

ASIC Design Project Management Supported by Multi Agent Simulation

Jana Blaschke, Christian Sebeke, Wolfgang Rosenstiel

Abstract The complexity of Application Specific Integrated Circuits (ASICs) is continuously increasing. Consequently chip-design becomes more and more challenging. To handle this complexity for a fast ASIC development, the existing design process has to become more efficient. To achieve this, we used an approach based on a multi-agent simulation. A multi-agent system is an intuitive way to represent a team of designers, creating an ASIC. Furthermore, MAS are capable of coping with the natural dynamics of the design process, reacting to and modelling unforeseen events. The resulting Model is capable of an extensive analysis of the design process. It can make reliable predictions on design project courses and identify weak spots within the design process. It can provide status-analysis of ongoing projects and suggestions on how to organize, plan and execute a new project efficiently.

1 Introduction

Because of smaller channel width and design automation in microelectronics, ASICs for electronic devices can realize more functionality and become more and more complex. If Moore's Law [5] persists, the required design effort for ICs exceeds the real possibilities. Therefore a way has to be found to make chip-design more efficient and plannable. This is a difficult task because design processes are complex, non-linear and depend on many human decisions. To make the design process more efficient it is necessary to make it assessable at first.

Existing approaches for design process modelling are static and generalizing. Numerics Management Systems [2] deals on a commercial basis with design process modelling. They provide a benchmarking service that evaluates the design process of completed projects. This allows only a retrospective analysis of terminated

Jana Blaschke
Robert Bosch GmbH, Tübinger Straße 123, 72762 Tübingen, Germany
e-mail: jana.blaschke@de.bosch.com

Please use the following format when citing this chapter:

Blaschke, J., Sebeke, C. and Rosenstiel, W., 2009, in IFIP International Federation for Information Processing, Volume 296; *Artificial Intelligence Applications and Innovations III*; Eds. Iliadis, L., Vlahavas, I., Bramer, M.; (Boston: Springer), pp. 87–93.

projects is enabled. This neither allows what-if analysis of projects nor a reasonable planning for investments and resources.

To overcome these disadvantages we developed a flexible approach that allows status-analysis and an efficient planning of project courses. The approach was developed within the context of a public enhanced research project called PRODUKTIV+. The objective of PRODUKTIV+ is the development of a comprehensible model and reference system to measure and assess the productivity and performance of design processes [7].

2 The Concept

The approach needs to be able to make predictions for real and suggestions for optimal project courses. It should give status analysis of ongoing and finished projects and has to be flexible enough to handle and compare heterogeneous designs.

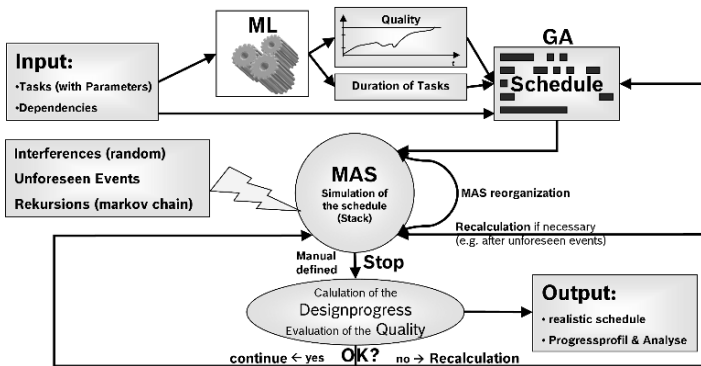


Fig. 1 Our dynamic approach to assess the design process in order to make it analysable and plannable. Inputs into the system are design tasks generated by a machine learning model. The core of the model consists of a multi agent simulation, processing the tasks. The simulation is organized by a scheduling algorithm. An interference module introduces dynamics to the system.

At first we generated a machine learning (ML) model of the design process. We split the design process into four subparts, the analogue and digital frontend and the analogue and digital backend. After detecting influential parameters, we analysed them regarding their impact on the generalization performance of the models. With the most relevant parameters we trained Neural Network, Gauss Process and Support Vector Machine models. None was precise enough to meet our requirements. The best model achieved was a support vector model for the analogue backend. It had a generalization error of 10% and a variance of 3.2. Another problem is that the models are static, neither considering the time course of the design process nor the

dynamics of human-dependent processes. However, the ML models can be used for modelling the average duration of basic design tasks.

To overcome the lack of the ML model we developed a time-dependent simulation that is able to handle the natural dynamics immanent to human-lead processes. The model consists of several interacting modules, creating a flexible, dynamic and accurate model of the design process. An overview of the system is given in Fig. 1.

A MA simulation builds the core of the system model. The tasks to be executed are basic design tasks, derived from historical data or generated by a ML model. The simulation is organized by an optimized schedule calculated by an integer linear program (ILP). Dynamics are introduced by an interference module, perturbing the smooth schedule execution. The schedule and the MAS organization are adapted during the simulation if their deviation exceeds a given limit. The tracked agent activity gives a realistic estimate of the design process duration and a suggestion how to organize it. Multiple design-simulation runs give a statistic of average runtimes.

3 The Multi Agent Systems Architecture

We use a Multi Agent System (MAS) to describe the design process. MAS accomplish time-dependent simulations of complex interactions within a group of agents. This is a very important characteristic for our purpose, because it allows an inspection and analysis of the design process at any point of time.

The structure of the MAS depicts the design process of ASICS. Therefore the agents, their interactions and organization has to describe a design team, the design environment and structure. Every agent runs in an independent thread. Basic design tasks that have to be accomplished during the design process, determining the structure of a simulation run. The duration and sector of the tasks as well as its dependencies to other tasks account for a large part of the simulation organisation.

We defined four different agent types. The *designer agent* resembles the human designer. At their initialization characteristics have to be defined that assign working areas and properties to the designer. A designer can access a tool-pool that it can use to accomplish tasks of a specified working sector. The *tool agent* specifies a design tool. Its properties denote the design tasks that can be executed with the tool. The management and alignment of tasks to agents is done by an *administration agent*. Dynamics are introduced to the simulation by an *interference module*. It accounts for a realistic simulation.

The simulation structure is determined by the tasks and the agent interactions. Designers can only execute a task belonging to their working area using an appropriate tool. A designer can accomplish only one task at a time. A tool license can only be used by one designer at a time. If a designer has snatched a free tool license, he sends a task-request to the administration agent. If an executable task matching the designer's capabilities exists, the administration agent will provide it. For implementation we used the agent platform A-globe [6].

4 Introducing Dynamics to the MAS: the Interference Module

To obtain a simulation that reflects reality as close as possible, dynamics are introduced to the MAS. This is accomplished by an interference module. We identified four main factors that have a strong influence on the course of the design process.

The first factor is the agent availability restricted by illness and holidays. If an agent is out of office, he is not able to take and process a task. If he is already working on a task, the task duration will be prolonged or the task has to be passed to another agent. The second factor is the occurrence of unforeseen events, say long term resource drop outs or the introduction of new workload. The recursion of tasks depicts a third factor. There can be intense recursions between front- and backend design. The recursion probabilities were extracted from historical data and are inserted into a Markov Chain. As the tasks that are passed from the ML models to the MAS have averaged durations, a fourth factor is the task duration deviation.

5 MAS Scheduling with ILP

The administration agent has to manage the design tasks and to assign them to the designers. A natural way of doing this is to use a task schedule. As the duration and efficiency of the whole design process strongly depends on the tasks arrangement, schedule optimization is very important. This problem is known as the precedence, resource constrained scheduling problem, which is NP-complete [3]. An efficient, heuristical way to address this problem is the formulation of a zero-one ILP [4].

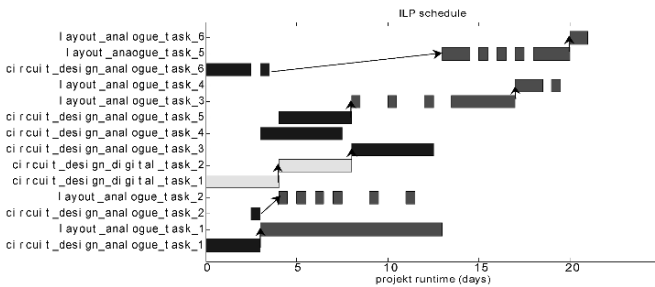


Fig. 2 ILP schedule of a real design subproject. It was optimized in terms of resource and time constraints and data dependencies between tasks. The constraints were formulized for three different regions, the analogue frontend (blue) and backend (red) and the digital frontend (yellow).

The objective of the optimization is to minimize the overall execution time of an ASIC-design project: $\min T_{design}$. The problem is represented by binary variables. x_i^j represents the decision to schedule (1) or not schedule (0) task i at time j . The variables form a $m \times n$ matrix X , representing a design task in every row and a point in time in every column. The last column containing a 1 denotes the overall design

execution time T_{design} . As the solution is restricted to some conditions, constraints have to be formulated to generate a valid schedule: The *must-schedule constraint* guarantees that every task is scheduled to one and only one point in time:

$$\sum_{j=1}^m x_i^j = 1, \quad (1)$$

To ensure that the number of r_{type} resources, here designers or tools, is not exceeded at any point in time, the *issue constraint* is formulated for every resource type:

$$\sum_{i=1}^n x_i^j \leq r_{type}, \quad (2)$$

The *precedence constraint* makes sure that dependencies between tasks are met, if a task k depends on a task i , i has to be scheduled before k :

$$\sum_{j=1}^m j * x_k^j + L_{ki} \leq \sum_{j=1}^m j * x_i^j, \quad (3)$$

where L_{ki} denotes the latency between x_k^j and x_i^j .

The *time constraint* ensures that deadlines $t_{deadline}$ are met:

$$\sum_{j=1}^m j * x_i^j \leq t_{deadline}. \quad (4)$$

The ILP was solved with the tool LPSolve [1].

The obtained schedules are used as input to the MAS.

6 First Results: Validation of MAS and Simulation of a Dynamic Design Process

We used part of a real ASIC-design project to evaluate our approach. The design task durations were averaged durations, gained from machine learning models. We initialized a MAS with one designer and one tool licence for digital the frontend, three designers and two tool licenses for the analogue frontend and 1.5 analogue layouter (one is part-time) and two tool licenses. We created a schedule by solving the ILP for these tasks.

At first we switched the interference module off and defined very tight degrees of freedom to obtain a validation of the MAS. Our expectations, that the MAS executes exactly the schedule were met. The ILP schedule, see figure 2, and the result of the simulation run, shown in figure 3(a), exhibit exactly the same project course.

The validation of the MAS was only a sanity check. To get a realistic simulation the interference module was switched on. Several runs gave an estimation for the

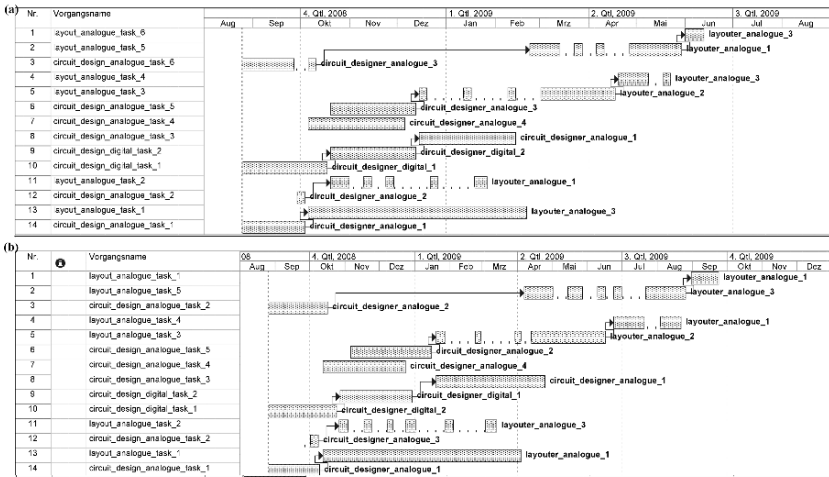


Fig. 3 (a) Simulation of the schedule provided by the ILP. The interference module is turned off. This run gives a simulation of the MAS. (b) Simulation of the schedule provided by the ILP. The interference module is switched on.

average runtime and the deviation of the project. Figure 3(b) shows an average run. The deviation of the runtime is 5 weeks . The simulation results reflect the real course of the project. They give a good picture of reality and allow an analysis of the project as well as suggestions on how to arrange the tasks in a good manner.

7 Conclusions and Future Work

The results of our simulation are quite satisfying. The MAS introduces a good possibility to simulate the complex and dynamic ASIC-design process and to make it assessible. A realistic simulation is achieved by the introduction of an interference module. To guide and improve the simulation, a schedule optimization is accomplished that offers ideal task-arrangements and resource allocations.

Up to 20 tasks (average 5 weeks duration for one task) can be handled efficiently by the ILP. Because of its ploynomial runtime, bigger problems have very long runtimes. Therefore we are working on a scheduling optimization based on genetic algorithms to overcome these limitations.

References

- [1] <http://sourceforge.net/projects/lpsolve>.
- [2] <http://www.numetrics.com/homepage.jsp>.

- [3] M. R. Garey and D. S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness*, Series of Books in the Mathematical Sciences. W. H. Freeman, January 1979.
- [4] D. Kastner, M. Langenbach, and Fb Informatik. Integer linear programming vs. graph-based methods in code generation. Technical report, 1998.
- [5] G. E. Moore. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86(1):8285, 1998.
- [6] David Šišlák, Milan Rollo, and Michal Pěchouček. A-globe: Agent platform with inaccessibility and mobility support. In Matthias Klusch, Sascha Ossowski, Vipul Kashyap, and Rainer Unland, editors, *Cooperative Information Agents VIII*, volume 3191 of Lecture Notes in Computer Science, pages 199–214. Springer, 2004.
- [7] J. Alt; A. Voerg. Produktiv+: Referenzsystem zur messung der produktivität beim entwurf nanoelektronischer systeme. *Newsletter edacentrum*, 1, 2006.