Projet ANR VALMEM

Délivrable : D1.3

Titre : Conception flow applied on an industrial development

Auteurs : R. Chevallier

Version : 0.1

Date : 22 November 2007

VALMEM : Validation fonctionnelle et temporelle des mémoires embarquées décrites au niveau transistor par des méthodes formelles
Conception flow applied on an industrial development

<table>
<thead>
<tr>
<th>Author</th>
<th>date</th>
<th>Release</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remy Chevallier</td>
<td>22 November 2007</td>
<td>0.1</td>
<td>First release</td>
</tr>
<tr>
<td>Remy Chevallier</td>
<td>14 December 2007</td>
<td>1</td>
<td>Improve full cut section</td>
</tr>
</tbody>
</table>
1. Introduction

The aim of this document is to have an overview of the time spent by the design teams on each main task during the development of a eSRAM compiler. For confidential reasons, the timescale is not provided in this document. In order to have a deep description of the main tasks, please refers to the delivery ‘D1.1: State of Art in eSRAM design and validation flow’.

2. Main design steps

2.1. Build critical path

The main steps are the following:

- **Database building**: Develop the schematic and layout database for the eSRAM compiler
- **Leafcells building**: Define, design and verify the leafcells. They are the elementary bricks for the eSRAM design
- **Leafcells extraction**: Generate the spice netlist with the layout parasitic network
- **Memory assembly**: Define the strategy and develop the program which generate the eSRAM layout
- **Functional verification**: Simulate each design marginality and check that the design follow the specification
3. Full cut verification

- **Parasitic comparison of critical path versus full cut:** Check that the critical path is modeling correctly the complete eSRAM at the parasitic (R, C) point of view.
- **Reference cut: LVS DRC checks:** Tune and verify the layout generation flow on the reference cut of the compiler.
- **Full cut extraction:** Generate the spice netlist of the full cut with the parasitic.
- **Back end validation:** Check the back end rules (DRC and LVS) on the layout for all the possible input of the compiler (i.e.: number of bit, of word, multiplex factor, speed, redundancy…)
- **Timing validation:** Verify that the full cut timings are consistent with the critical path timings.
- **Power Analysis:** Verify the power integrity of the compiler.
4. HDL model design

- **Verilog building**: specify, design and verify the eSRAM modeling according to the specification.
- **Check Verilog versus Spice design**: check that the behavior of the Spice netlist is compliant with the behavior of the Verilog models.

5. Conclusion

According to this study, the design team spends:
- 1/3 part of its time in building the database
- 1/3 part of its time is used to simulate the design and proves that the specifications will be verified in all cases.
- 1/3 part of its time is split between the remaining tasks: HDL model building, layout verification.

According to this schedules, it makes sense to improve the functional verification (transistor based simulation) in order to speed-up the development time and improve the coverage of the functional verification.