Simulation of CPUID of Intel Architecture

Grigory Rechistov* Name Surname†

March 14, 2014

Contents

1 Introduction

TODO Write me

1.1 Contributions

In this paper we make the following contributions.

1. Evaluate and compare existing means of processor features identification of different architectures.
2. Describe, implement and evaluate a structured solution to the simulation of CPUID instruction of Intel IA-32.

*Moscow Institute of Physics and Technology, grigory.rechistov@phystech.edu
†email@mail.com

2 Overview of Processor Identification

2.1 MIPS

2.2 ARM

2.3 PowerPC

2.4 Intel IA-64 (Itanium)

[1]

2.5 Intel IA-32 and Intel 64

The common PC architecture, starting from Intel Pentium and its clones, provides `CPUID` [2] instruction. Since its inception it has been extended numerous number of times.
There is a number of complications that have resulted from long uncontrolled expansion of the CPUID.

Elements adressing To inspect a value of a particular

- Leaves
- Subleaves
- Registers
- Bit range

Non-constant values Firmware is able to suppress certain features indicated by CPUID by manipulating bits of model specific register (MSR) IA32_MISC_ENABLE. For example: TODO NX, Leaf3, 1GB pages

Topology-varaible elements Finally, it should be noted that, besides EAX, EBX, ECX, EDX, one more register may be affected by CPUID, namely IA32_SIGNATURE TODO .

3 Existing Approaches to CPUID Simulation

What is required from a CPUID model.

- Be accurate TODO
- Be configurable TODO

3.1 Bochs

3.2 Xen

3.3 Qemu

3.4 Simics

4 The Structured Approach

The described approach

1.
2.
3.

It has the following advantages

- Uses natural unit of configuration state — a field.

- In the meantime, it allows its users to operate in terms of 32 bit leaves values which are more convenient and compact in many cases.
-

5 Evaluation

6 Conclusions

7 Acknowledgements

References

[1] Intel Corporation, *Intel® Itanium® Architecture Software Developer's Manual*, 2010.

[2] Intel Corporation, *Intel® 64 and IA-32 Architectures Software Developer's Manual. Volumes 1–3*, 2012.