Lecture 05 (GHR and stuff)

1 Global History Register

Maintain state of the last n branches

2 GAp Predictor

1. Index in the table using n bits of address and k bits of GHR
2. Pattern History Table (PHT) is used to store this information

Petition to make blackboards as smart boards instead of doing Leetcode

2.1 SharkTank Scam

1. Two IITD graduates in the panel were unable to catch a scam
2. Scam was related to "mid-brain activation" magic trick

2.2 Issue with GAp

1. Only stores information about last branch path
2. Need to do it for a region

3 PAp Predictor

1. Use $n1$ middle bits to select k bit GHR
2. Remaining is GAp

3.1 Better Solution

1. Instead of appending k bits for the index, mangle the bits (using xor or a better algo)
2. When using xor, predictor is called Gshare
3. Maintain multiple predictors, choose the better predictor
    i. can run both predictors
    ii. select when required

iii. wastes energy, reduces critical path length

4 Prediction and Compression

1. They are related
2. We cannot predict better than Fano's bound

5 Prediction of Branch Target

1. Use IST as Branch Target Buffer (BTB)
2. Return address can be maintained using a stack - 100% accuracy (Return Address Stack - RAS)

6 Decode Stage

6.1 CISC

1. Issue implementing OOO with CISC
2. Most CISC processors internally convert from CISC to RISC (micro OPs)
3. Some instructions map to large number of micro OPs, use microcode cache
4. Can be solved by pre-decoding the instructions when fetching into i-cache
    - 8 bits
    - start bit
    - end bit
    - functional bit
    - two-ROP
    - three-ROP

6.2 Optimizing Operations on Stack Pointer

1. Store stack pointer during decode stage
2. Directly compute memory address
3. No need to pipeline, can directly load or store
4. Need to nullify this optimisation when we have something like `ld sp 12[r1]`
    - but can store the offsets
    - introduce these offsets in future when value is loaded

6.3 Instruction Compression

1. Reduced-width instructions (thumb ISA)
    - avoid encoding complicated flags and options
    - reduce size of immediate fields

- implicit operands
- to overcome the issue of varied size instructions, have thumb instructions in groups such that total length is 4 (or 8) bytes