

BISC support in GCC

Zbigniew Chamski

zbigniew.chamski@gmail.com

Why bother at all?

- According to Ken Zadeck (2009 Summit)
 - BISC = Bull*it Instruction Set Computer
 - not RISC, not CISC
 - thus, irreleevant to GCC
- Challenge: VLIW, DSP, EPIC...
 - HW folks do look at power/energy consumption
 - some don't want to put a compiler backend in HW
 - they keep designing "lower-power" cores...
 - ... and asking for **effective** compiler support

Why should we bother?

- GCC support now a must for manufacturers
 - end-customer requirement
 - marketing
- some RISC targets have BISC-ish features
 - delay slots, partial interlocks
 - pipeline/architecture constraints visible in SW
- CISC could possibly get power reductions

BISC "features"

- explicit ILP and scheduling constraints
 - simultaneous operations: **must** be in same cycle
 - fixed delays between ops: **must** be X cycles apart
 - delay slots: X cycles x Y issue units (!)
 - lookup-based VLIW: compiler-defined microcode
- resource bottlenecks
 - clustered register files, constrained datapath
 - issue slot usage rules
- "weird" ops: saturated arithmetic, auto-incr/decr

Current limitations

- no general time-centric view of the schedule
- scheduling: DFA
 - complexity explodes if target has few constraints
 - peephole optimization of the schedule difficult
- register allocation
 - decoupled from scheduling
 - non-reg resource allocation on the scheduling side
- "weird" ops: non-blocking
 - solutions available, "just" plain development

Possible directions

- Squeeze features into existing infrastructure
 - C6x
 - other targets?
- Provide infrastructure for explicit scheduling/RA
 - insn reservation patterns (cycles x resources)
 - cycle-oriented intra-/inter-BB reservation tables
 - replicated resources (need just one extra RTX)
- Provide strategies based on the infrastructure
 - list/superblock scheduling, tail duplication, ...

Issues (& ancillary benefits)

- Size of explicit schedule representations
 - `#cycles x #resources x sizeof (single_use_descr)`
 - order of magnitude: `1k x 1k x 4words = 16-32MB`
- Infrastructure is a stepping store
 - need to implement algorithms/strategies
 - there are quite some published research results
- Resource awareness can help CISC targets
 - potential power/energy impact on OOO execution

Going forward

- There's GPLed, RTL-based proof it works
 - Salto package (Inria, ~1999-2000)
 - support for C6x, TriMedia; MIPS/Sparc peephole
 - contains basic scheduling and RA implementations
- Chicken-and-egg problem
 - no compiler support vs. no industry drive
- We'd need champions and sponsors ;-)
 - drive the design + promote the approach (int/ext)
 - get funding/support to make & complete the move

Thanks!