ORGANIZING AND PLANNING THE ASIC DESIGN PROCESS BY MEANS OF A MULTI-AGENT SYSTEM

Jana Blaschke[1], Christian Sebeke[1]
[1]*Robert Bosch GmbH, Tübinger Straße 123, 72762 Reutlingen, Germany*

Wolfgang Rosenstiel
Department of Computer Science, University of Tübingen, Sand 13, B 207, 72076 Tübingen, Germany

Keywords: Multi agent simulation, Global scheduling, Local scheduling, Task planning, ASIC design.

Abstract: Because of constantly improving technologies, the complexity of Integrated Circuits (ICs) is continuously increasing. Consequently IC design becomes continously more challenging and complex. A huge number of different possible design flows exists, delimited by different constraints. The design flow dynamically changes as recursions between design tasks occur. An approach that allows a fast and efficient ASIC design and that can deal with this huge complexity and dynamics is needed. Therefore we propose a methodology based on a multi-agent simulation combined with global and local scheduling techniques to construct a time-dependent, detailed model of the ASIC design process, which permits an extensive analysis and efficient organization.

1 INTRODUCTION

Smaller channel width in microelectronics together with design automation enables a realization of very complex ICs for electronic devices, comprising a vast number of functionalities (Moore, 1998). As several recursion steps occur between the frontend and backend design, the whole design process is extremely dynamic and difficult to predict. Another degree of complexity is introduced by different methods to realize the design steps. Altogether the design process of ASICs is a highly complex, dynamic and non-linear procedure, depending on human decisions.

If Moore's Law persists, the required design effort for ICs exceeds realistic imaginations. Limiting factors restricting the ASIC design are e.g. CPU power, human capability to work faster or tool runtimes. An approach that allows an analysis and efficient planning of ASIC design projects is needed to realize more complex projects. As Multi-agent systems (MAS) inherit properties that are very suitable for modelling and simulating highly complex problems involving human decisions in domains within a dynamically changing environment, we developed a methodology based on MA simulation combined with scheduling mechanisms. The simulation is organized in two distinct phases: a long term planning based on a Genetic Algorithm (GA) optimization for a high-level plan, and a short term planning to react dynamically to events. By this means we are able to develop optimized but realistic project courses of ASIC design projects for an efficient project organization. Additionally it allows a what-if analysis and comparison of different design flows with different design tools.

2 PREVIOUS WORK

MAS are widely used for simulation of human networks, organizing themselves to reach a defined goal (Schurr et al., 2005).
Scheduling of ASIC design projects is an issue for a long time (Easley et al., 1989). But due to a quickly changing design environment, design automation and new technologies ASIC design project planning requirements are changing constantly, old approaches become obsolete. Recent research in Design Project Scheduling focuses on selected design steps, as for example the high level synthesis of net lists (Lin and Kim, 2006), instead of scheduling the whole project. Models of aother research groups are based on derived metrics (Leppelt et al., 2006). But this neither allows what-if analysis of projects nor a reasonable planning for investments and resources.

Blaschke J., Sebeke C. and Rosenstiel W. (2010).
ORGANIZING AND PLANNING THE ASIC DESIGN PROCESS BY MEANS OF A MULTI-AGENT SYSTEM.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Artificial Intelligence*, pages 459-463
DOI: 10.5220/0002698704590463

3 APPROACH

Our approach has to enable the analysis and optimal organization of the ASIC design process and has to provide an efficient planning of ASIC project courses considering a lot of prerequisites. First, a simulation of finished projects for evaluation purposes and comparison to other projects needs to be enabled. Second, new and ongoing projects need to be planned in consideration of budget, resources, deadlines, alternative design styles and the task organization.

As we have to deal with a highly complex problem and a dynamic environment, MAS are an adequate way of simulating a group of humans executing design tasks. So a MAS simulation is chosen as the core of the system model. The ASIC-MAS architecture is shown in fig. 1. For task execution we need two types of ressources agents, a *designer agent* and the *design tool agent*. These agents belong, according to their expertise and functions, to different design areas. A high-level plan for task execution and design organization is provided by a planning agent, also responsible for the resource allocation of the tasks. Dynamics are introduced to our system by an interference module modelling recursive processes and unforeseen events, perturbing the smooth schedule execution and modelling dynamics inherent in processes lead by humans. A monitoring agent is tracking the simulation, calculating the deviation of the high-level schedule and the actual simulation. The schedule and the MAS organization are adapted during the simulation if their deviation exceeds a given limit. Local scheduling methods can re-organize small parts of the simulation in a sensible way. The tracked agent activity gives a realistic estimate of the duration of design process and a suggestion how to organize it.

4 MULTI-LEVEL ORGANIZATION OF THE ASIC-MAS

We developed a two-stage planning strategy to obtain an efficiently organized MA simulation.

4.1 Long-term Planning

We implemented a long term scheduling that takes place at the beginning of the simulation to gain a high level plan for an efficient simulation organization. This algorithm assigns resources to tasks and produces a (near) optimal task execution order. The objective of the optimization is the minimization of

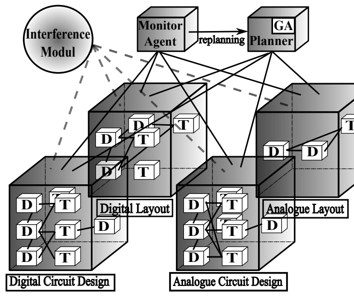

Figure 1: ASIC-MAS architecture.

the overall execution time of an ASIC design project: min T_{design}, while tasks need to be assigned to designers with different experience modes out of different design regions and tools considering task dependencies and resource constraints. We use a scheduling optimization heuristic based on genetic algorithms (GA) (Blaschke et al., 2009). Designers planned holidays are incorporated in the optimized schedule.

The high level scheduling is managed by one dedicated *planning agent*. He passes the necessary task and resource information to the GA scheduling algorithm and receives the optimization result of the GA. He assigns the tasks to requesting designers.

4.2 Short-term Planning

The high level plan is an optimized schedule under optimal designing conditions: every designer acts exactly as planned. But reality is different: designers become ill, take one or two days off, task execution takes longer or shorter than assumed and many recursions occur between tasks, resulting in new recursive tasks. The design process is dominated by these dynamic events, which cannot be modelled and considered by the static high level plan produced by the GA.

To model these events we developed an *interference module*. It introduces the dynamics inherent in the ASIC design process. Therefore, during processing the agent simulation deviates more and more from the optimized GA schedule. The reaction to this divergence is the local planning of the designer agents. Different mechanisms are used for different events, which are shown in fig. 2. We will describe the mechanisms in detail in the following subsections.

The *monitoring agent* tracks the simulation. He compares it to the high level plan at frequent intervals. The deviation at simulation time s_j is:

$$dev = \sum_{i=1}^{n} \Big(f(\delta(t_i, s_j)) \cdot (1 + a(t_i) - p(t_j)) \Big) + \delta(ra(s_j)),$$

where n is the number of tasks, $f(\delta(t_i, s_j))$ calculates the delay of the task t_i in the simulation compared to the schedule, $a(t_i)$ weights the number of dependent

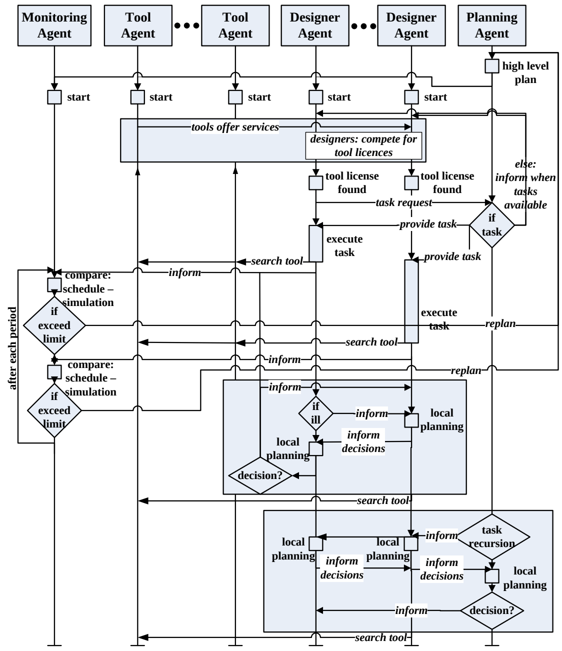

Figure 2: Short term scheduling negotiation architecture.

tasks to t_i, $p(t_j)$ is the buffer time of the task in the optimized high level schedule and $\delta(ra(s_j))$ is the resource workload of the simulation at time s_i. If this value exceeds a given limit, a global re-scheduling is initiated. All tasks that are being executed at that moment are still finished. All other tasks are scheduled anew with the GA.

Illness and short holidays are dynamic events induced by the interference module and thus not planable by the GA. As a global re-schedule would be oversized and far to timeconsuming, a local planning mechanism is chosen for an adequate reaction. If one of these events occurs while a designer is accomplishing a task, the task execution is interrupted and the overall execution time prolonged. As many tasks do have dependent tasks and the schedule is also optimized with respect to the resource workload, this prolongation delays tasks scheduled later, which leads to a bad resource workload and project elongation. The consequences of task delays can often be reduced by a *local re-planning* of the designer agents and a transfer of partial tasks from one agent to another. The decision and planning process be seen in flow chart 2. The goal is the minimization of a project delay.

Recursions often occur between frontend and backend tasks if the backend discovers that some specifications can not be fulfilled. These recursions cannot be planned. Hence we defined a Markov Chain that evokes recursions. This implicates new tasks with interdependencies to other tasks which have to be inserted into the simulation. As a new GA scheduling is not always necessary, a *local planning* between the designer agents takes place. They try to insert the new tasks into the simulation in an efficient manner. The local planning process is shown in fig. 2.

5 MAIN RESULTS

In this section we present the results of our ASIC-MAS and give a statement about its applicability.

5.1 Verification of the Model

We generated several problems to verify the GA optimization as well as the MA simulation in conjunction with the interference module.

In this section we describe and analyse the performance of our MAS by applying it to a representative but clearly arranged problem. At first we demonstrate the correct performance of the simulation. We chose a simulation run where a re-scheduling occurs, a task is handed over from one agent to another and a task recursion happens. Then we analyse the simulation part of the MAS before a re-schedule is initiated (fig. 3) and the second GA schedule that is produced after the re-schedule (also fig. 3). At the end we analyse the whole simulation run (depicted in fig. 4).

A project execution schedule is produced at every begin of a simulation run by the GA. The makespan of the schedule is 20 days. This is the minimal makespan possible under consideration of all constraints.

After the schedule optimization we applied the high level schedule to the MAS with the interference module switched on. Fig. 3 shows the MA simulation until the re-schedule stop occurred, and the new GA-optimized global schedule for the remaining tasks. The simulation deviates from the schedule due to different reasons. Some tasks need a longer execution time than planned, delaying the schedule and other dependent tasks. Abscent designers prolong task executions. After 11 days the simulation-schedule deviation is too large, a re-schedule is initiated. All tasks that have already been started are finished despite the re-schedule. The new global schedule supplements perfectly the simulation considering all constraints.

The Gantt chart of the tracked agent activity of the whole simulation is shown in fig. 4. Task4 needs one day less to be accomplished than planned. Therefore Task1, which is dependent on resources previously occupied by Task4, can be started and ended earlier. The execution of Task5 takes one day longer than planned. The first digital designer becomes ill while executing Task6. He offers the remaining task

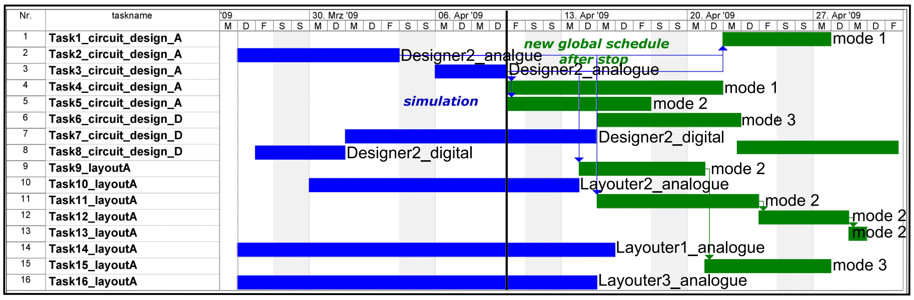

Figure 3: Two stages of the whole simulation, separated by a horizontal bar.

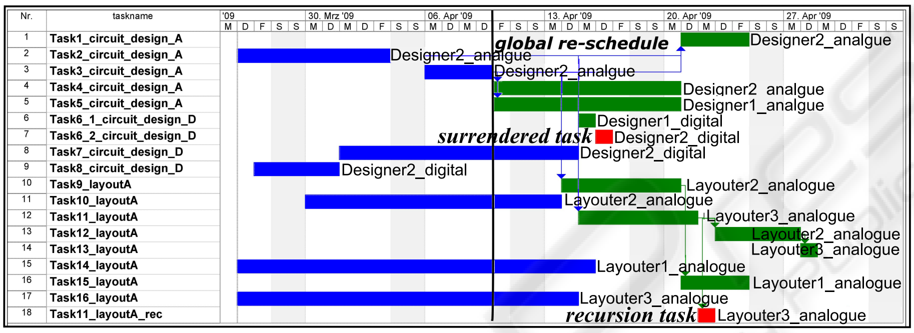

Figure 4: Tracked agent activity of the whole simulation.

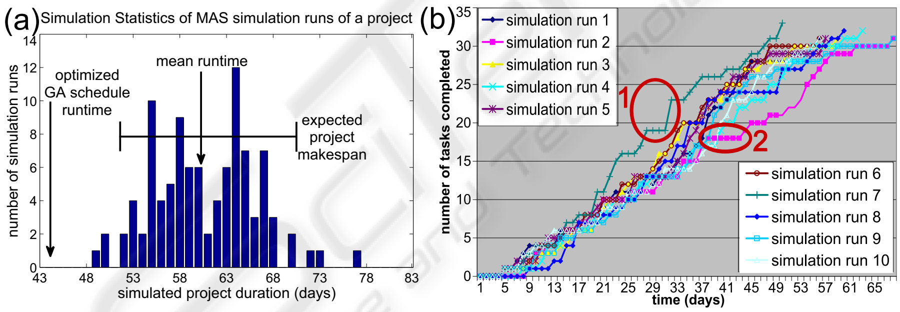

Figure 5: (a) Statistic of several simulation runs, (b) completion degree of simulation runs.

to the other agents of his design region. As Designer2_digital does not work on a task at that point he takes Task6. Task11 is processed in two days less than planned. A recursion is induced after it is completed. A recursive task, Task11_rec, is created and proposed to all analogue layouters. Only Layouter2_analogue and Layouter3_analogue are free and can compete for that new task. According to the schedule Layouter2_analogue has to execute a task earlier than Layouter3, Layouter2 wins the task. The simulation of the second part is finished one day earlier than the second global schedule proposed.

The optimized global schedule has an overall execution time of 20 days. The overall execution time of the MA simulation lasts 23 days. In consideration of all interrupting events a deviation of three days from the optimal execution time is little. As well the global re-scheduling as the two local scheduling techniques re-organize the simulation so efficiently that these perturbations do effect the overall project simulation time little. The re-scheduling techniques respond efficiently to and head of the prolonging effects of interruptions and improved the resource utilization.

5.2 Analysis of Projects

The ASIC-MAS can be used for an extensive analysis of projects. Multiple design-simulation runs give statistics of estimated runtimes. A sample of a statistic shows fig. 5(a) for a project with 30 activities. It shows the results of 100 simulation runs. The optimal GA-schedule runtime (45) days is indicated by an arrow, as well as the mean simulation runtime of 61 days, a project runtime of 60 days $\pm$ 9 days can be expected. The mean degree of project completion for every day of the simulation depicts fig. 5(b) for 10 simulation runs. The analysation of several runs gives an overview of advantageous start configurations (circle 1) and critical activities that often lead to bottlenecks (circle 2). Here, simulations starting with activity 8, 5 and 15 tend to result in shorter runtimes than simulations starting with activity 2 or 13. Bottlenecks are entailed sometimes by activities 1, 3, 6, 26 and 30. With this information the project leader can react adequatly.

6 CONCLUSIONS AND FUTURE WORK

The generated schedule and the MA simulation can be used for analysis, optimization and planning of finished, ongoing or future projects. The schedule optimization provides suggestions for an efficient organization of projects and permits a subsequent analysis for identification of weak spots within the project course. The setting of different project parameters, e.g. different resource availabilities, allows what-if-analysis. The simulation allows to observe the project execution. An interfere module disturbes the smooth simulation of the optimized schedule, allowing as-real-as-possible simulations to gain reliable and realistic predictions on project courses and makespans. Local scheduling and global re-scheduling methods head off prolonging effects by organizing and allocating parts in an efficient and sensible way.

In future experiments we are going to evaluate the performance of the System to investigate how good our whole model reflects reality.

REFERENCES

Blaschke, J., Sebeke, C., and Rosenstiel, W. (2009). Using genetic algorithms for planning of ASIC chip-design project flows. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2009)*, page P135.

Easley, S., Freeman, E., and Steele, D. (1989). Asic design project scheduling. *ASIC Seminar and Exhibit, 1989. Proceedings., Second Annual IEEE*, pages P11–2/1–4.

Leppelt, P., Hassine, A., and Barke, E. (2006). An approach to make semiconductor design projects comparable. In *7th Asia Pacific Industrial Engineering and Management Systems Conference (APIEMS 2006)*, pages CD–ROM. Asian Institute of Technology.

Lin, C.-H. and Kim, J.-C. (2006). A novel scheduling methodology for asic design. *System-on-Chip for Real-Time Applications, The 6th International Workshop on*, pages 131–134.

Moore, G. E. (1998). Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86(1):82–85.

Schurr, N., Marecki, J., and Tambe, M. (2005). The future of disaster response: Humans working with multiagent teams using defacto. In *In AAAI Spring Symposium on AI Technologies for Homeland Security*.