Optimizing for Energy Efficiency

Jeremy Bennett

Key Steps (Roy & Johnson)

- Choose the best algorithm for the problem at hand and make sure it fits well with the computational hardware. Failure to do this can lead to costs far exceeding the benefit of more localized power optimizations.
- Minimize memory size and expensive memory accesses through algorithm transformations, efficient mapping of data into memory, and optimal use of memory bandwidth, registers and cache.
- Optimize the performance of the application, making maximum use of available parallelism.
- Take advantage of hardware support for power management.
- Finally, select instructions, sequence them, and order operations in a way that minimizes switching in the CPU and datapath.

Kaushik Roy and Mark C. Johnson. 1997. "Software design for low power". In Low power design in deep submicron electronics, Wolfgang Nebel and Jean Mermet (Eds.). Kluwer Nato Advanced Science Institutes Series, Vol. 337. Kluwer Academic Publishers, Norwell, MA, USA, pp 433-460.

Modeling Energy

Energy Cost *(E)* of a program *(P)*:

$$E_P = \sum_i (B_i \times N_i) + \sum_{i,j} (O_{i,j} \times N_{i,j}) + \sum_k E_k$$

Instruction Base Cost, B_i, of each instruction i

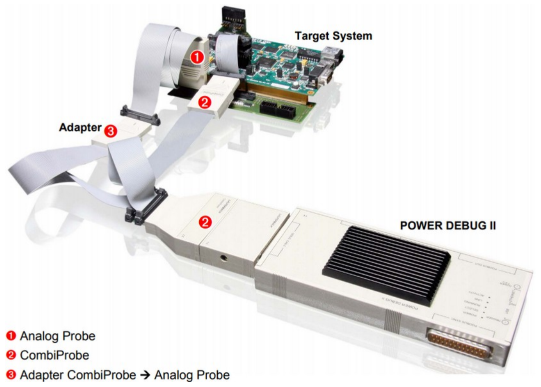
Circuit State Overhead, $O_{i,j}$, for each instruction pair

V. Tiwari, S. Malik and A. Wolfe. "Instruction Level Power Analysis and Optimization of Software", Journal of VLSI Signal Processing Systems, 13, pp 223-238, 1996.

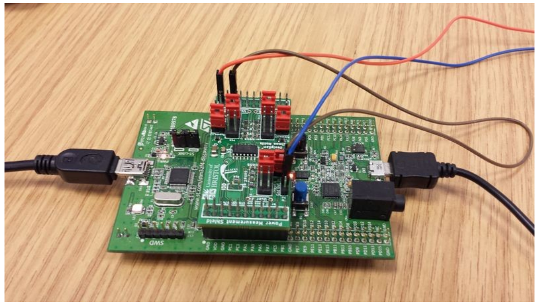
Modeling tools

- Wattch: A Framework for Architectural-Level Power Analysis and Optimizations
  - Proceedngs of the 27th International Symposium on Computer Architectures, June 2000, Vancouver BC, pages 83-94
  - 90% accuracy, fast
- TLM POWER3: SystemC TLM power estimation
  - D. Greaves and M. Yasin, "TLM POWER3: Power estimation methodology for SystemC TLM 2.0," Proceeding of the 2012 Forum on Specification and Design Languages, Vienna, Austria, 2012, pp. 106-111.
- Back end ASIC modeling
  - 99%+ accuracy, very slow

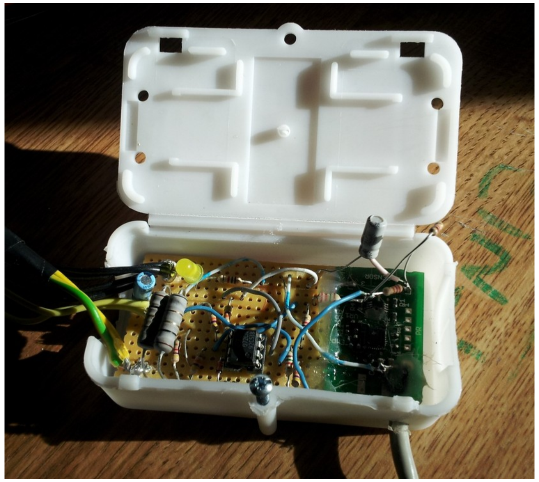
Measurement Hardware



Lauterbach Power Debug II



MAGEEC "Wand"



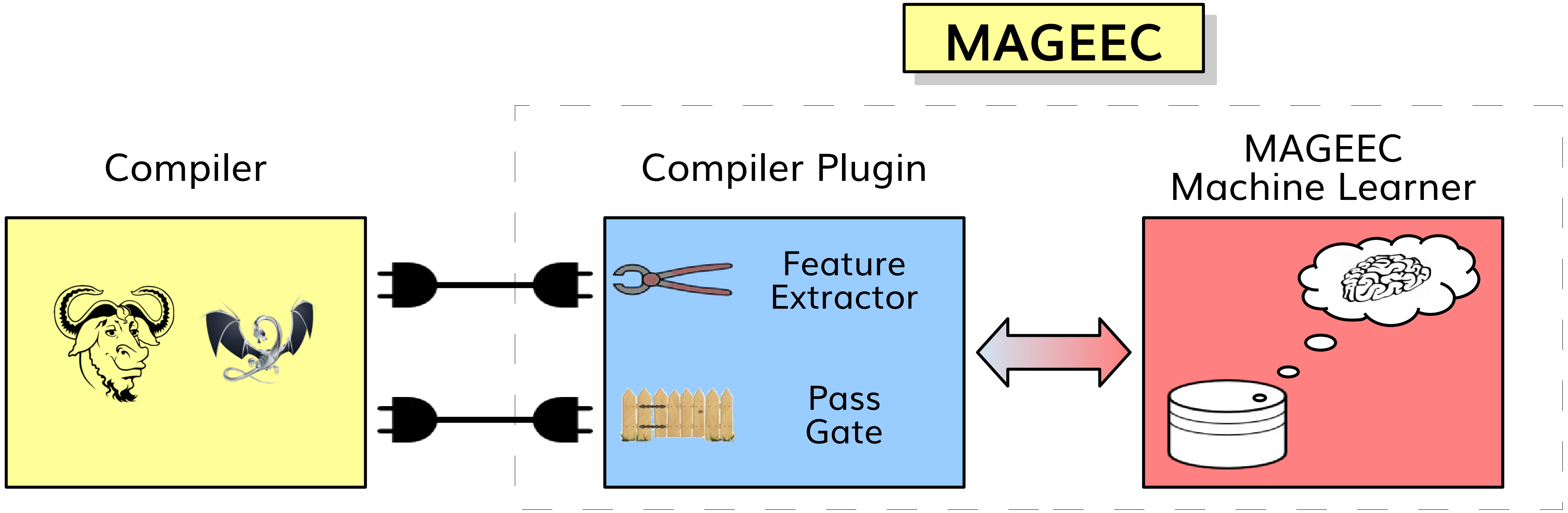
spEEDO Power Probe

- Intel RAPL on chip
- Arm on chip counters
- UltraSoc debug interface

Energy Awareness

- The Energy-Aware Computing Framework
  - https://github.com/eacof/eacof
- ENTRA: Whole Systems ENergy TRAnsparency
  - EC FP7 FET MINECC
- spEEDO: Debug centric solution
  - Innovate UK grant 131192

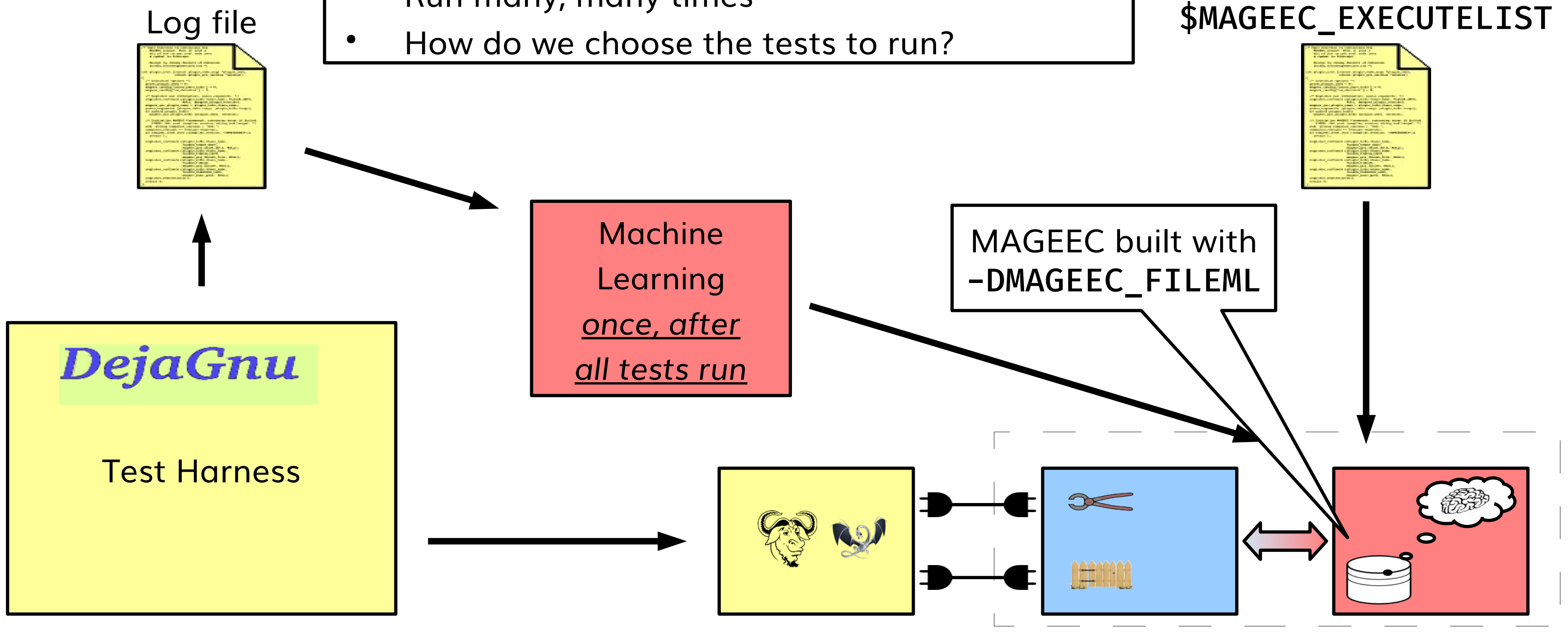
The MAGEEC Approach



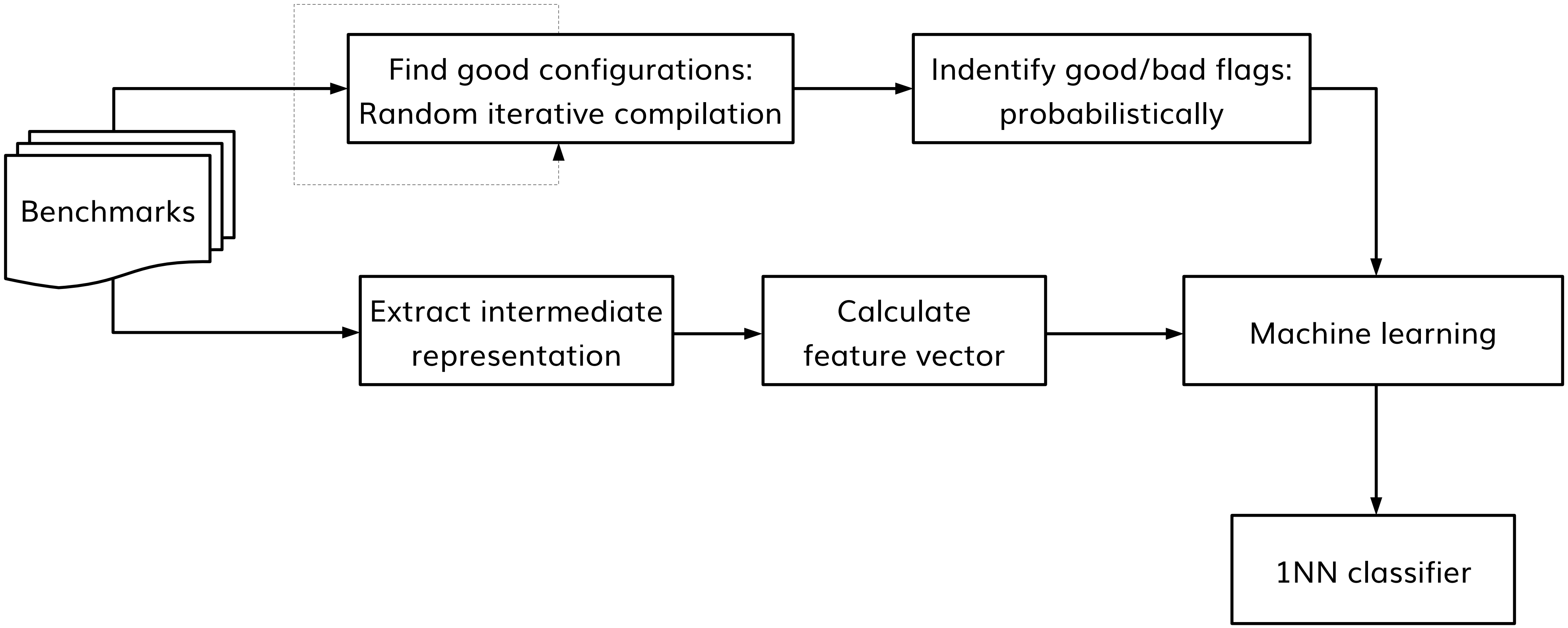
- *Identifying Compiler Options to Minimise Energy Consumption for Embedded Platforms.* James Pallister, Simon Hollis, Jeremy Bennett, https://arxiv.org/abs/1303.6485.
- Successor to MILEPOST, followed by TSERO

MAGEEC: Building the Database

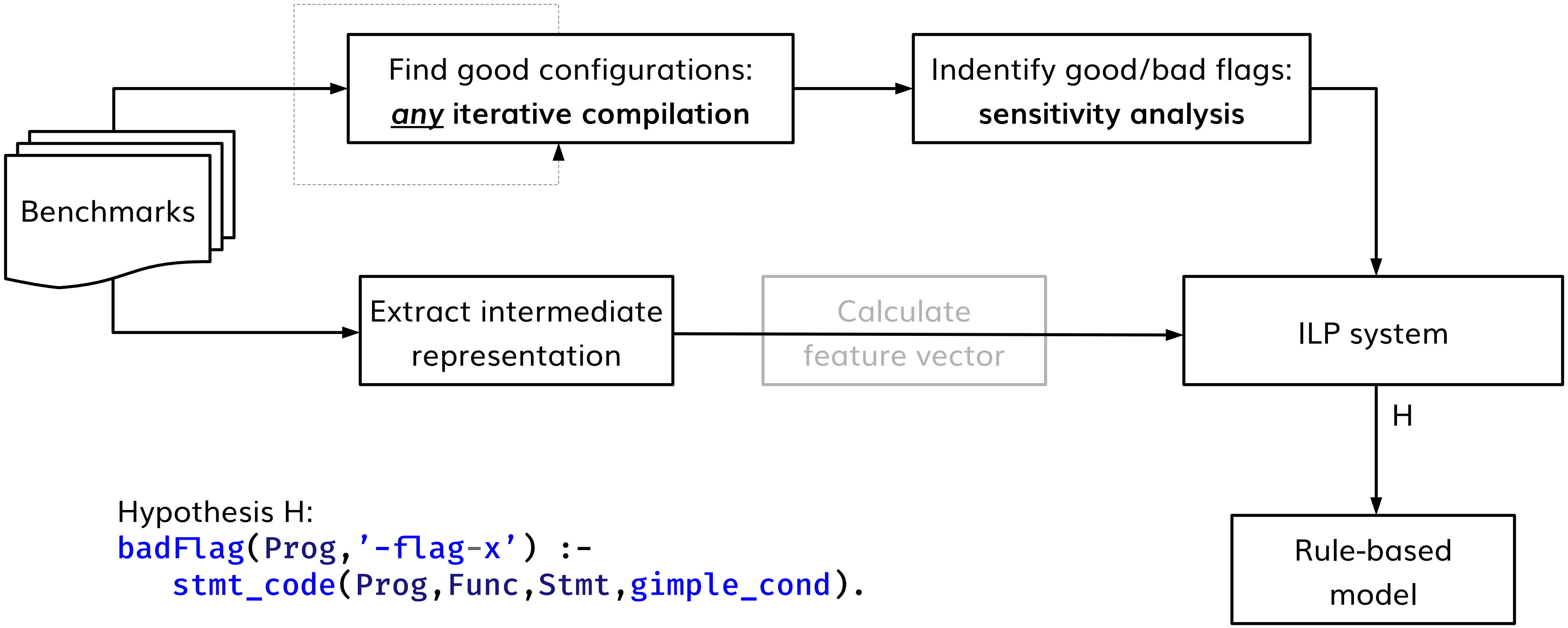
- Run many, many times
- How do we choose the tests to run?


Log file
DejaGnu
Test Harness
Machine Learning *once, after all tests run*
MAGEEC built with -DMAGEEC_FILEML
$MAGEEC_EXECUTELIST

MILEPOST/MAGEEC/TSERO in Summary


Benchmarks
Find good configurations: Random iterative compilation
Indentify good/bad flags: probabilistically
Extract intermediate representation
Calculate feature vector
Machine learning
1NN classifier

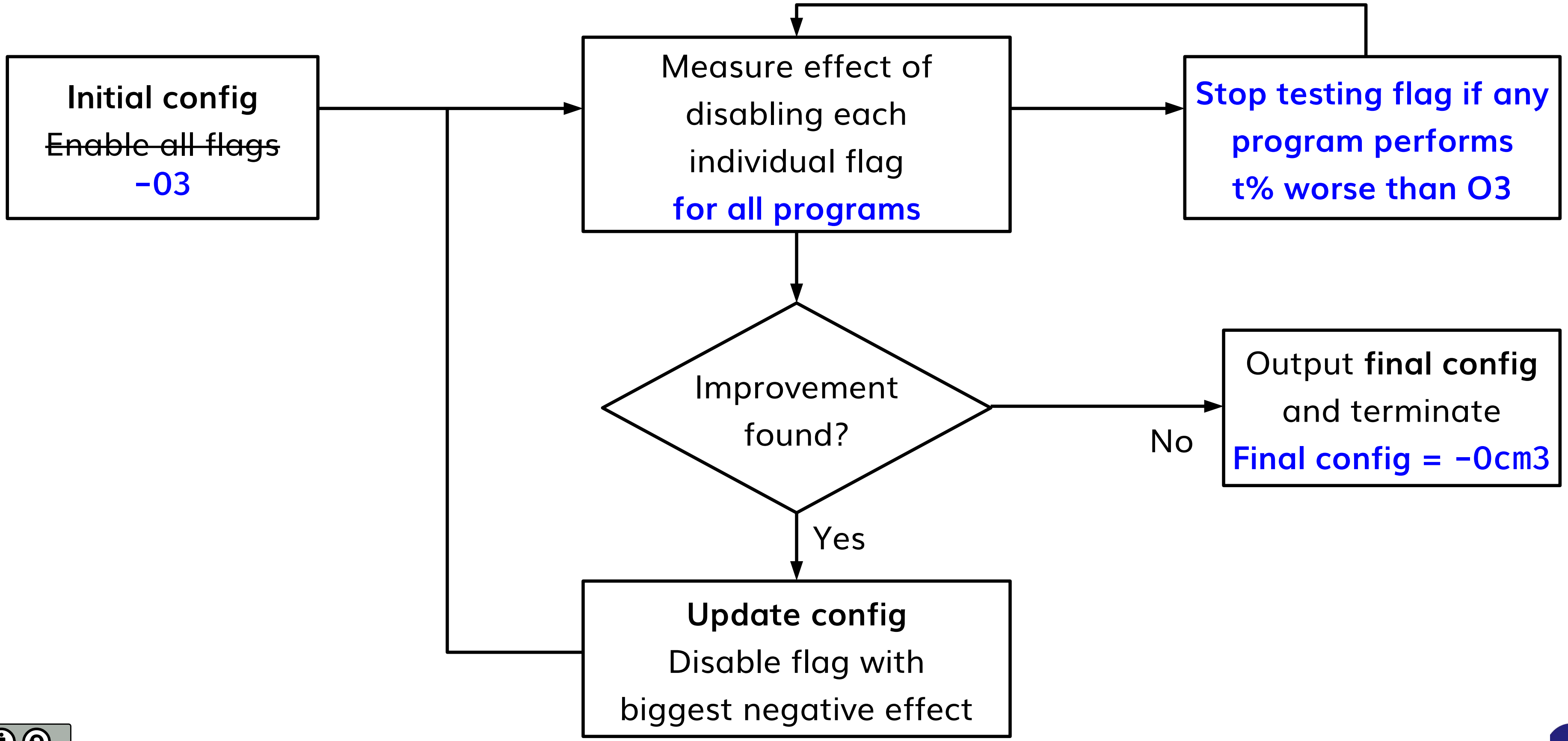
Craig Blackmore's ILP Method


Benchmarks
Find good configurations:
any **iterative compilation**
Indentify good/bad flags:
sensitivity analysis
Extract intermediate representation
Calculate feature vector
ILP system
H
Rule-based model

Hypothesis H:

```
badFlag(Prog,'-flag-x') :-
   stmt_code(Prog,Func,Stmt,gimple_cond).
```

Using CE To Create New -O Flags

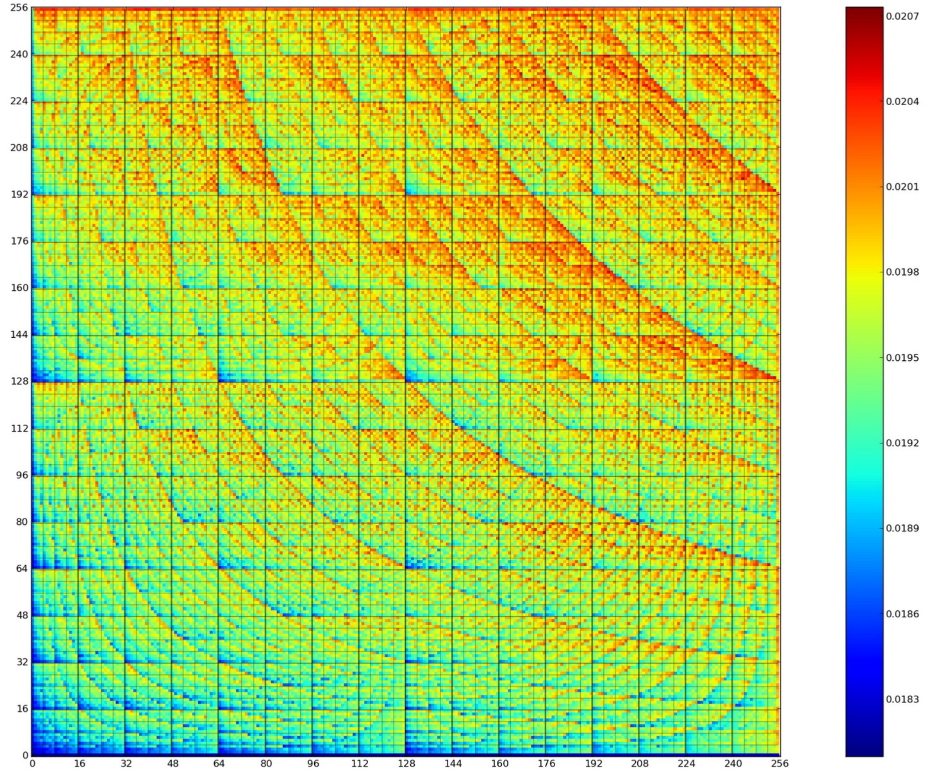

Initial config
~~Enable all flags~~
-O3
Measure effect of disabling each individual flag for all programs
Stop testing flag if any program performs t% worse than O3
Improvement found?
No
Output final config and terminate
Final config = -Ocm3
Yes
Update config
Disable flag with biggest negative effect

Energy Specific Optimizations

- Today optimizations are for code speed or code size.
- You can write optimizations for energy
- Align hot loops in bit lines of flash (up to 15% energy saving)
  - *A high-level model of embedded flash energy consumption.* James Pallister, Kerstin Eder, Simon J. Hollis, Jeremy Bennett, https://arxiv.org/abs/1404.1602
- Move hot loops from flash into RAM (up to 22% energy saving)
  - Optimizing the flash-RAM energy trade-off in deeply embedded systems. James Pallister, Kerstin Eder, Simon Hollis, https://arxiv.org/abs/1406.0403

EMBECOSM®

Surprising Things You Learn



EMBECOSM®

Thank You

jeremy.bennett@embecosm.com

embecosm.com

Jeremy Bennett