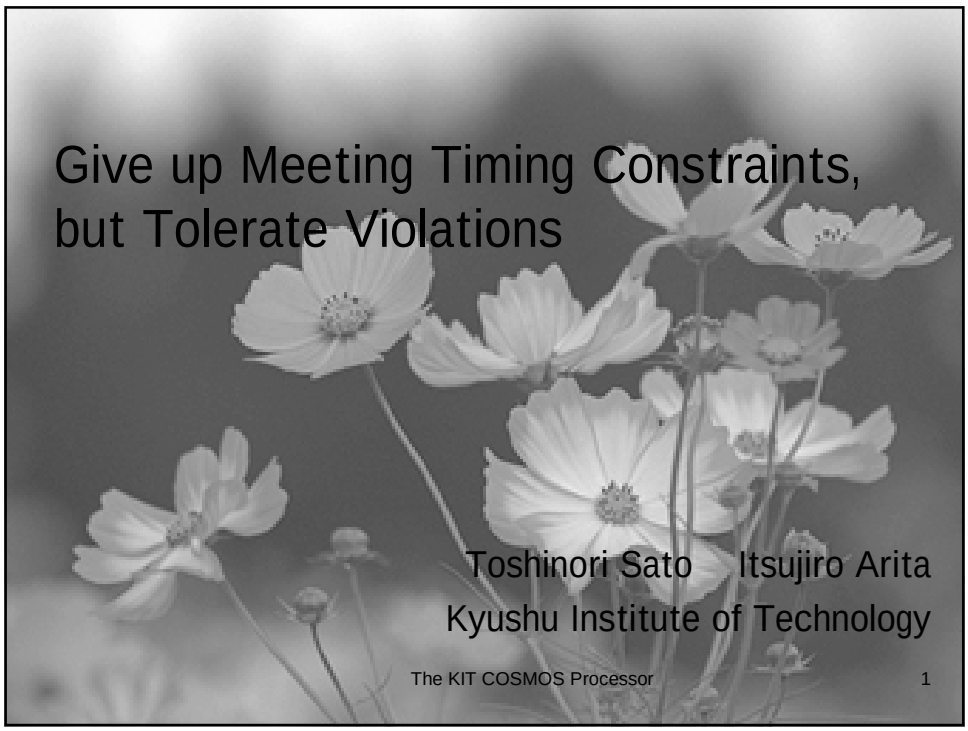


Give up Meeting Timing Constraints, but Tolerate Violations

Toshinori Sato Itsujiro Arita
Kyushu Institute of Technology

Outline

- Background and motivation
- Proposal
- Preliminary evaluation
- Conclusions

Background

- Microprocessors clock frequency is significantly increasing.
- Wire delay is being dominant for deciding critical paths.
- It becomes quite difficult to meet timing constraints under condition of fast TAT.

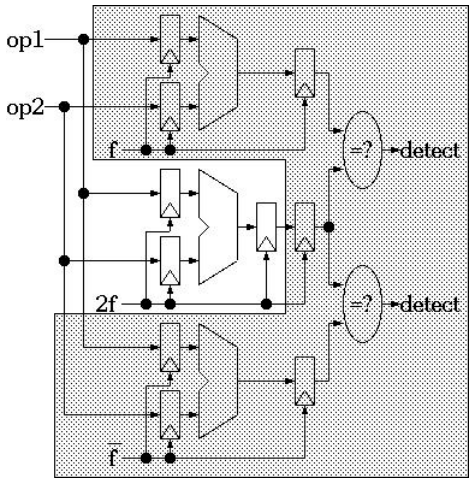
Motivation

- Circuit designers will become happy, if they are free from tedious timing design.
- This is possible when circuits correct timing failures by themselves.
- We propose to give up timing constraints, but to tolerate violations.

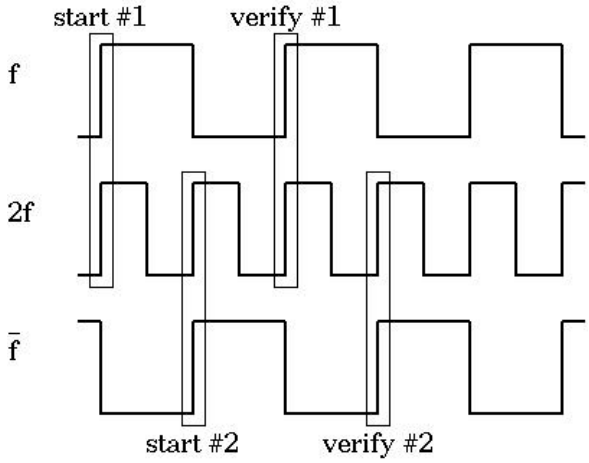
Tolerating timing faults

- Main and checker parts are provided.
  - They are equivalent circuits, but clock frequencies distributed are different.
  - Main part works at higher frequency and may occur timing failures.
  - Checker parts support main part, and detect timing failures.
- Correcting mechanism for branch prediction is exploited for timing failure correction.

Example circuit



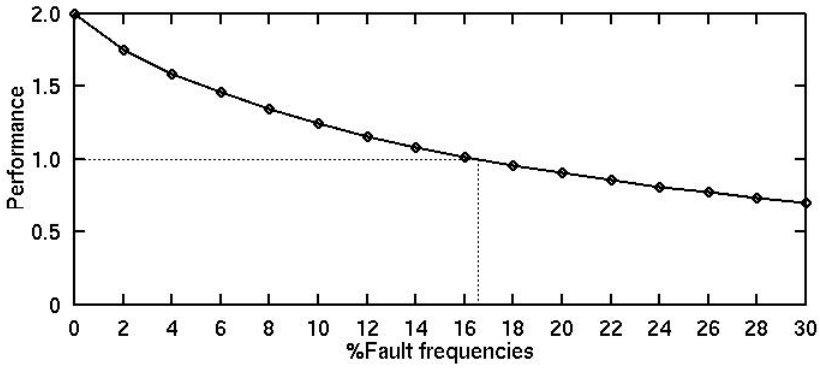
Verification process example



Evaluation methodology

- Processor model
  - Out-of-order execution 4-way superscalar processor
  - Compaq/Alpha instruction set architecture
- Timing simulator
  - SimpleScalar tool set version 3.0a
- Benchmark
  - SPEC2000 CINT

Simulation results - GCC



Discussions

- Performance is normalized by baseline.
- Clock frequency of main part is two times higher that that of checker parts.
- Performance is improved until fault frequency reaches 16%.
- This is encouraging result, that it is expected fault frequency is relatively small.

Conclusions

- We have proposed to give up tedious timing design, but to provide fault-tolerant mechanism for transparently correcting timing failures.
- From timing simulations, it is expected that clock frequency increases with maintaing performance when fault frequency is relatively small.

Web page

- More information is available at

  http://www.mickey.ai.kyutech.ac.jp/~tsato/cosmos