RSVP: A hybrid model of Register Sharing and Value Prediction

Kleovoulos Kalaitzidis, INRIA
André Seznec, INRIA
Value Prediction [Lipasti&al.96][Mendelson&al.97]

- Breaks true data dependencies to extract more ILP:
  - The result of an instruction is predicted before scheduled execution.
  - Dependent instructions (*consumers*) can speculatively execute ahead of their *producer*.

```
I1 → I2 → I3 → I4
```
Value Prediction [Lipasti&al.96][Mendelson&al.97]

- Breaks true data dependencies to extract more ILP:
  - The result of an instruction is predicted before scheduled execution.
  - Dependent instructions (consumers) can speculatively execute ahead of their producer.
Value Prediction schemes [Sazeides&al.97]

**Instruction-based VP schemes:**

- **Context-agnostic:**
  - **Last Value**: Pred = Result of last instance
  - **Stride**: Pred = Result of last instance + constant

- **Context-aware:**
  - **FCM**: Pred based on results of n recent instances
  - **VTAGE**: Control-flow based Last Value Predictor (use of global branch history)
  - **DVTAGE**: Hybrid VTAGE and Stride
Last Value Prediction

Predict:
- Inst PC
  - Last Value Table
  - Prediction
    - Last Value
    - Confidence
  - Saturated

Update:
- Confidence++
- Equal
- Reset/Update
- Committed Value
Last Value Prediction

Predict:

- Inst PC → Last Value Table
- Last Value Table → Prediction
- Prediction → Last Value
- Last Value → Confidence
- Confidence → Saturated

Update:

- Confidence++ → Equal
- Equal → Saturated
- Saturated → OoO Engine
- OoO Engine → Commit Value
- Commit Value → Equal
- Equal → Confidence++
- Confidence++ → Reset/Update
Last Value Prediction

Predict:

Instance \([n+p]\) to predict last committed value

Update:

Previous instance \([n+(p-1)]\) not committed yet
Proposal

*Combine Last Value Prediction with an Equality Predictor*

- **Equality Predictor** identifies equality between consecutive occurrences of the instruction.

- When equality fits in the instruction window, the two consecutive instructions **share the same destination register**.

- **Last Value Prediction** is selected when the previous instance of the instruction is not inflight.
Introducing Last Value Equality Prediction

- **Last Committed-Value Table:**
  - 3-way set associative (13bits tag)
  - 3-bit saturating confidence counter
  - 64-bit value

- **Equality Table:**
  - 3-bit saturating counter
  - 1-bit equality indicator

- **Result Buffer**: Circular-FIFO double size of the instruction window storing committed results.

- **Fully Associative Instruction Window:**
  - Entries associated with the instruction-PC
  - Keeps destination register index
Introducing Last Value Equality Prediction
Introducing Last Value Equality Prediction
Introducing Last Value Equality Prediction
Introducing Last Value Equality Prediction
Introducing Last Value Equality Prediction
Microarchitecture Design

- VP procedure [Perais & Seznec.14]:
  - Index Last Value Equality predictor at Fetch.
  - Consider only confident predictions.
  - Squash the pipeline on a misprediction or fault register sharing.
  - Validate **beyond the OoO engine**, at commit.
  - Update predictor using the committed value.

![Diagram of Microarchitecture Design]

- **Fetch**
- **Decode**
- **Rename**

- **Last Value Equality Predictor**
- **Validation + Squashing @commit**
- **Selector**

- **n-issue Out-of-order Engine**
  - **Fully Associative Instruction Window**
    - **ROB**
    - **IQ**
    - **PRF**
    - **FUs**

- **In-order Frontend**

- **PC**
What we expect

- Keep benefits of **Value Prediction**.

- Offload Value Prediction logic by exploiting equality between inflight instructions with **Register Sharing**.
  - A chain of inflight instances will have their result ready as soon as the oldest one has retrieved it
What we expect

- Keep benefits of **Value Prediction**.

- Offload Value Prediction logic by exploiting equality between inflight instructions with **Register Sharing**.
  - A chain of inflight instances will have their result ready as soon as the oldest one has retrieved it

> **Conclude to simplified speculation technique based on the trivial model of Last Value Prediction (not history-based a la TAGE).**
Thank you architects!