The research of some key technologies of compiler optimization for on-chip multi-core processor

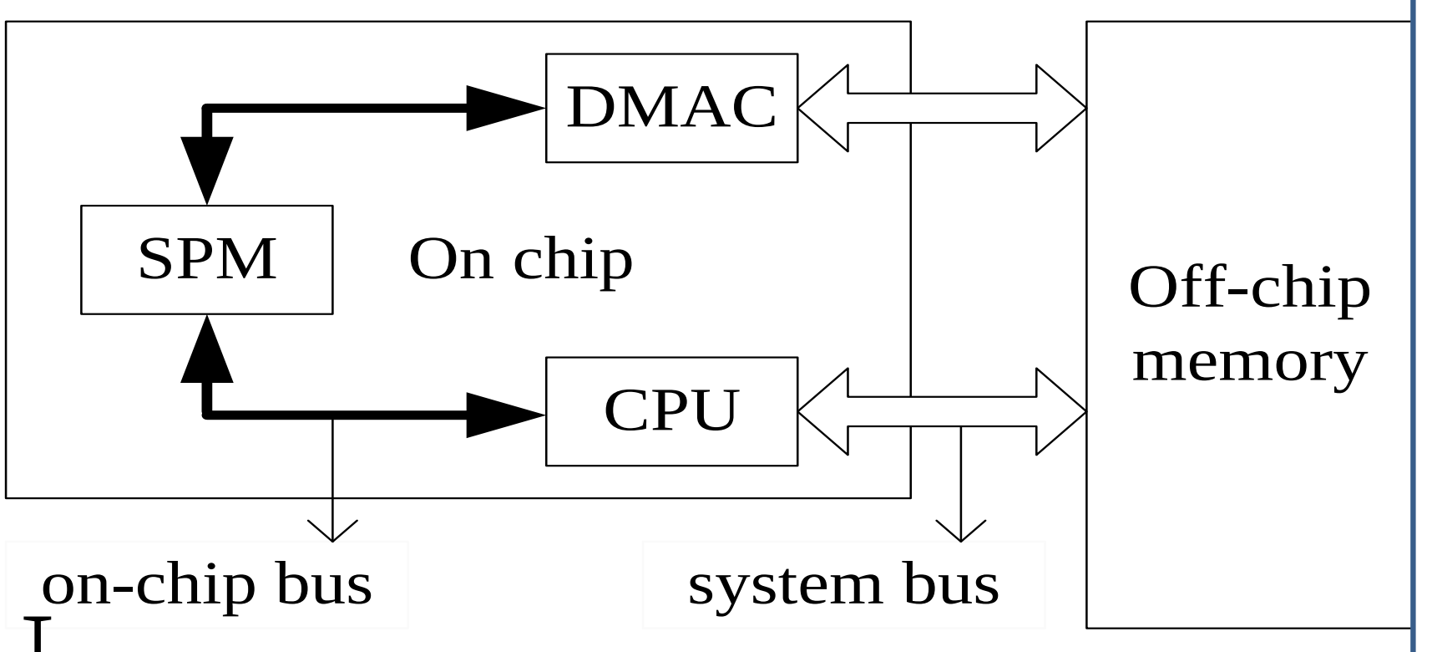
Minyi Guo (guo-my@cs.sjtu.edu.cn)
Director Professor of Embedded Pervasive Computing Center,
Chairman of Department of Computer Science and Engineering,
Shanghai Jiao Tong University, China.

Introduction

Multi/Many-core processor is the inevitable trend of development of processor, and has been applied in desktop, server and mobile and embedded system. However, the **power**, **memory** and the **parallelization of software** limited the development of multi-core processor. In this project, we studied several key technologies of the compiler optimization for on-chip multi-core processor: **Architecture of Multi-core Processor**, **Acceleration Model**, **Compiler Optimization**, **Low-power framework and methods**, **Task Stealing Scheduling** etc.

Multi-core Processor Modeling

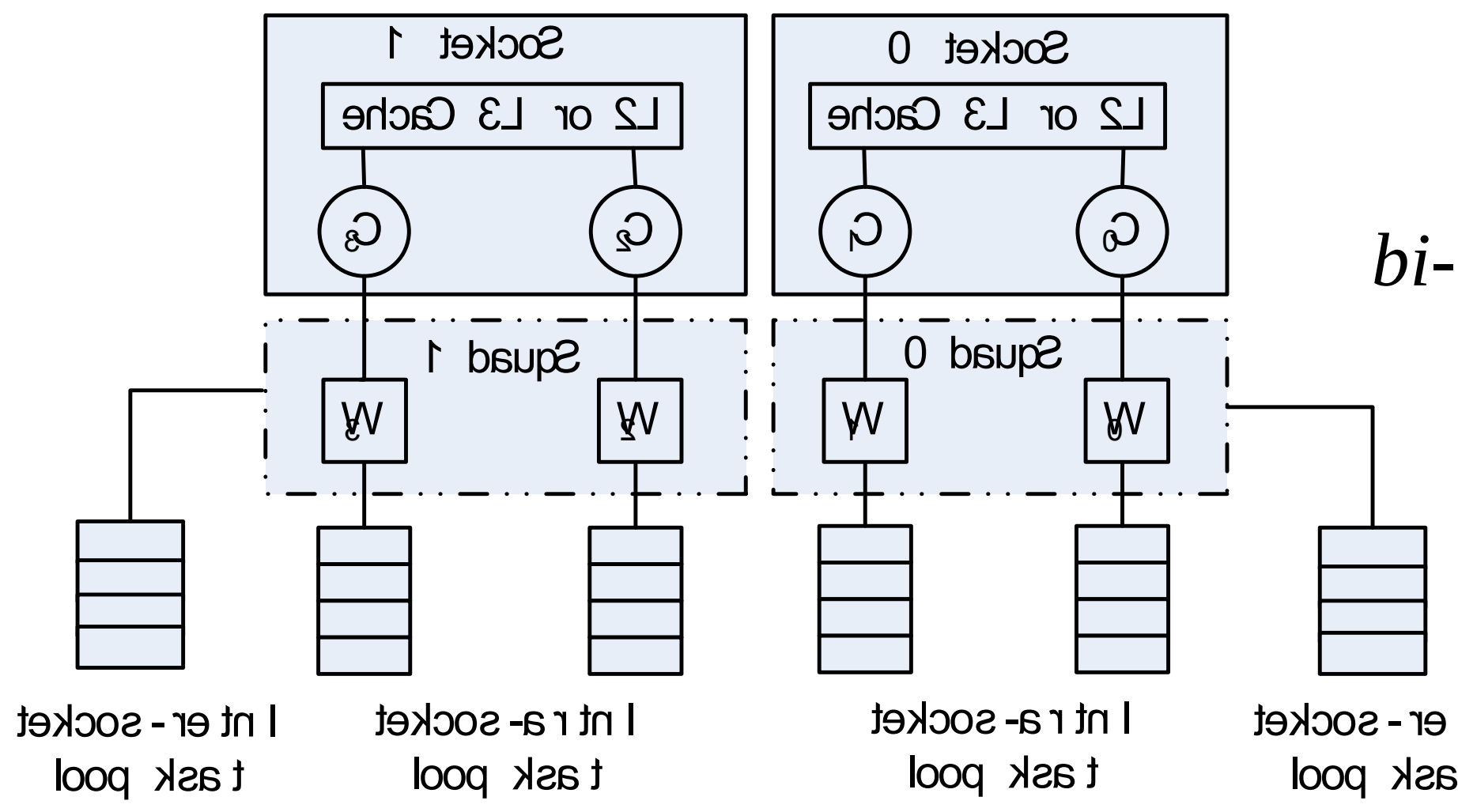
- **Scratch Pad Memory Model**
  - ◎Characteristic: Small-Capacity; low-power Consumption; high-speed access
  - ◎ Allocation strategy: Static Allocation; Dynamic allocation
- **Data Pipeline Model**
  - ◎ Dynamic voltage scaling
  - ◎ Instruction-level speed modeling
  - ◎ Graphical data fetch modeling
  - ◎ Independence between DMA and CPU
  - ◎ Data Predictability from loop iteration
- **Power Cost Model**
  - ◎ Quantify static power consumption
  - ◎ Quantify command power
  - ◎ Quantify power of cache and memory access
  - ◎ Multi-core communication power modeling



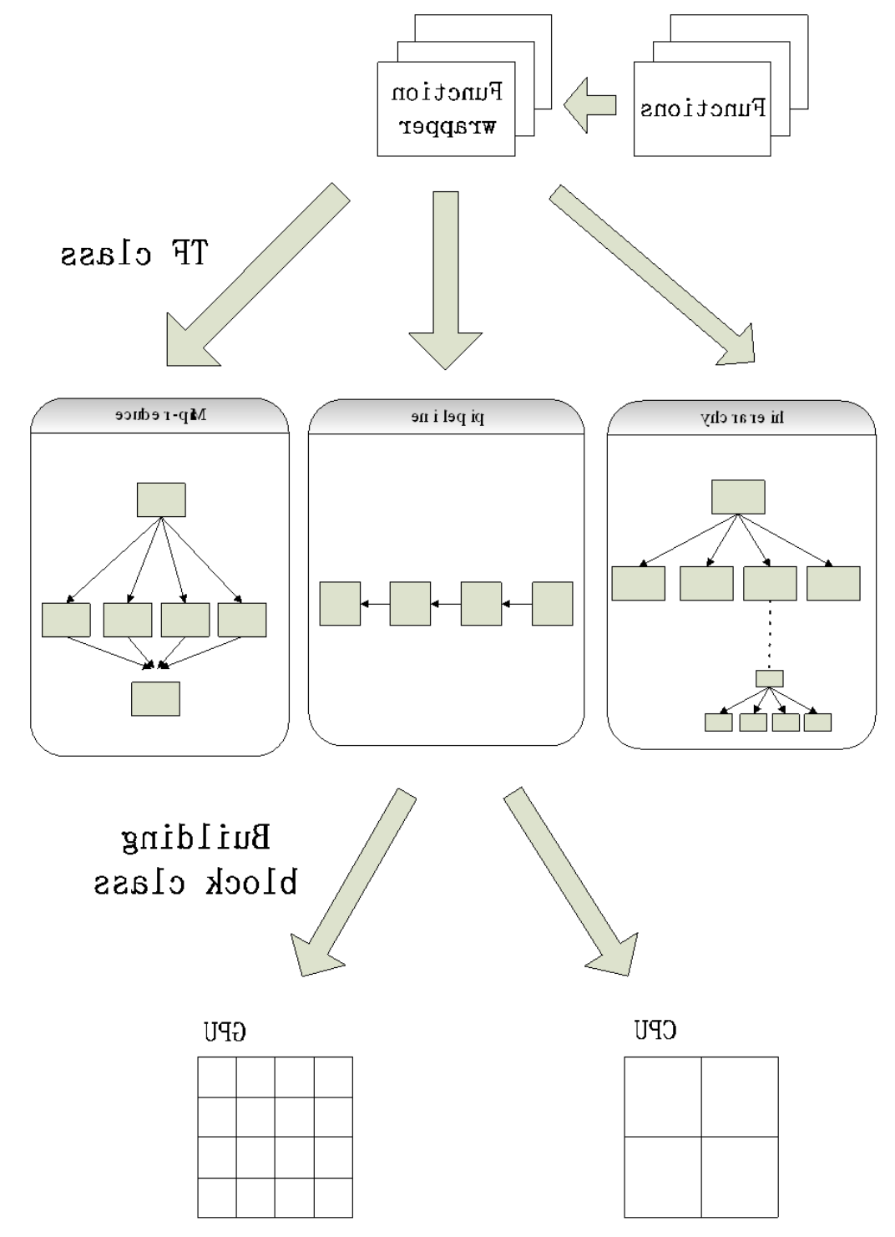
Basic model of McP

Multi-thread parallelization

- Template Meta-programming method
- Architecture of bi-tier task-stealing
- Cycle of sentence migration transformation



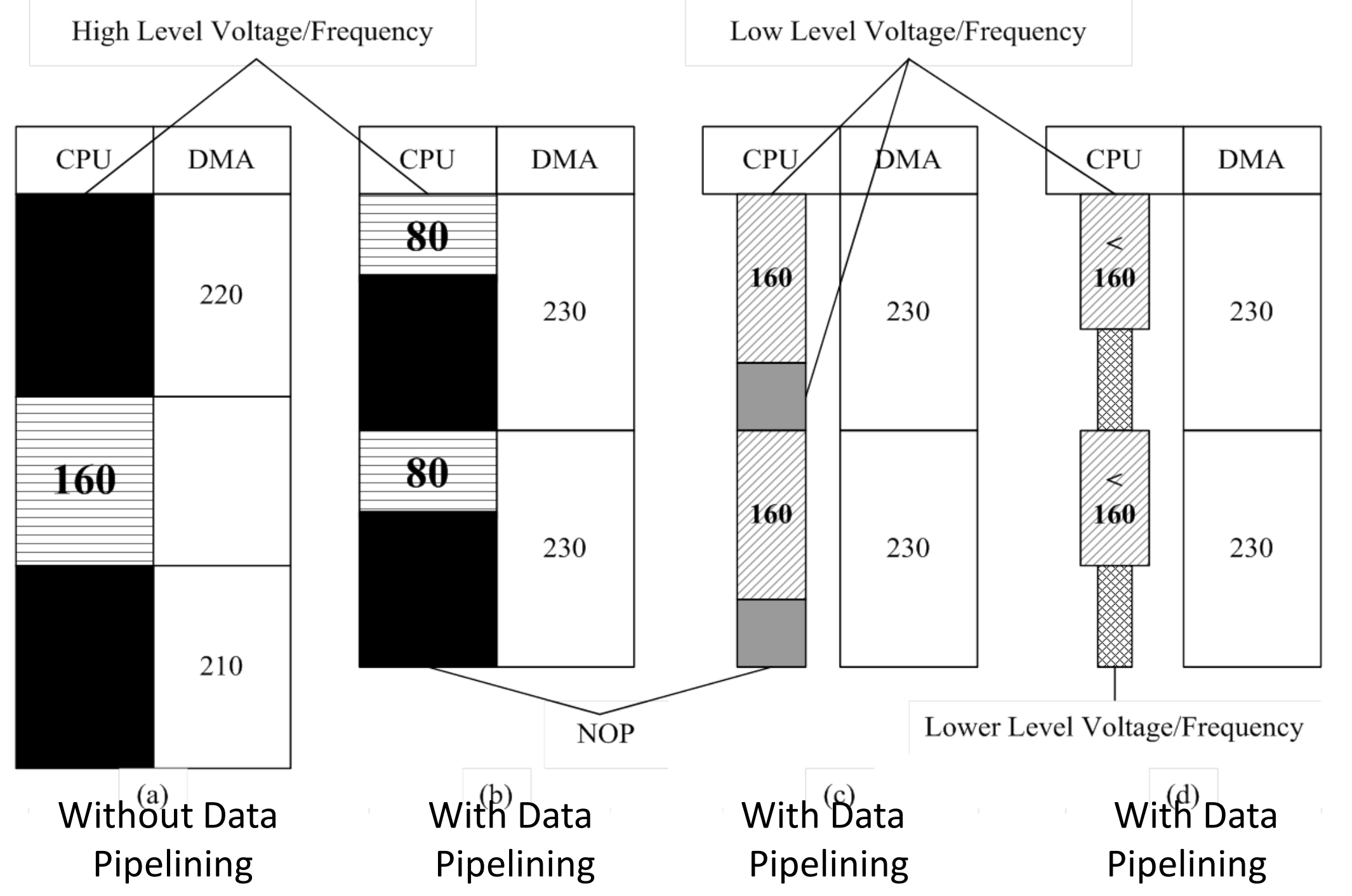
bi-tier task-stealing



Programming Model Overview

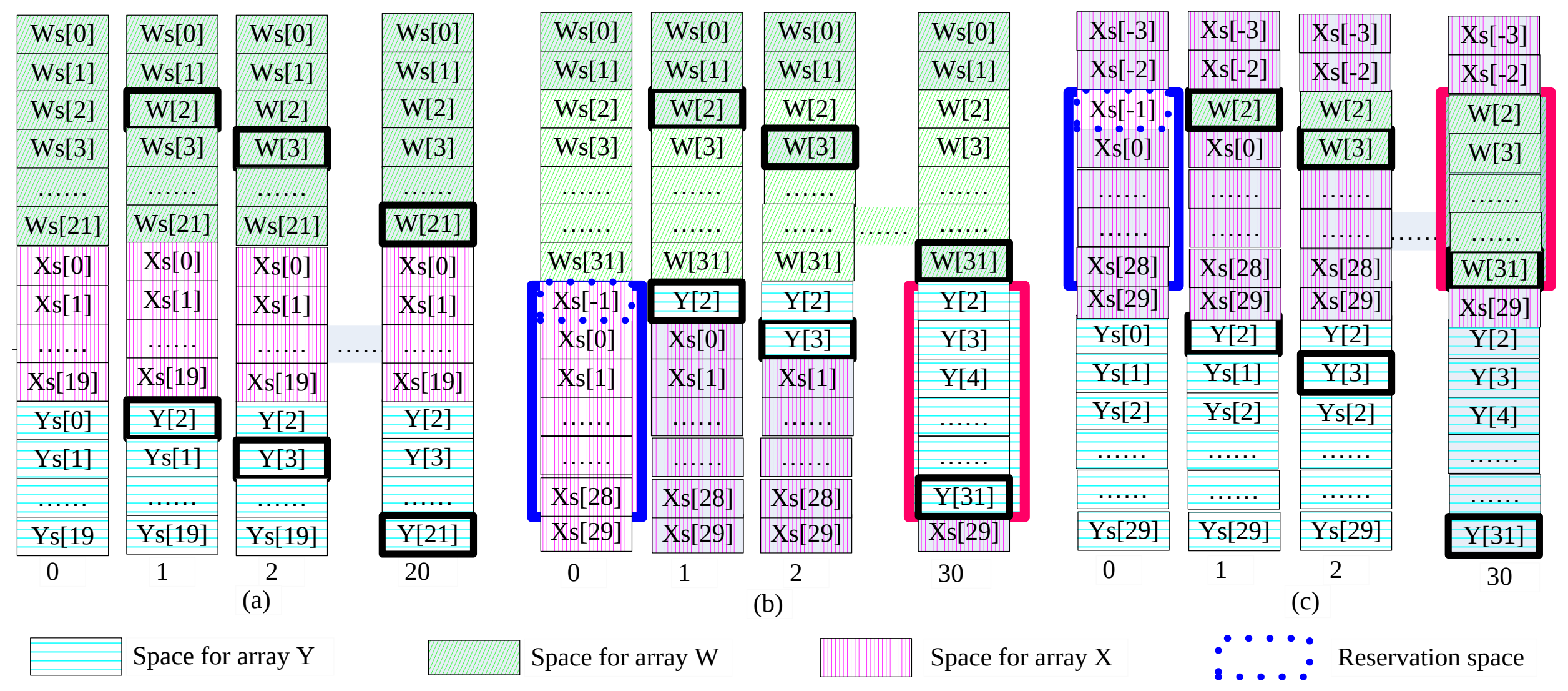
Compiler optimization and low-power-cost method

- Space overlapping model
- Data pipelining
- Voltage power adjusting
- Power Reduction Structure
- Label compression

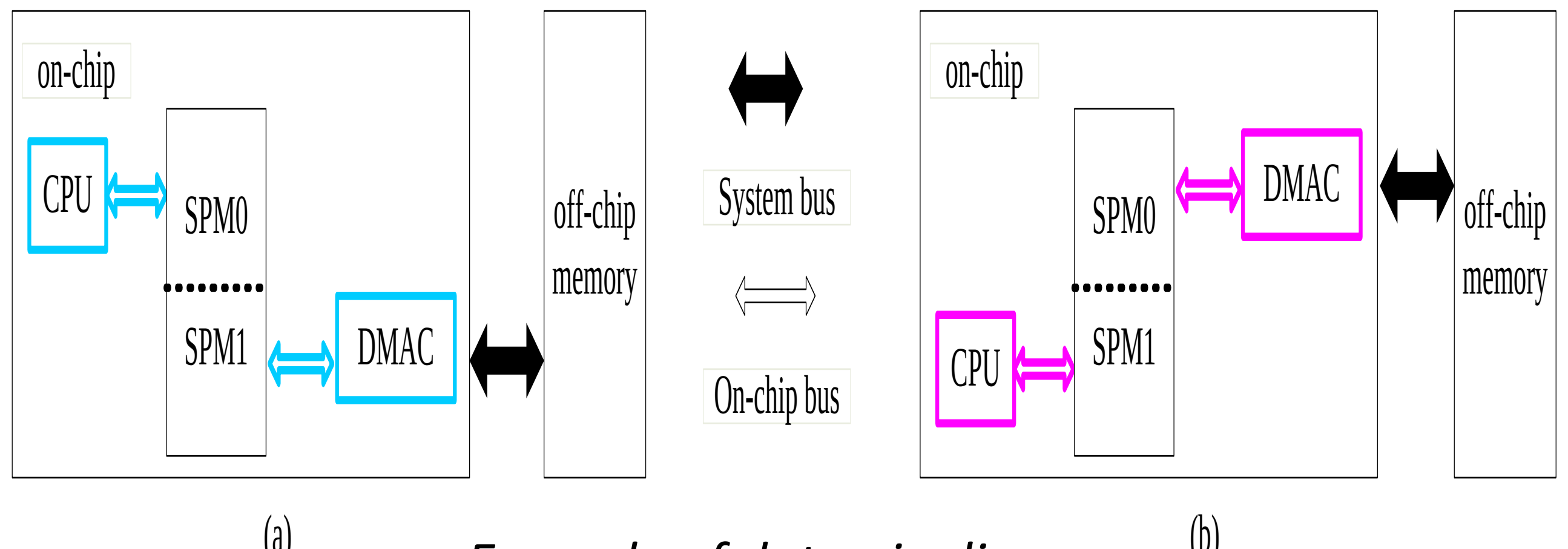

High Level Voltage/Frequency
Low Level Voltage/Frequency
CPU
DMA
220
160
210
80
80
230
230
160
160
230
230
&lt;
160
&lt;
160
230
230
NOP
Lower Level Voltage/Frequency
(a)
(b)
(c)
(d)

Without Data Pipelining | With Data Pipelining | With Data Pipelining | With Data Pipelining

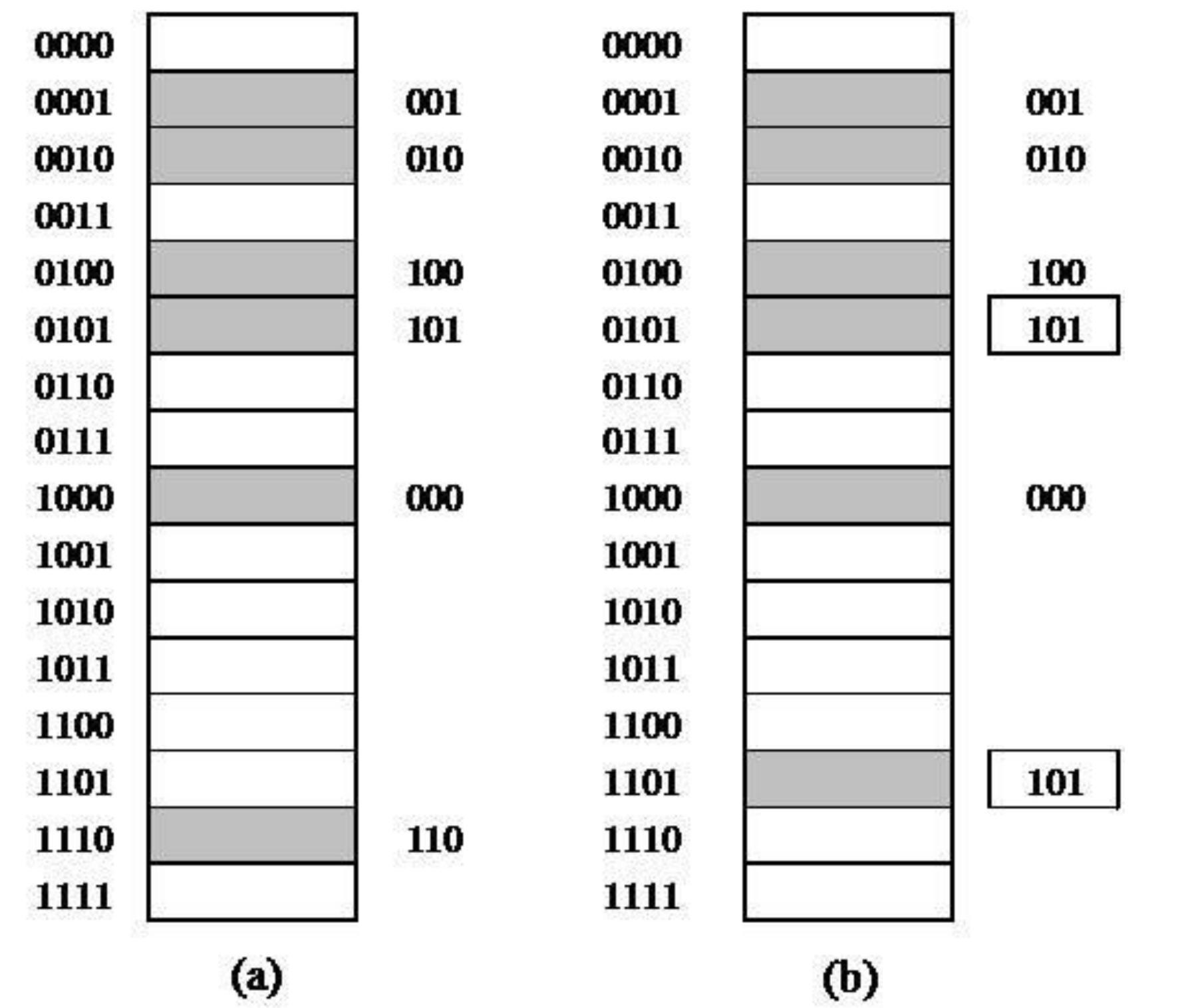
Different Operation scheduling



Space overlapping model



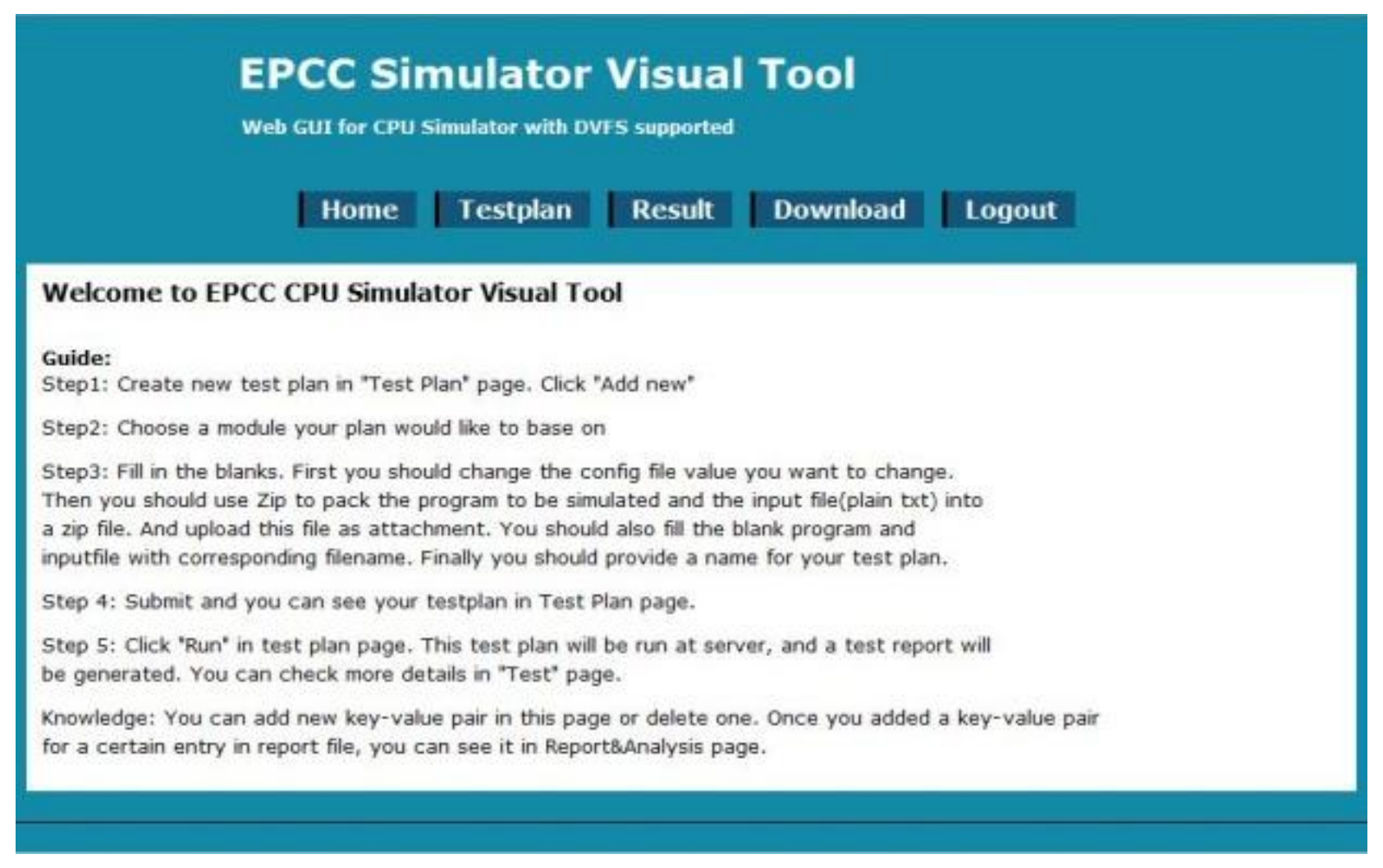
Example of data pipeline



Example of Label compression

Multi-core simulation tools (epcc.sjtu.edu.cn/EPCC_SESC.htm)

- Add DVFS instructions
- Calculate DVFS power using Watch model
- Provide three Benchmark examples
- Web-based visualization, user-friendly


EPCC Simulator Visual Tool
Web GUI for CPU Simulator with DVFS supported
Home
Testplan
Result
Download
Logout

Welcome to EPCC CPU Simulator Visual Tool

Guide:
Step1: Create new test plan in "Test Plan" page. Click "Add new"

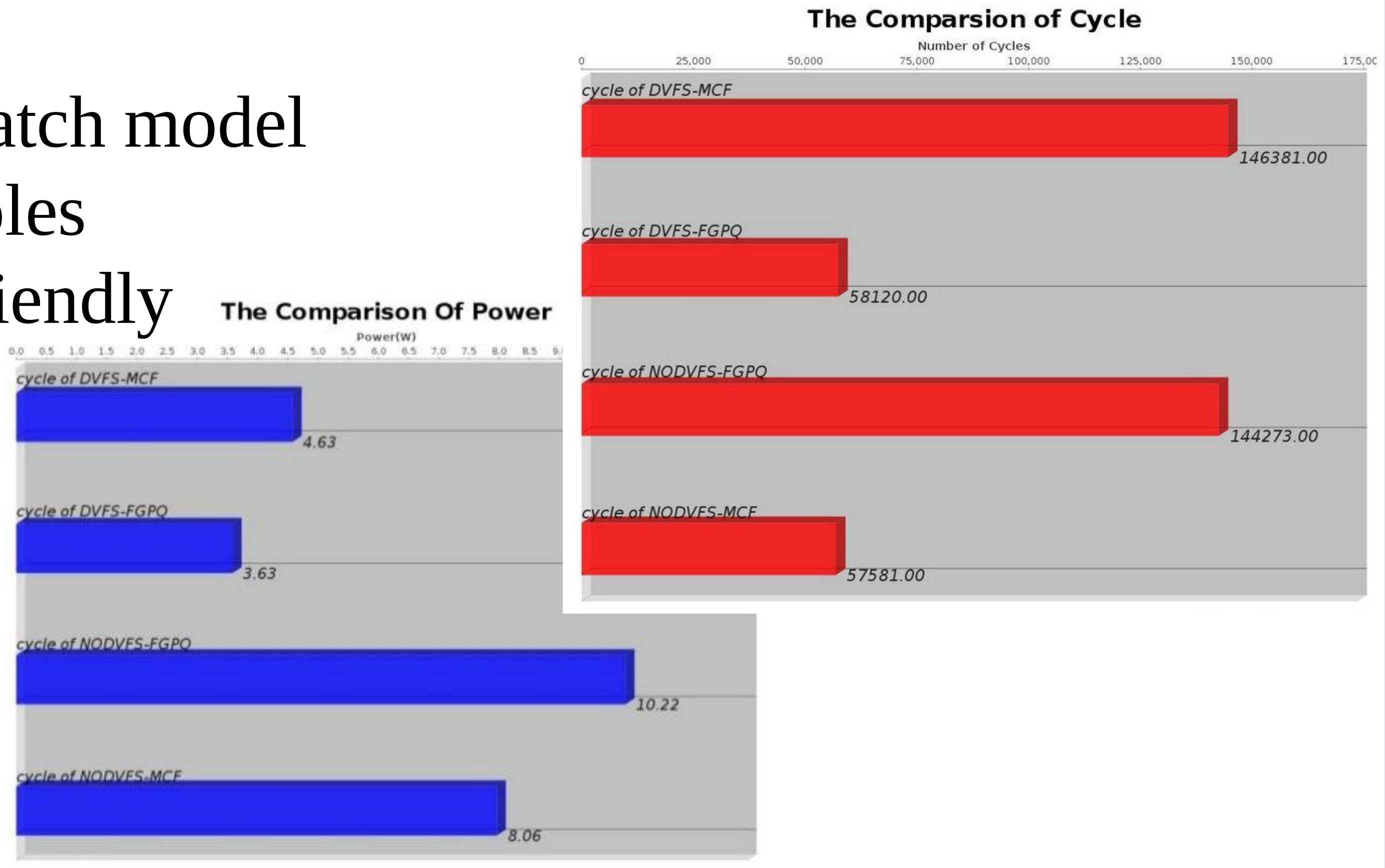
Step2: Choose a module your plan would like to base on

Step3: Fill in the blanks. First you should change the config file value you want to change. Then you should use Zip to pack the program to be simulated and the input file(plain txt) into a zip file. And upload this file as attachment. You should also fill the blank program and inputfile with corresponding filename. Finally you should provide a name for your test plan.

Step 4: Submit and you can see your testplan in Test Plan page.

Step 5: Click "Run" in test plan page. This test plan will be run at server, and a test report will be generated. You can check more details in "Test" page.

Knowledge: You can add new key-value pair in this page or delete one. Once you added a key-value pair for a certain entry in report file, you can see it in Report&Analysis page.



Conclusion

We use compiler to analyze and optimize user programs automatically. In the popular multi-core architecture scenario, the optimized programs are scheduled among cores to save power with guarantee of no vital performance degradation. Compiler is our tool of achieving high performance and low power consumption. These technologies provide the support for high-performance computer, cloud computing and software parallelization.

This material is based upon work supported in part by China National Funds for Distinguished Young Scientists Projects No. 60725208

NSFC EPCC 上海交通大學 SHANGHAI JIAO TONG UNIVERSITY