RISE:2015
RESEARCH, INNOVATION AND SCHOLARSHIP EXPO
Category: Engineering and Technology
Degree Level: Graduate
Abstract ID#: 937

Northeastern University

System Level Performance Estimation using Calibrated Weight Table Approach

Authors: Kasra Moazzemi, Smit Patel and Gunar Schirner

Contact: Smit Patel – patel.smi@husky.neu.edu - R306, 140 The Fenway, Northeastern University, Boston, MA 02115

I. Abstract

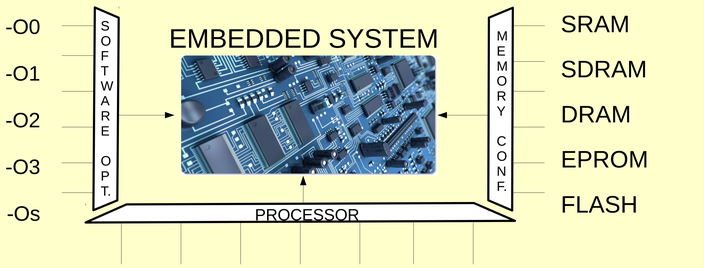
In Electronic System Level Design, it is essential to gauge the design decision at each abstraction level. With an exploding Design Space, performance estimation at highest level is a critical step. It must be capable of evaluating large number of design alternatives rapidly without compromising fidelity.

The proposed approach is based on populating the weight table of each Processing Element (PE) in the Design Space with automatic calibration and Linear Programming Formulation. It is implemented on 2 PEs, 4 software optimizations and 2 hardware configurations. This technique estimates with 24% average error, 90% fidelity and takes less than a second to complete. The absolute error is large because it operates at Source-Level and is retargetable and extremely fast to evaluate the large Design Space with high fidelity rate.

II. Introduction

Motivation

- Choice of Processing Element (PE), software optimization and hardware configuration is a crucial high level design decision.
- An exploding Design Space and timing constraints on system design adds challenges in this design decision.



Goal

- Source level performance estimation methodology to quickly explore the Design Space.
- Fidelity is more important than absolute accuracy of the estimation technique.
- Insights about the methodology :
  - Operates at Source level (C/C++)
  - Uniform for all targets (Highly Re-targetable)
  - Extremely fast
  - Deterministic estimation time

III. Approach – Rapid Retargetable Performance Estimation

- Each training benchmark is profiled at Source Level.

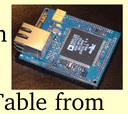
```
void main() {
    int a=1; float b=2.5;
    for(int i=0;i<10;i++)
        a=a+7;
    for(i=0;i<12;i++) {
        b=a*b;
        a=a+b; }
a=b*a }
```

Profiler

	bool	int	float	...
add	0	10	12	
multiply	0	0	13	
branch	0	24	0	
...				

The numbers denote occurrence frequency of each operation of every data type.

- The execution time (delay) of each benchmark is measured on Hardware, in terms of CPU Cycles.



- Linear Programming Formulation (LPF) generates a Weight Table from profile data and delay of training benchmarks.

How LPF works?

Minimize: $\sum_i Calibration\text{-}fudge(i)$

Subject to: B_1: $R1_1 . W_1 + R1_2 . W_2 + \ldots + R1_M . W_M + Calibration\text{-}fudge_1 = Delay(B_1)$

...

B_N: $RN_1 . W_1 + RN_2 . W_2 + \ldots + RN_M . W_M + Calibration\text{-}fudge_N = Delay(B_N)$

Goal: *Extract every W_i with minimum estimation error.*

W_i are the entries in the Weight Table

R is occurrence frequency appearing in the Application Profile

- Weight Tables are generated for all elements of the Design Space.
- **Performance of each PE is estimated by multiplying it's Weight Table with the Application Profile.**

	bool	int	float	...
add	54	18	9	
multiply	108	99	0	
branch	2	9	18	
...				

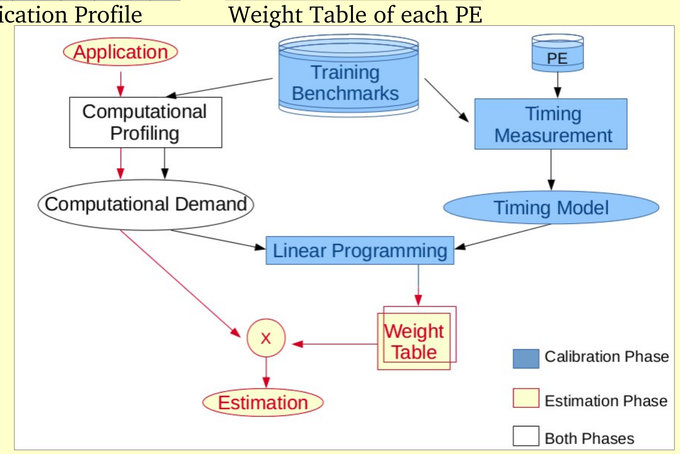
Application Profile

.*

	bool	int	float	...
add	1	1	6.5	
multiply	0.8	7	63	
branch	0.4	1.8	4	
...				

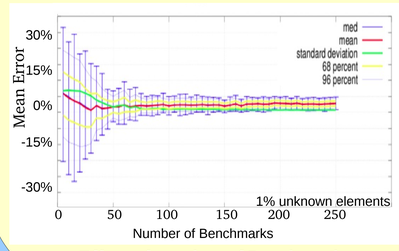
Weight Table of each PE

= Estimated Cycles

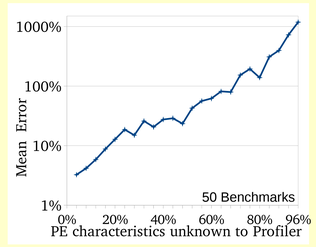


IV. Synthetic Model

- Used to validate the approach and analyze elements affecting the estimation accuracy.
- Incorporates synthetic benchmarks and synthetic PEs along-with real framework.
- Following factors affect the estimation accuracy :
  - Number of benchmarks used for calibration
  - Processor characteristics unknown to the profiler
  - Ability of LPF to derive the weight tables



Effect of number of Benchmarks

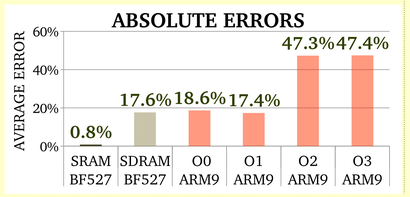


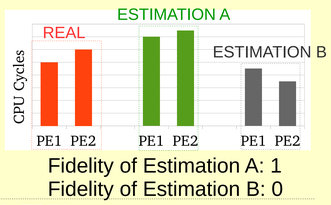
Effect of unknown elements

V. Results

Experiment Platform

- 48 benchmarks for calibration
- 4 SW optimizations (O0,O1,O2,O3)
- 2 PEs : BF-527 and ARM-9
- 2 HW configurations (SRAM,SDRAM)





Fidelity of Estimation A: 1
Fidelity of Estimation B: 0

FIDELITY MATRIX

	BF527 SRAM	BF527 SDRAM	ARM9 O0	ARM9 O1	ARM9 O2	ARM9 O3
BF527 SRAM	100%	90%	59%	76%	80%	76%
BF527 SDRAM	-	100%	69%	63%	67%	63%
ARM9 O0	-	-	100%	94%	96%	96%
ARM9 O1	-	-	-	100%	57%	61%
ARM9 O2	-	-	-	-	100%	55%
ARM9 O3	-	-	-	-	-	100%

- Developed Calibrative Performance Estimation framework which exposes compiler optimizations and memory hierarchy and automates Weight Table generation.
- Limitations include being agnostic to processor characteristics and dependency of estimation accuracy on training benchmarks.