Measuring instruction latencies with llvm

Guillaume Chatelet
C. Courbet, B. De Backer, O. Sykora

Google Compiler Research
Why?

- Scheduling needs latencies and μOp decomposition
  - This talk is about latency measurement only
- Vendors release some information
  - May be incomplete / not be in a machine readable format
- Updating LLVM td files
  - is tedious / requires careful guesswork and analysis.
- Consequences
  - scheduling information is incomplete for most X86 models
How it works

∀ processor, ∀ instruction:

start_measure
.rept 10000
.add rax, rax
.endr
.end_measure
How it works - actually subtler than this...

∀ processor, ∀ instruction:

\[
\text{start\_measure} \\
\text{.rept 10000} \\
\text{andn eax, ebx, edx} \ # \text{processor can execute these in parallel} \\
\text{.endr} \\
\text{end\_measure}
\]

- We need a way to make the execution **sequential**
MCInst in LLVM

Explicit inputs (e.g. GR16)

Implicit input (e.g. EFLAGS)

Implicit output

Explicit output

<table>
<thead>
<tr>
<th>USE</th>
<th>USE</th>
<th>IUSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPCODE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEF</td>
<td>IDEF</td>
<td></td>
</tr>
</tbody>
</table>
Sequential execution: Create Dependency

Current instruction must use an output of previous instruction
Implicit self cycle

Possible cycle:

Possible instance: AAA
Implicit self cycle - through register aliasing

Possible cycle:

Possible instance:
AAA
Possible explicit self cycle

Possible cycle:

```
GR32 | Imm
-----|-----
AND32ri
-----|-----
GR32 | EFLAGS
```

Possible instance:
```
AND32ri EAX, EAX, 1
```
Possible cycle through another instruction

Possible cycle:

Possible instance:

```
MMX_PMOVMSKBrr R10D, MM1
MMX_MOVD64rr MM1, R10D
```
Possible cycle through another instruction

Possible cycle:
- VCMPPSZ256rri
- VFMSUBADD213PDZrk

Possible instance:
- VCMPPSZ256rri
  - K5, YMM31, YMM31, 1
- VFMSUBADD213PDZrk
  - ZMM31, ZMM25, K5, ZMM29, ZMM9

Keep in mind:
This process is fully automated
> llvm-exegesis -opcode-name IMUL16rri8 -benchmark-mode latency
---
asm_template:
  name: latency IMUL16rri8
  cpu_name: sandybridge
  llvm_triple: x86_64-grtev4-linux-gnu
  num_repetitions: 10000
  measurements:
    - { key: latency, value: 4.0115, debug_string: '' }
  error: ''
...

- Identified discrepancies between TD files and measurements
What's next?

- Extend to memory operands
- Automate fixing of TD files
- Measure the effect of
  - immediate: ±0, 1, ~1, $2^{8,16,32,64}$, ±∞, nan, denorm
  - register values: SUB EAX, EAX, EAX vs SUB EAX, EAX, EBX
- Make it work on other CPUs (ARM under way, Power?)
Try It Out!