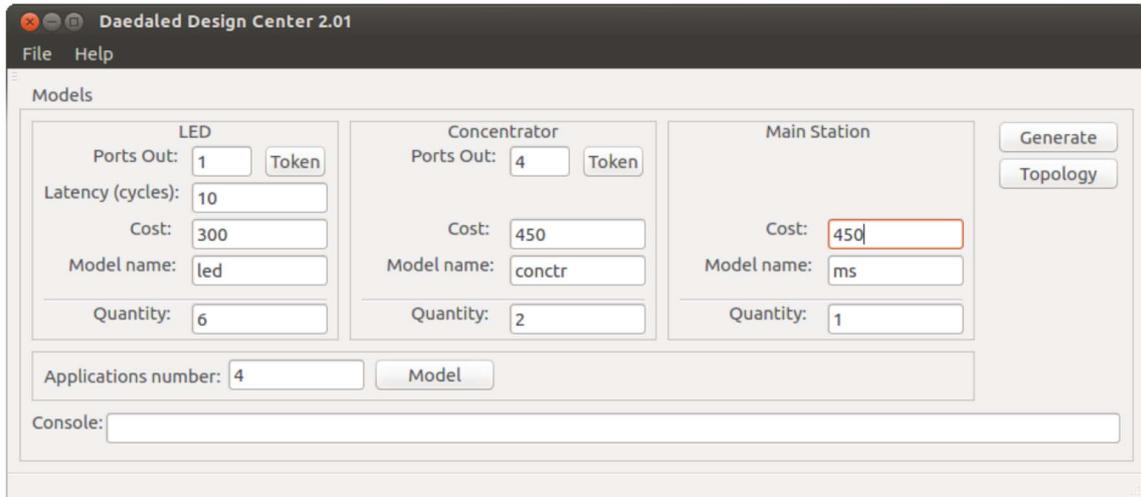


Fig. 4. DAEDALED Design Studio v.2

The required parameters for the LED, CONCENTRATOR and MAIN STATION are:

Ports Out – number of output ports that connect the entity to the surrounding environment.

Latency – the computational load to execute a single iteration of a Kahn process expressed in processor cycles (Kahn tasks are represented in the Kahn Process Network with an ellipse).

Quantity – number of devices in the city or neighborhood.

Cost – the cost of the entity. May be real currency, or may be in relative units. Due to SESAME restrictions, cost can only be expressed as an integer value.

Model name – example name of the entity/device controller. Can be any string.

The MAIN STATION contains an additional field:

Applications number – the number of user applications that have to be mapped to processing elements in the network. Examples of these front-end applications include statistical or graphical representation of data, LED faults and LED lamp current operating modes.

The DSE optimization framework uses a user selectable genetic algorithm and generates a Pareto front of optimal mappings between the KPN and the architecture. The best mapping for the given requirements has to be chosen – either a solution with a lower cost or one with a higher throughput. To aid the developer a graph of the results will also be displayed like the one in Fig. 5. A Pareto front of three points is shown there. The X coordinate indicates system cycles, the Y coordinate is the system cost. In this example we see that the Pareto front at $Y = 800$ is the most balanced solution in terms of cost/performance.

V. CONCLUSION

The paper discusses the implementation of a GUI tool for city street lighting optimization. The lighting is modeled with processing elements for the smart LED lights and a tree connection between the nodes. The optimization of the mapping between the software processes and current hardware configuration (i.e. lamp connections) is left to a high-level mapping optimization framework called SESAME. The GUI tool automates the

process of project creation for SESAME and speeds up development time.

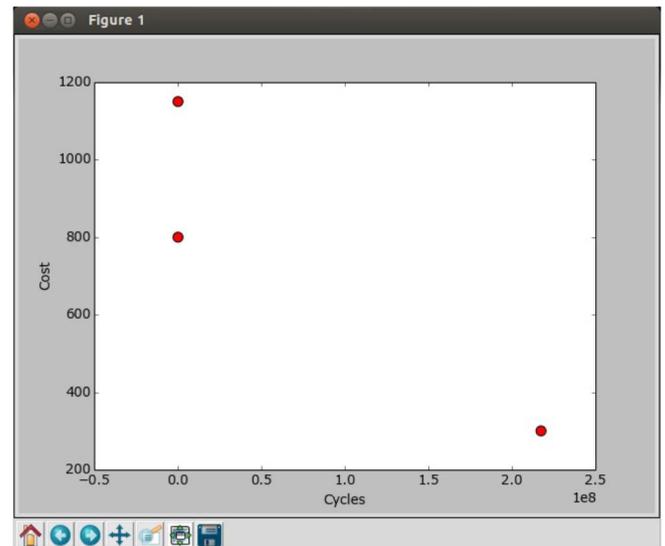


Fig. 5. Optimization results shown in a graph

ACKNOWLEDGEMENTS

The research and the paper preparation are supported by Grant agreement No. FP7-ICT-609491 – TETRACOM Project “Technology Transfer in Computing Systems”.

REFERENCES

- [1] M. Shahidehpour, C. Bartucci, N. Patel, T. Hulsebousch. P. Burgess and N. Buch. *Streetlights Are Getting Smarter*, IEEE Power and energy magazine, vol. 23, No. 3, May/June 2015, pp. 67-80.
- [2] <http://daedalus.liacs.nl>.
- [3] T. Stefanov, E. Deprettere, H. Nikolov, M. Marinov, A. Popov. *Embedded Systems: components, modelling, design and case study*, Technical University – Sofia, 2012.
- [4] L. Bogdanov, H. Nikolov, T. Stefanov. *Embedded Multi-Processor Systems-On-Chip: Laboratory Experiments and Users Manual*, Technical University - Sofia, 2013, ISBN: 978-619-167-034-5.