1. Publishable summary

1.1 Project context and objectives

Increasing design costs are the main challenge facing the semiconductor community. Assuring the correctness of the design contributes to the major part of the problem. However, while diagnosis and correction of errors are more time-consuming compared to error detection, they have received far less attention, both, in terms of research works and industrial tools introduced.
Another, orthogonal threat to the development is the rapidly growing rate of soft-errors in the emerging nanometer technologies. According to roadmaps, soft-errors in sequential logic are becoming a more severe issue than in memories. However, the design community is not ready for this challenge because existing soft-error escape identification methods for sequential logic are inadequate.

The aim of DIAMOND is improving the productivity and reliability of semiconductor and electronic system design in Europe by providing a systematic methodology and an integrated environment for the diagnosis and correction of errors. DIAMOND will develop:
- A unified, holistic diagnostic model for design and soft errors;
- Automated localisation and correction techniques based on the unified model, both pre-silicon and post-silicon;
- Implementation of a reasoning framework for localisation and correction, encompassing word-level techniques, formal, semi-formal, and dynamic techniques;
- Integration of automated correction with the diagnosis methods.

DIAMOND reaches beyond the state-of-the-art by proposing an integrated approach to localisation and correction of specification, implementation, and soft errors. In addition, it considers faults on all abstraction levels, from specification through implementation down to the silicon layout. Handling this full chain of levels allows DIAMOND to take advantage of hierarchical diagnosis and correction capabilities incorporating a wide range of error sources.

The partners of the project include Tallinn University of Technology (TUT), Linköping University (LIU), The University of Bremen (UNIB), Technical University of Graz (TUG), TransEDA (TEDA), Testonica Lab (TL), Ericsson and IBM Israel.

1.2 Description of work and results of the first 12 months

In order to get a general picture about the DIAMOND work plan refer to Figure 1 that shows the project flow. As the starting point, requirements and end-user needs will be defined by the industrial partners of the project. Based on these requirements a unified, holistic diagnostic model for design and soft errors will be developed and appropriate reasoning engines will be prepared. The core of DIAMOND is the design error diagnosis and correction and soft error analysis methods developed at the transaction and implementation levels, as well as post-silicon in situ diagnosis and repair methods. Finally, the project results will be evaluated on the benchmark designs constituting the DIAMOND platform. In the following, the results of the first project year are presented.

Requirement analysis and benchmarking
We conducted a study of the requirements and expectations of the end-users. This study was based on the industrial partners' inputs regarding the technology bottlenecks, development problems and particular requirements from the entire diagnosis and repair framework that is going to be developed in the scope of the DIAMOND project. The results indicate that there is a vital need for cross-level solutions for both diagnosis and repair flows for different kinds of errors. Moreover, the performance of many current

methods that are used in diagnosis and repair flows (e.g. simulation, equivalence checking, etc.) is very poor. Finally, a lot of manual effort should be eliminated and substituted by automatic flows. These problems and requirements correlate with the key objectives of the DIAMOND project.

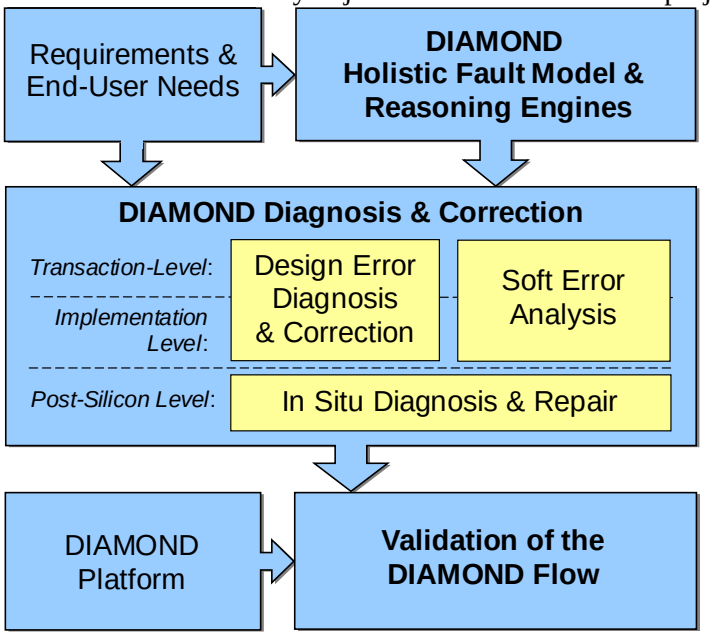

Figure 1. The DIAMOND Flow

A number of individual diagnosis and repair benchmarks were made available within the consortium in order to allow evaluation of DIAMOND tools. The industrial partners—IBM and Ericsson—chose benchmarks from their current activities. In addition the consortium selected 5 open source designs. Altogether, these benchmarks cover a wide range of designs and architectures, of sufficient size and complexity to provide challenges for diagnosis and repair of faults on different levels and the tools supporting it.

The holistic diagnostic model

Requirements for the holistic diagnostic model to be used in DIAMOND have been defined. The diagnostic model can be considered the "backbone" of the DIAMOND infrastructure. DIAMOND considers different application domains – diagnosis and correction – at different levels in the design flow – transaction, implementation and post-silicon. The aim of the diagnostic model is to integrate those different views as much as possible, to capture the individual requirements by the different applications. We devised use cases for the different abstraction levels considered within DIAMOND. For each abstraction level the use cases span all applications. From these use cases the requirements and the diagnostic model were derived.

Reasoning engines for diagnosis and correction

Formal, semi-formal, and dynamic reasoning engines are the core of the diagnosis, correction, and analysis approaches developed within DIAMOND. Different types of engines are required to trade reasoning power for resource requirements. Moreover, fine-tuning of the reasoning engines for particular types of applications is necessary to achieve the required efficiency and the expected quality of results.

Formal reasoning engines include an engine for debugging specifications using model-based diagnosis, engine for diagnostic test pattern generation, an engine particularly adopted for determining latency values for circuits (in order to identify for how many time cycles the computation of a sequential circuit may be influenced by erroneous values on certain signals), and tuning the formal reasoning engines for

evaluating the vulnerability of latches with respect to soft errors. The development of these engines was mainly driven by IBM, TUG, TUT, and UNIB.

Semi-formal reasoning engines contain a symbolic and concolic execution engine for localization and correction at the transaction level, a semiformal engine for latency analysis mentioned above and semiformal
engines for the analysis of soft errors along with mechanisms to handle soft-errors. These engines were developed by IBM, TUG, TUT, and UNIB.

Dynamic engines are a powerful approach to analyze very large systems. Additionally, dynamic engines are applied to further utilize results returned from other, more powerful engines. The engines developed within DIAMOND include the simulation engine based on HLDDs, a dynamic engine for latency analysis which is complementary to the formal and semi-formal engines mentioned above, and more general frameworks for pre-silicon and post-silicon/in-situ diagnosis. The dynamic engines and reasoning frameworks were designed and implemented by TEDA, TL, TUT and UNIB.

Diagnosis methods and tools
At the implementation-level (i.e. register-transfer and logic level) several methods and tools have been developed. This includes a design error localisation method based on HLDD backtrace, the latency analysis tool for identifying the time a sequential circuit may be influenced by erroneous values, a model-based diagnosis tool, a dual model framework to increase the throughput of a soft errors injection framework by identifying error vanish time and a critical fault list generation to limit the number of soft errors to be simulated. The implementation level tools have been developed mainly by UNIB, TUG, IBM, TEDA and TUT.

The post-silicon and in situ diagnosis tools have been mainly developed by Ericsson, LIU and TL. These include a method for efficient embedding of deterministic test data to enable flexibility in the type of commands that can be issued while keeping the memory requirements for storing deterministic test data low and the fault management approach containing techniques for error classification and for parallel scheduling of tests with synchronisation. In addition, a fault management architecture for on-line diagnosis is being developed.

Regarding the tool integration within DIAMOND, TEDA is building a verification Portal that will make use of techniques developed. The Portal will accommodate each of the partner's tools running on their discrete machines connected by a common network. A common set of design cores, the DIAMOND platform, will be used so that each participant's engine can be tested against the same designs. Metrics will be developed to show the effectiveness of each partners' algorithms. Initial integration of the first tool from TUT into the TEDA workflow engine took place in December 2010.

Dissemination
The web site for the DIAMOND project was created under the URL: http://fp7-diamond.eu/. In addition, two flyer designs have been created, including a one-sided project flyer that can be also printed out as a poster and a foldable, brochure-type flyer. Both flyers have been made available for download at the DIAMOND project web site

A large number of press releases were published worldwide, altogether in 10 different languages, including local TV, radio interviews, articles in items like EETimes, CNN, USA Today, Processor, IT Jungle, Design&Reuse, PR Newswire and several others.

The project was exhibited at the European Project Booth of the Design Automation and Test in Europe (DATE) Conference and Exhibition in March 8-12, 2010, Dresden, Germany. In addition, 21 scientific papers related to the topics of DIAMOND were published by the consortium.

1.3 The expected final results

The result of the project will be a **scalable and robust environment for handling faults in semiconductor design.** This development environment will provide diagnostic information needed for the correction of faults, be they design faults, implementation faults, or soft errors. The diagnostic information will be either on the source-level, the logic level, or the gate level, depending on the level of abstraction on which the design is provided. Such information may consist of the location of a fault or a soft error vulnerability. The environment will also suggest measures for correction of the fault or vulnerability wherever possible, where the type of the correction will be suitable to the level of abstraction in which it is needed. The environment will significantly reduce the time spent on error analysis and design modification, hence increasing the designers' productivity.

1.4 Potential impact and use

The goal of DIAMOND is to reduce the silicon area overhead and the performance degradation while keeping power and energy consumption under control. The objective is to be met by intelligent use of coding and selective use of corrective measures. Further, the purpose will be to develop techniques using partial re-execution. An aim is to utilise currently unused cores in order to minimize the need of additional silicon. The goal of DIAMOND is to keep the overhead (silicon, performance degradation, power, and energy consumption) at a minimum.

Fault localisation and correction in ICs constitute 2/3 of the total time spent on verification and debug. Verification and debug (i.e. assuring the correctness of the design) or in other terms the time spent on it, in turn **represent the main reason of excessive costs**. At the same time, due to entering the deep-submicron era (below 45nm), the semiconductor industry is faced with rapidly **growing rates of soft errors**. **We estimate that DIAMOND design flow would bring about**

- 50 % reduction in time spent on fault localisation and correction
- 25 % reduction of the time to identify a given number of soft error escapes
- 30 % increase in the number of identified unprotected soft errors

Note, that the provided improvements take into account the design costs only and do not include costs required for chip construction, such as e.g. mask costs.

The planned impact of DIAMOND will not be achieved during the project, but as a result of it.