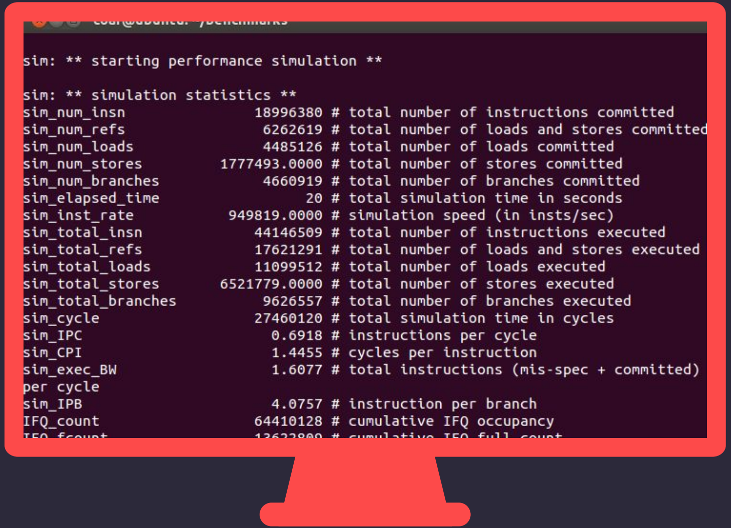
Experimental Evaluation WCET-Aware Hybrid Global Branch Prediction Approach

< COSC 4331/6384 Justine Tran >

{ Research Goal

Evaluation Xuesong Su, Hui Wu and Qing Yang claim that the use of Hybrid Global Branch Prediction Approach reduces the WCET, through the use of the SimpleScalar simulation tool



```
sim: ** starting performance simulation **

sim: ** simulation statistics **
sim_num_insn                 18996380 # total number of instructions committed
sim_num_refs                  6262619 # total number of loads and stores committed
sim_num_loads                 4485126 # total number of loads committed
sim_num_stores           1777493.0000 # total number of stores committed
sim_num_branches              4660919 # total number of branches committed
sim_elapsed_time                   20 # total simulation time in seconds
sim_inst_rate             949819.0000 # simulation speed (in insts/sec)
sim_total_insn               44146509 # total number of instructions executed
sim_total_refs               17621291 # total number of loads and stores executed
sim_total_loads              11099512 # total number of loads executed
sim_total_stores         6521779.0000 # total number of stores executed
sim_total_branches            9626557 # total number of branches executed
sim_cycle                    27460120 # total simulation time in cycles
sim_IPC                        0.6918 # instructions per cycle
sim_CPI                        1.4455 # cycles per instruction
sim_exec_BW                    1.6077 # total instructions (mis-spec + committed)
per cycle
sim_IPB                        4.0757 # instruction per branch
IFQ_count                    64410128 # cumulative IFQ occupancy
IFQ_fcount                   13622809 # cumulative IFQ full count
```

}

Background Information

This presentation is based on the following Thesis

Experimental Evaluation of Code Properties for WCET Analysis

Colin, A., & Petters, S. M. (2003). Experimental Evaluation of Code Properties for WCET Analysis. In Proceedings of the 24th IEEE International Real-Time Systems Symposium (RTSS'03) (pp. 135-144). IEEE. doi: 10.1109/REAL.2003.1253268

An Efficient WCET-Aware Hybrid Global Branch Prediction Approach

Su, X., Wu, H., & Yang, Q. (2016). An Efficient WCET-Aware Hybrid Global Branch Prediction Approach. In 2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (pp. 1-8). IEEE. doi: 10.1109/RTCSA.2016.7579929

Table of contents

01 Introduction

Discussion on branch prediction and its effect on Worst-Case Execution Time

02 Hybrid Global Branch Prediction Approach

Explanation of research thesis

03 Application of Simple Scalar

Experimental Results

01 { ..

Introduction

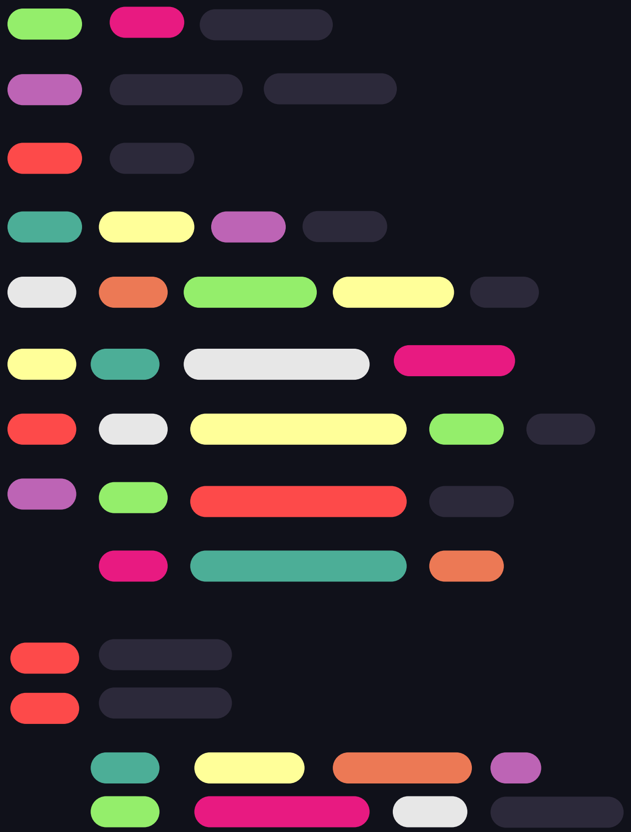
< Discussion on branch prediction and its effect on Worst-Case Execution Time >

} ..

Introduction

Embedded processors are equipped with a variety of features to enhance the average performance.

- Pipelines - increase the performance by overlapping the execution instructions.
- Cache - stores frequently used data or instructions
- Branch Predictions - determines outcome of branch in order to continue issuing instructions down the pipeline



Branch Predictions

< Branch Predictions creation

- Prevent degrading performance of pipelines

Correct Prediction

- execution of program proceeds without interruption

Misprediction

- Undo instructions, fetch new instruction made by branch

New Issue: Mispredictions is the single most significant performance limiter for improving processor performance >

Introduction

In real-time embedded systems, determining the WCET of each task is the key to finding a feasible schedule for a set of tasks.

- Branch Predictions have a significant impact on the WCET of a task.

However, most branch prediction approaches today (2016) is not suitable for reducing the WCET of a task.

02 { ..

Hybrid Global Branch Prediction Approach

< Discussion of Thesis and Application of Research >

} ..

Hybrid Global Branch Prediction Approach

Xuesong Su, Hui Wu and Qing Yang thesis suggest that in order to reduce the number of branch mispredictions for a task such that the WCET is minimized, they have developed a Branch correlation-branch, Hybrid Global Branch Prediction.

Uses:

- Correlation-based analysis
- Static WCET analysis

Outcome: Identify the branches that do not have any impact on the WCET of the task. And focus on predicting the outcome of most critical branches

Hybrid Global Branch Prediction Approach

Static Prediction Approach

All branches prediction are performed by the compiler at compile time

Dynamic Branch Prediction

Made by hardware during the execution time

Hybrid Global Branch Prediction Approach

Static Prediction Approach

All branches prediction are performed by the compiler at compile time

Dynamic Branch Prediction

Made by hardware during the execution time

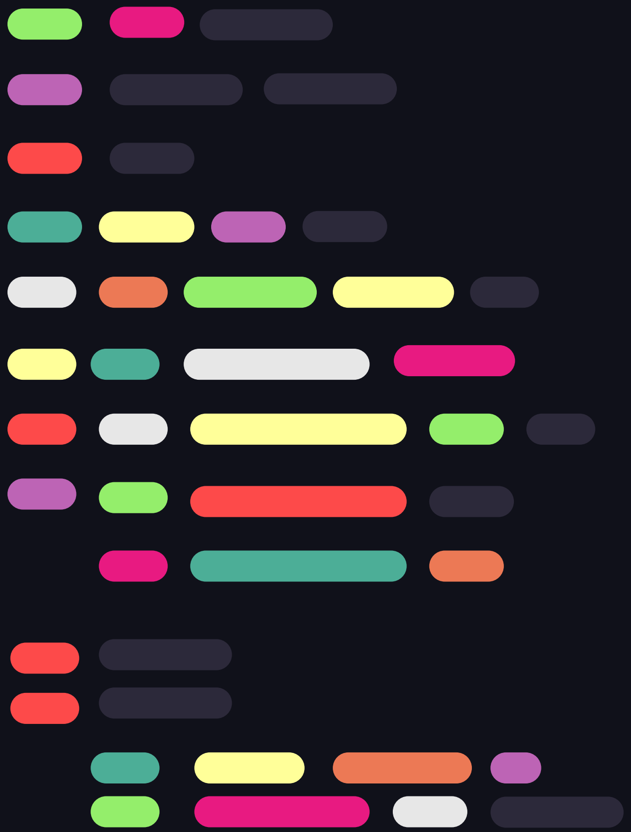
Hybrid Global Branch Prediction Approach

Determining which branches have impact on WCET

WCET-aware graph reduction to find all branches, that have no impact on WCET of program.

1) Construct Weighted Control Flow Graph (CFG)
2) Convert to Directed Acyclic graph (DAG)

Producing Set A, which contains basic blocks that no impact on WCET



Static WCET analysis

< Done with the basic blocks not in set A
In experiment three disjoint groups are created:

1) Statically predicted branches
2) Dynamically Predicted branches using local prediction scheme
3) Dynamically predicted branches using correlations between branches

Calculations:

- Static: predict outcome store in its predicted outcome field

- Correlated: take static outcome and place it with branch correlation tuple
- Local: Majority-Win Strategy >

Hybrid Global Branch Prediction Approach

Experimental Results

Hybrid Global Branch Prediction outperformed static and dynamic branch prediction.

- With a WCET improvement to the tight estimate of the WCET

03 { ..

Application of Simple Scalar

< Implementing Hybrid Global Branch Prediction in SImpleScalar to prove Xuesong Su, Hui Wu and Qing Yang work >

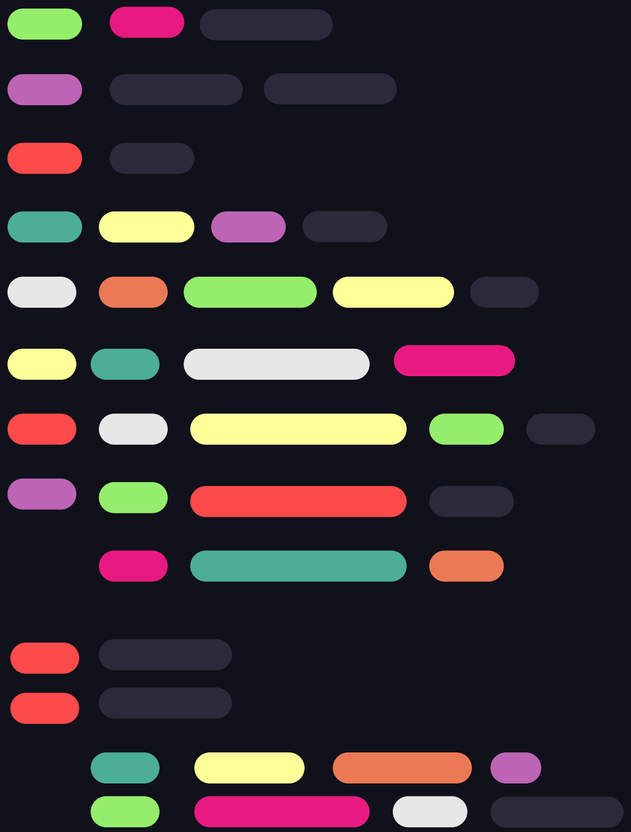
} ..

Application of Simple Scalar

Experimental Evaluation of Code Properties for WCET Analysis

- Used to evaluate the impact that advance features placed of embedded processors can lead to overestimation factors.

Taking their experimental design and applying it to determine if Xuesong Su, Hui Wu and Qing Yang approach will minimize the WCET.



< Concerns:

Although it is very difficult to match the timing to a real-processor the goal of our research is to determine validate the experiment.

Approach:

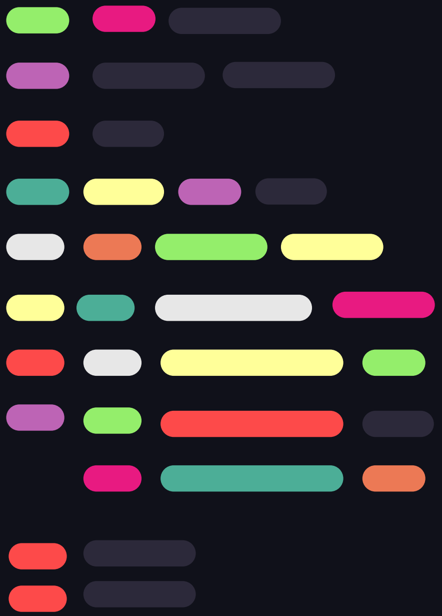
Simple Approach for WCET analysis.

- Determining the absolute maximum execution time for each basic block (single integer)

(last instruction complete)-(first instance of basic block)

- given proper testing of hardware features without having to model them >

Application of Simple Scalar



Use of Simple Scalar

- Allowance of configuration of processor
  - Ex: Branch Prediction (type and size)
  - Trace holds the state of the pipeline at each cycle
  - Ability: Determine each path the processor program follows at what time instant

Application of Simple Scalar

Determining WCET Branch Predictions

Problem: Miss-predictions

- Measurement invalid

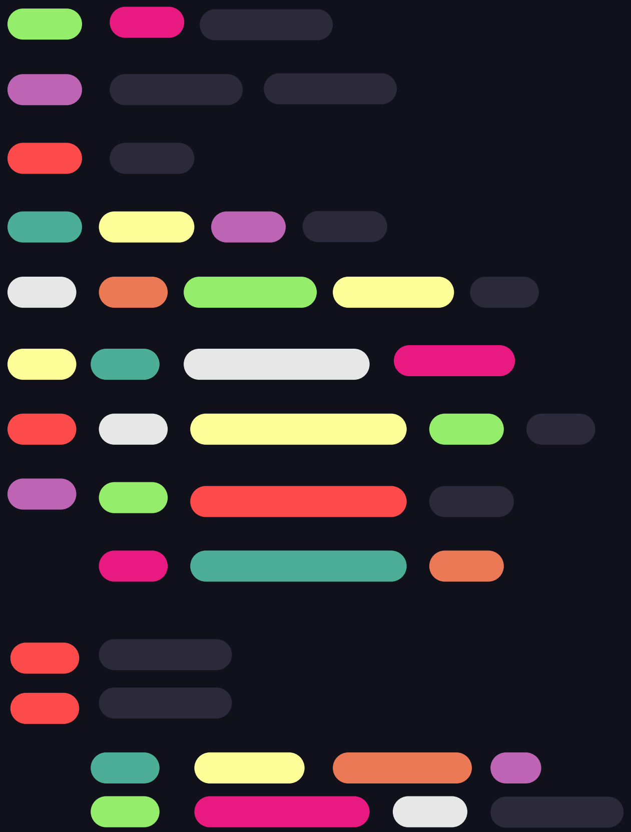
Solution WCET of basic block:

OLD: (first instance of any successor basic block)-(first instance of basic block)

NEW: (first instance of correct branch)-(first instance of basic block)

REMOVE:

- Trace of any miss-prediction instruction before analysis process



< Execution of Code in SimpleScalar

- Execute code between 500 to 1000 execution

For:

- Hybrid Global Branch Prediction
- Dynamic Prediction

Executing the code multiple times will reduce:

- Cache Misses
- Producing Regular Cache Pattern, reducing miss-predictions >

Application of Simple Scalar

Tight Estimate

This approach does not produce a tight estimate, but we are only looking to determine if Xuesong Su, Hui Wu and Qing Yang claims are correct.

Instead we seek a high-level analysis that can result in a safe estimate on the WCET

Application of Simple Scalar

Experiment Setup:

Code: Bubble Sort (C/C++)

- To determine WCET in Simplescalar

Goal: Obtaining WCET of Dynamic branch prediction and Hybrid Global Branch Prediction to determine if HGBP outperformed Dynamic Branch Prediction

```
{ cout << “ Thank You! ”;
  cout << “ Questions? “;
                                      }
```