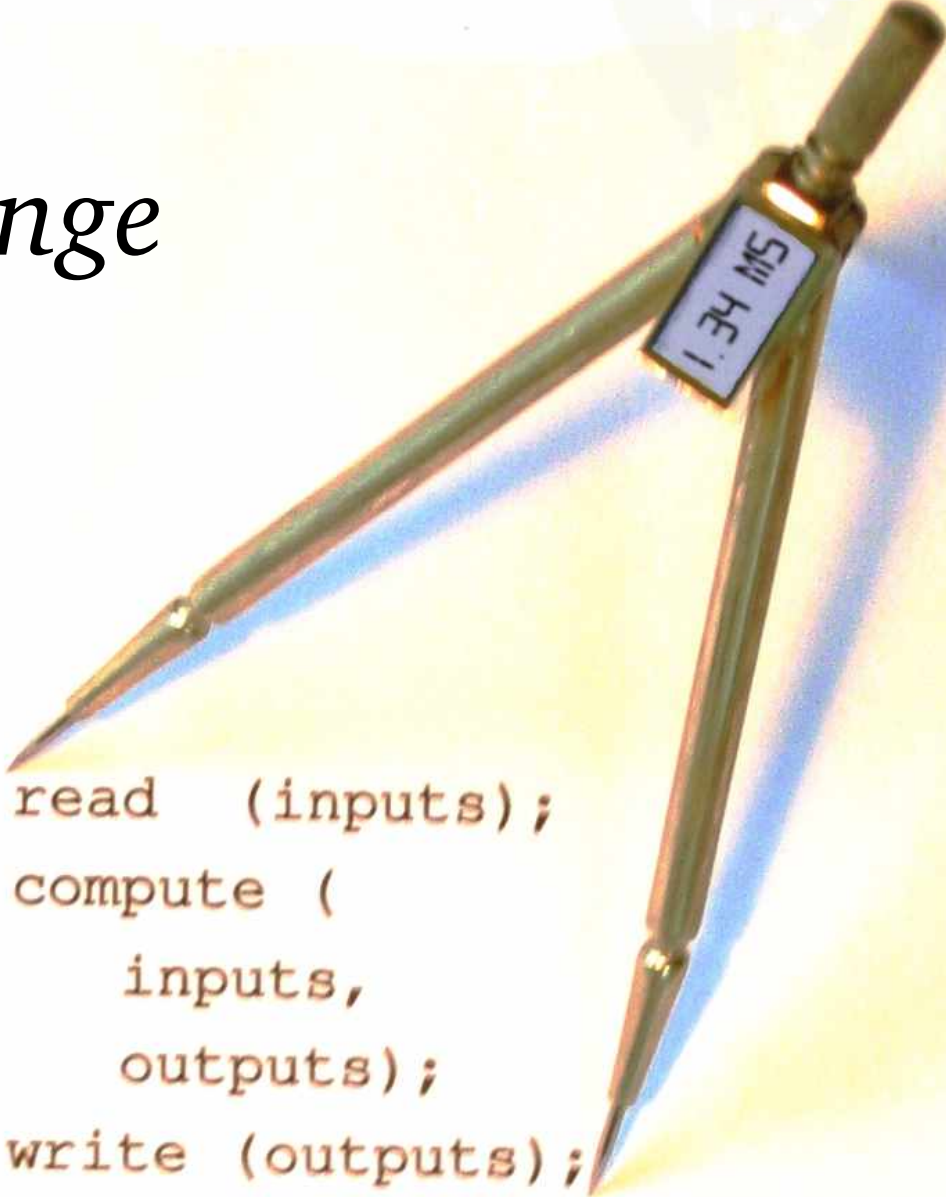


WCET Tool Challenge

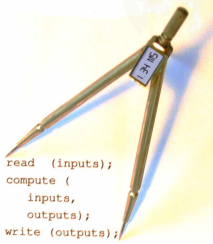
introduction
and
status report
and
questions arising



```
read (inputs);  
compute (  
    inputs,  
    outputs);  
write (outputs);
```

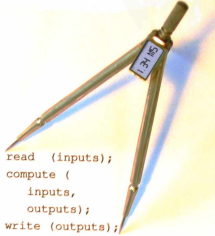
WCET Tool Challenge: Aims

- To be **useful** to **you** – the WCET analysis community
 - academic researchers
 - tool developers and vendors
 - users of WCET analysis, current and potential
- Useful how? **You** decide; perhaps some of these:
 - exhibit the nice features of **your** methods and tools
 - compare **your** methods and tools to **your** m. & t.
 - for **you** to be challenged and inspired by
 - difficult target programs
 - real target programs
 - new target processors, etc.
 - define a “cutting edge” by the best features of each tool
- What next? Up to ... **us**



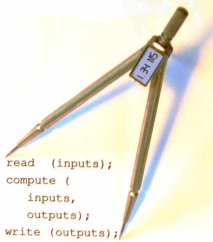
From 2006 to 2008

- First WCET Tool Challenge 2006 – pioneering success
- Planned changes for WCC 2008:
 - include **measurement-based** tools
 - suggest a **common target processor** (system)
 - better **definition** of benchmarks, problems, and results:
 - separate code to be analysed from “driver” code
 - portable code, eg. from 8-bit to 32-bit processor
 - **test suites** for measurement-based tools
 - define the analysis problems (questions)
 - include pure **flow-analysis** problems
 - common format for results from all tools
- Other changes that just happened:
 - no external, independent analyst (viz. Lili Tan in 2006)
 - completely new set of benchmarks (so far)
 - using Wiki for benchmarks and results



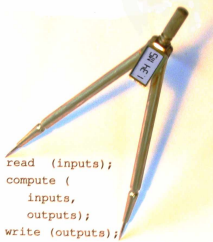
Status of Challenge

- Benchmarks
 - **DEBIE-1 DPU SW**, courtesy Space Systems Finland Ltd
 - SW to control space-located instrument
 - 8758 C lines, three ISR, three other threads
 - test suite for RapiTime (created with ARTIST2 support)
 - **“Loops and arrays”** by Rathijit Sen (Saarland U.)
 - synthetic C code; stress I/D cache; 4 programs
 - open to additions, eg. PapaBench, Mälardalen, ...
 - some new bm were promised, but not delivered
- Common target
 - **NXP LPC2138**, ARM7TDMI-S, 512 KiB flash, 32 KiB SRAM
 - cache-like Memory Accelerator Module for flash
 - two configurations: MAM off, MAM on.
 - not supported by all participants
 - some support only “MAM off” (fixed timing)



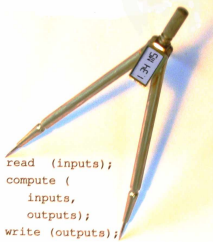
Status (cont.)

- Tools still participating (alphabetical order):
 - Bound-T
 - MTime
 - OTAWA
 - RapiTime
 - TuBound
- Participants who withdrew:
 - Chronos – too little time for ARM7 port (from SimpleScalar)
 - Heptane – no staff available to finish
 - some others whose participation was tentative
- **No** results by “deadline” (15 June)
 - OTAWA first to enter results
 - others (close to) completion
 - static analyses also for instrumented binaries from RapiTime
- Results to be announced... later (see Wiki site)

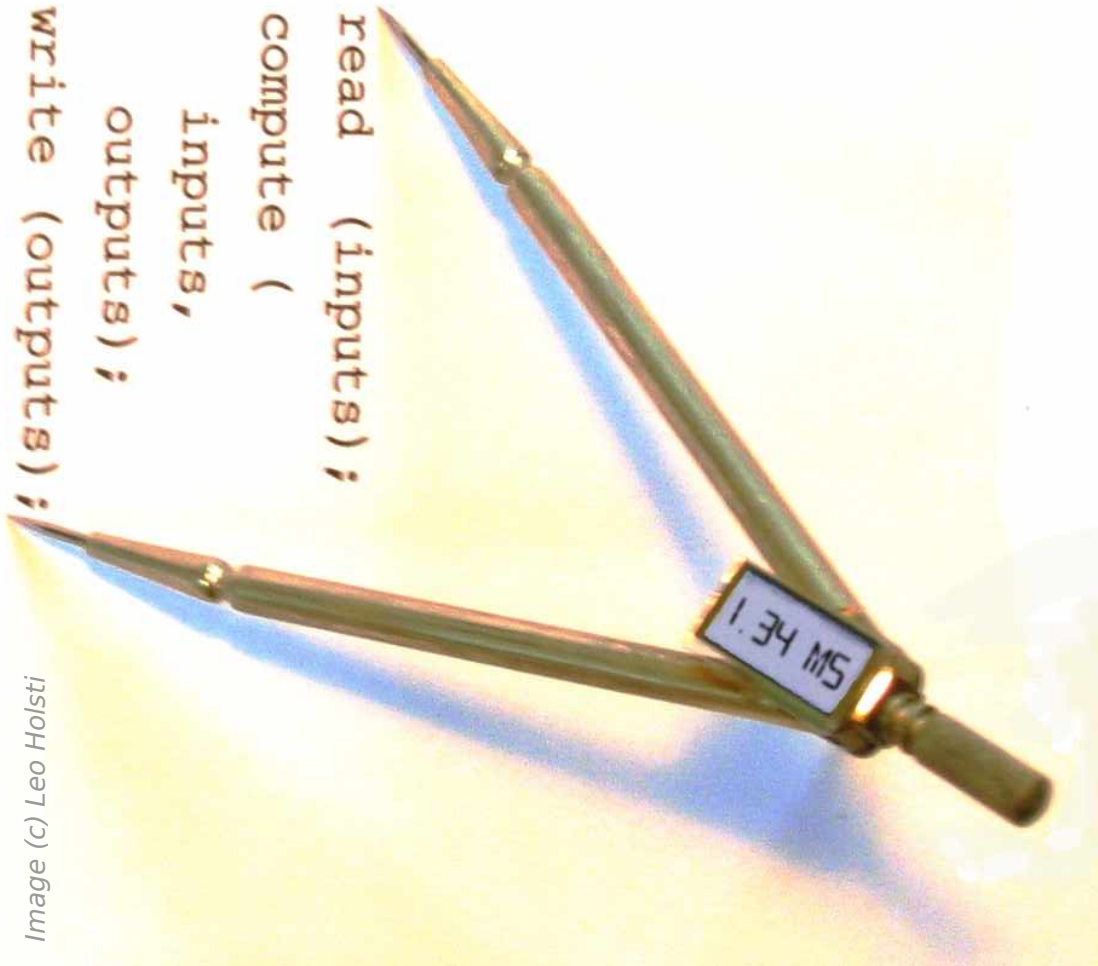


Questions arising

- Was (is) it useful to you? Should it be continued?
- What should be changed?
 - Were the benchmarks OK?
 - Is the LPC2138 + MAM too complex?
 - More work on MAM modelling?
 - Pick another “suggested” common target? Which?
 - Analysis problem definitions OK?
 - Are pure flow-analysis problems useful?
 - Using Wiki OK?
 - Editing result tables is cumbersome, error-prone
 - Other ways to enter & collect results?
- Organization for next Challenge (if any)
 - Who?
 - Financial support?



Thanks



Thanks to all Challenge participants!

WCC'08 working group:

Niklas Holsti (Tidorum)

Jan Gustafsson (Mälardalen U.)

Guillem Bernat (Rapita Systems and York U.)