Basics of CodeGen (O-o-O vs. in-order part)

Toshihiro KONDA

Background

- **There are basically two types of hardware in the world**
  - **in-order**
    - **Execution order guarantee**
  - **O-o-O (Out of Order)**
    - **Sort the instruction to the extent that the result matches**
  - **(Other)**
    - **Mixing of both**

When to generate code…

- What is definitely different between in-order and O-o-O.
  - in-order
    - Software guarantees instruction level parallelism (ILP)
  - O-o-O
    - Hardware cooperates with software to exploit ILP

In particular・・・

- For O-o-O, it cuts the live range as much as possible
  - Register pressure will be higher, but it is not a problem.

  - Why?
    - Register pressure is high = Originally ILP is high.
    - Hardware guarantees efficient execution of high ILP code.
  - However, there is a limit, of course, and it is mainly determined depending on the number of entries of Reorder Buffer (ROB) etc.

in-order case

- We would like to use what we defined after as much as possible (software viewpoint)
- We guarantee what we had secured with O-o-O case with software.
- As a result, code generation is quite different in O-o-O and in-order.

Conclusion

- The compiler has to code generation depending on the hardware configuration (an example is shown).

Thanks.