Lecture 18 (Trace Cache)

i-cache is more important than d-cache since instruction fetch happens in order, hence it is more performance sensitive.

1 Trace Cache

1. Basic blocks are defined - single point of entry and exit
2. Have a cache that can store such traces
3. If trace is accurate, prediction is not needed

1.1 Approach

1. Trace consists of multiple cache lines
2. Linked list of cache lines
3. We store the decoded micro ops

1.2 Design

1. Tag array
2. Data array
3. Controller
4. Fill buffer

1.2.1 Tag Array

1. Tag
2. Valid
3. Type
4. Next way
5. Prev way
6. NLIP - address of next CISC instruction
7. micro IP - index into the table of micro ops

1.3 Storage

1. Store trace segments in consecutive sets
2. The way number is stored in next set
3. Set # is used to determine the max size of linked list

1.4 Rules

1. Never distribute micro ops across cache lines
2. Terminate a data line if more branch micro ops than a threshold
3. Terminate trace if we encounter indirect branch
4. Interrupt or branch misprediction notification
5. Maximum length reached