On the compilation of a parallel language targeting the self-adaptive virtual processor
Bernard, T.A.M.

Citation for published version (APA):
Enschede: Print partners Ipskamp

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
## Contents

### 1 Introduction

1.1 Classical microprocessor improvements ........................................... 2  
1.2 Multicore architectures ...................................................................... 2  
1.3 Exploiting concurrency as a solution .................................................. 3  
1.4 Impact of concurrency on software systems ......................................... 9  
1.5 Contribution of this thesis .................................................................. 12  
1.6 Overview of this thesis ....................................................................... 12

### 1 Foundations

2 Background in parallel computing systems ............................................. 15

2.1 Approaches in concurrent execution models ........................................ 18  
2.2 Relevant parallel architectures ............................................................ 22  
2.3 Modeling concurrency in compilers ..................................................... 24  
2.4 Requirements for a concurrent execution model ................................. 25

### 3 SVP Execution Model and its Implementations

3.1 Our approach to multicore programming ............................................ 27  
3.2 Presentation of the SVP execution model ............................................ 28  
3.3 Hardware implementation: Microgrid .................................................. 29  
3.4 Software implementation: μTC language ............................................. 36  
3.5 SVP system performance .................................................................... 44  
3.6 Discussion and conclusion .................................................................. 49
# II  Compilation for Parallel Computing Systems

4  From basics to advanced SVP compilation

4.1 Basics in compiler transformations

4.2 SVP compilation schemes

4.3 Under the hood of SVP compilation

4.4 Conclusion

5  On the challenges of optimizations

5.1 Hazards with optimizations

5.2 Investigating some optimizations

5.3 Discussion and conclusion

6  Implementing the SVP compiler

6.1 Role of the compiler

6.2 Compiler design decisions

6.3 Compilation challenges

6.4 Discussion and conclusion

7  SVP evaluation

7.1 Evaluation of SVP compilation

7.2 Evaluation of SVP computing system

7.3 Discussion and conclusion

III  Discussion and conclusion

8  Discussion and conclusion

8.1 Thesis overview

8.2 Limitations

8.3 Future work

8.4 Conclusions

A  $\mu$TC language syntax summary

Summary

Samenvatting
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgements</td>
<td>165</td>
</tr>
<tr>
<td>Publications</td>
<td>167</td>
</tr>
<tr>
<td>Bibliography</td>
<td>174</td>
</tr>
<tr>
<td>Index</td>
<td>175</td>
</tr>
</tbody>
</table>