



Status of the Bound-T WCET tool

Niklas Holsti and Sami Saarinen
Space Systems Finland Ltd.

Overview

- Brief intro to Bound-T
- Detail 1: Control-flow analysis of DSP code
- Detail 2: Concurrency of IU & FPU in SPARC
- Status and current work
- Questions for discussion

Brief intro to Bound-T

- *Static WCET analysis methods*
 - Reads compiled, linked binaries (with debug info)
 - Generates bounds on WCET and stack usage
 - Automatically bounds simple loops (Presburger arithmetic model, *Omega* tool)
 - Assertion language, not embedded in source code
 - Uses IPET for calculation (*lp_solve* tool)
- *Implemented by SSF*
 - Targets ADSP-21020 and SPARC V7 for ESA
 - Target Intel-8051 with SSF own funding
 - Platforms Sun Solaris, Intel Linux, Intel Win



Detail 1: Control flow in DSPs

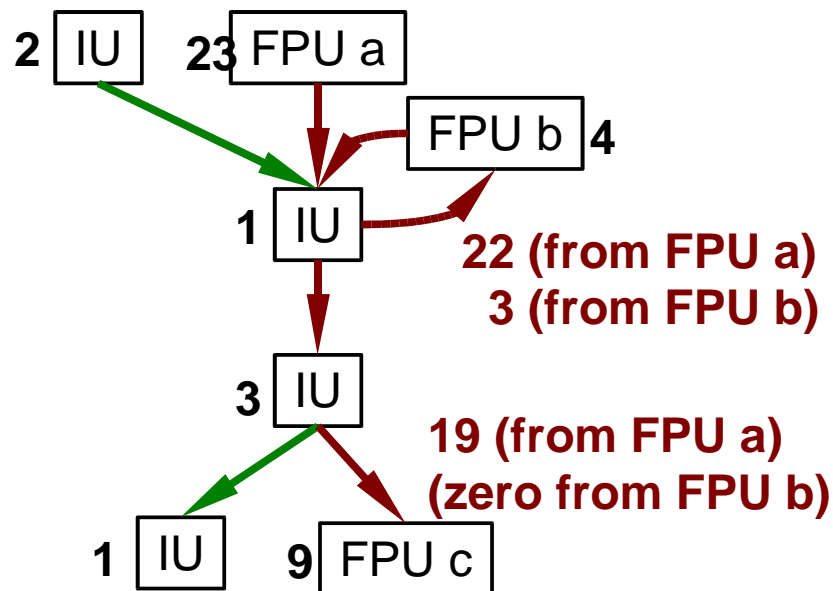
- *Special program sequencing in DSPs (e.g. 21020)*
 - Delayed branch due to instruction pipelining
 - Zero-overhead loop, nestable six deep
LCTR=30, DO Loop_End until LCE;
r14=3 // first instruction in loop
...
r2=r2+r14 // last instruction in loop
Loop_End: nop
- *Solution: CFG based on full sequencing state:*
 - All instruction pipeline contents (fetch, decode, exec)
 - All (dynamically) containing DO UNTIL loops
- *One instruction can cause many CFG nodes*
 - E.g. branch (to be) taken / not taken



Detail 2: SPARC IU/FPU concurrency

SPARC V7 (ERC32) functional units

- IU fetches instructions, dispatches to IU or FPU
- FPU executes concurrently with IU, up to 80 IU cycles
- On next FP instruction, IU waits until FPU is idle



"FPU busy" delay (red numbers) at second FP instruction on any delayed path (red edges) is distributed over all edges that lie on delayed paths, to minimize the impact on non-delayed paths (green edges), using ILP with a heuristic goal function.

Status and current work

- *Space area*
 - Some evaluation by customers going on
 - SSF will use Bound-T/SPARC on the GOCE on-board SW
- *Non-space area*
 - Looking for pilot customers, partners, distributors
 - Target: medium-sized mass-market embedded processors
- *Current technical work*
 - Evaluation support, minor improvements (manpower)



Questions to the audience

- *Division between syntax and semantic models?*
 - Syntax = program sequencing = CFG = **partitioning** states
 - Semantics = abstract CFG execution = **comprehending** states
- *General analysis of concurrent functional units?*
 - Our technique for SPARC IU & FPU seems a special case
- *Combining info from source code and object code?*
 - Convince compiler writers to export info for WCET
 - Define standard formats (ELF extension? XML?)
 - Working group => list of recommendations?
- *Is industry starting to know about WCET analysis?*