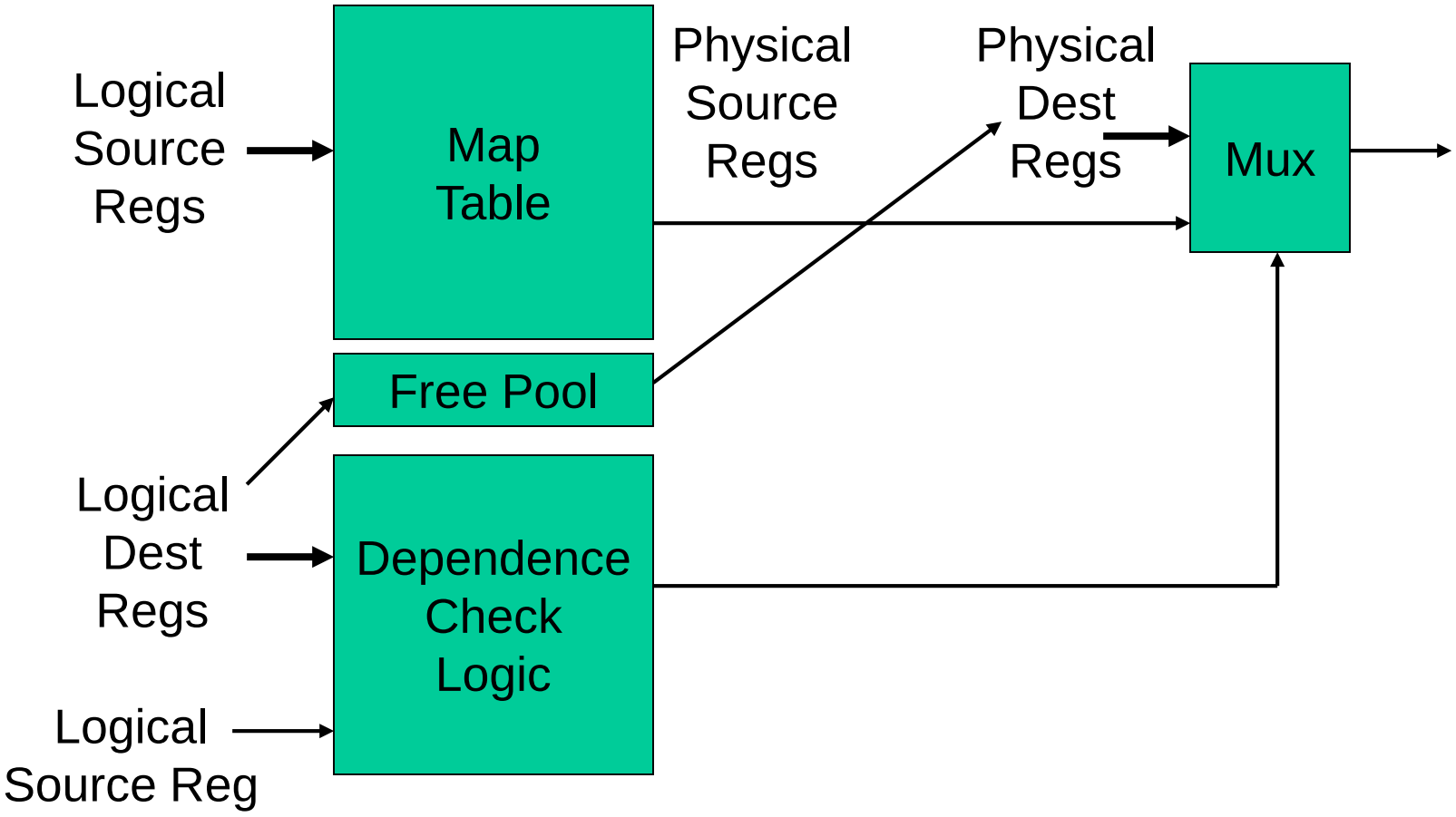
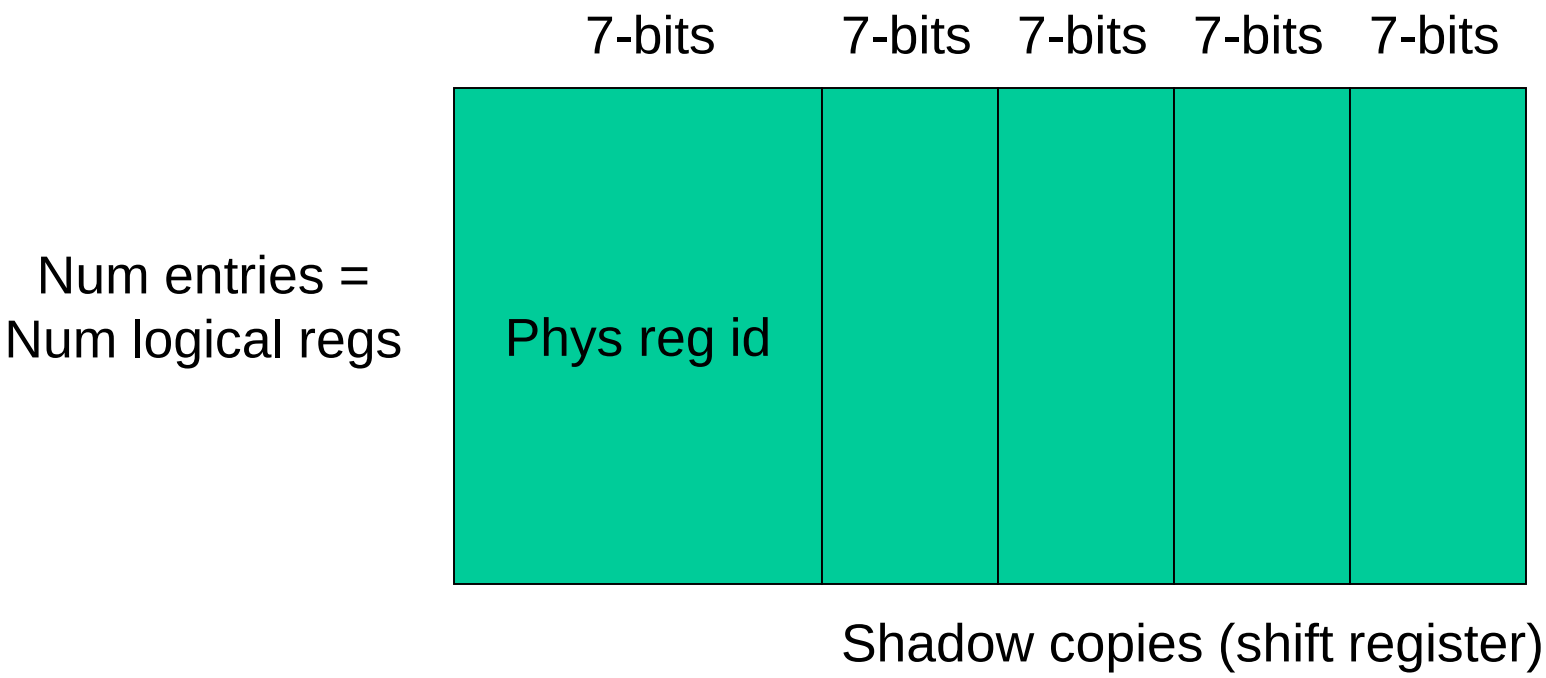
Lecture 17: Core Design

- Today: implementing core structures – rename, issue queue, bypass networks; innovations for high ILP and clock speed

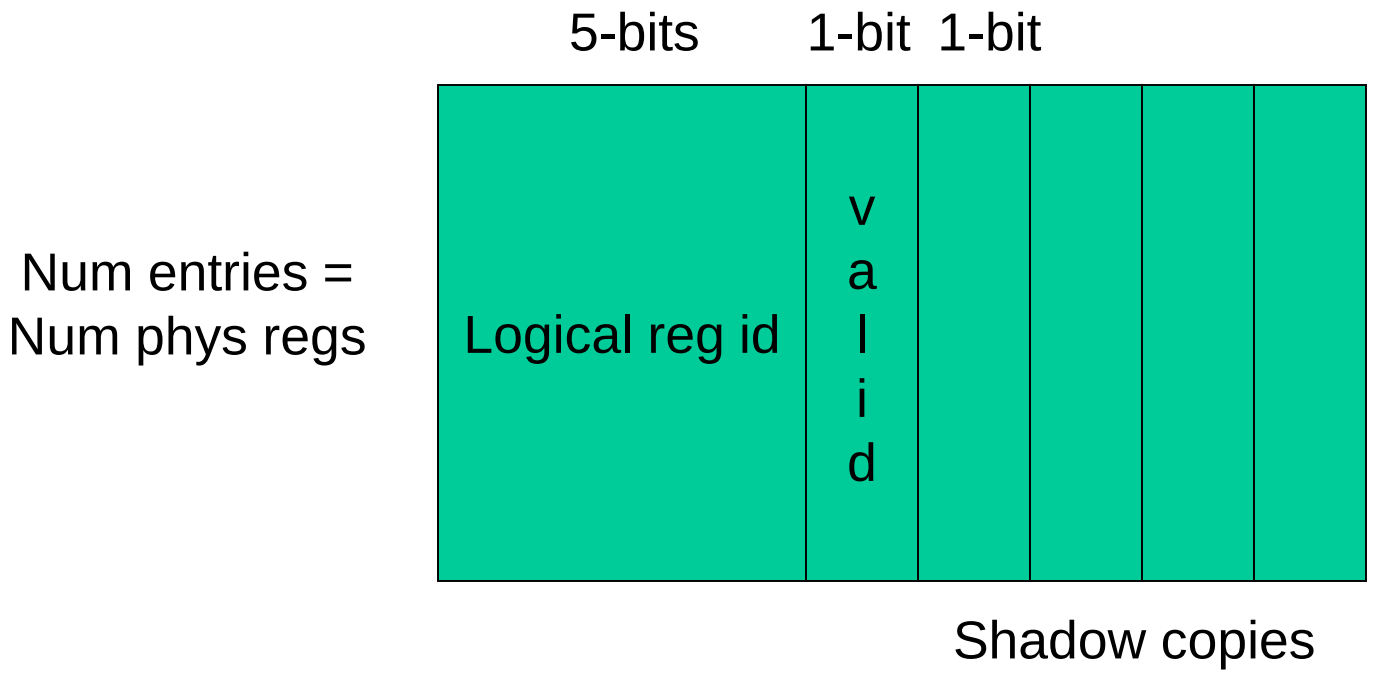
Register Rename Logic



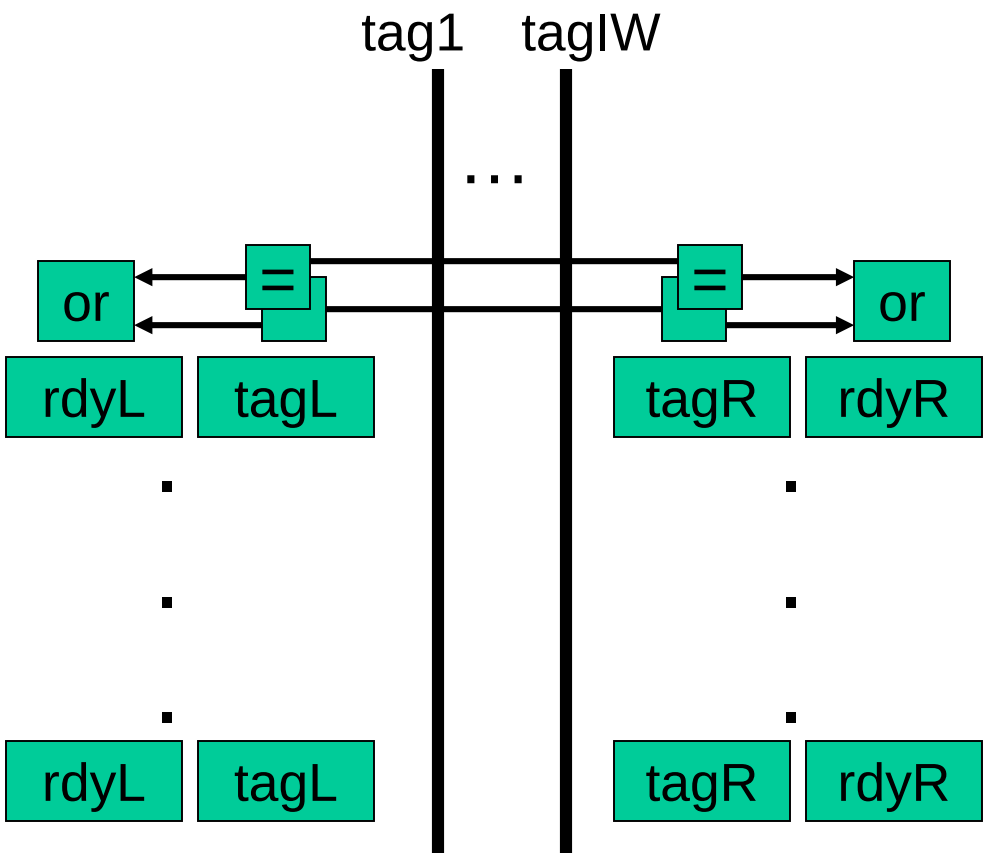
Map Table – RAM



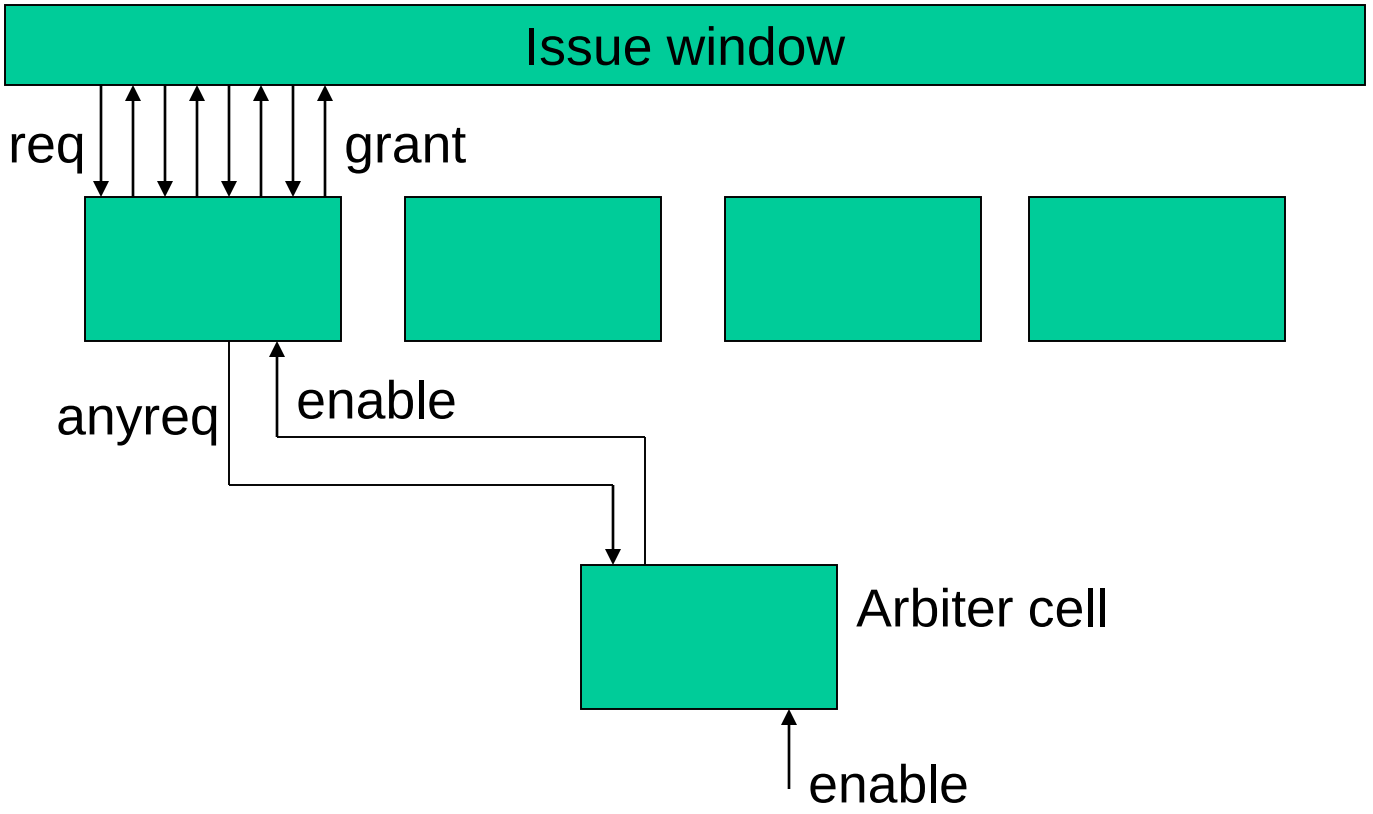
Map Table – CAM



Wakeup Logic



Selection Logic



- For multiple FUs, will need sequential selectors

Structure Complexities

- Critical structures:
  register map tables, issue queue, LSQ, register file, register bypass

- Cycle time is heavily influenced by:
  window size (physical register size), issue width (#FUs)

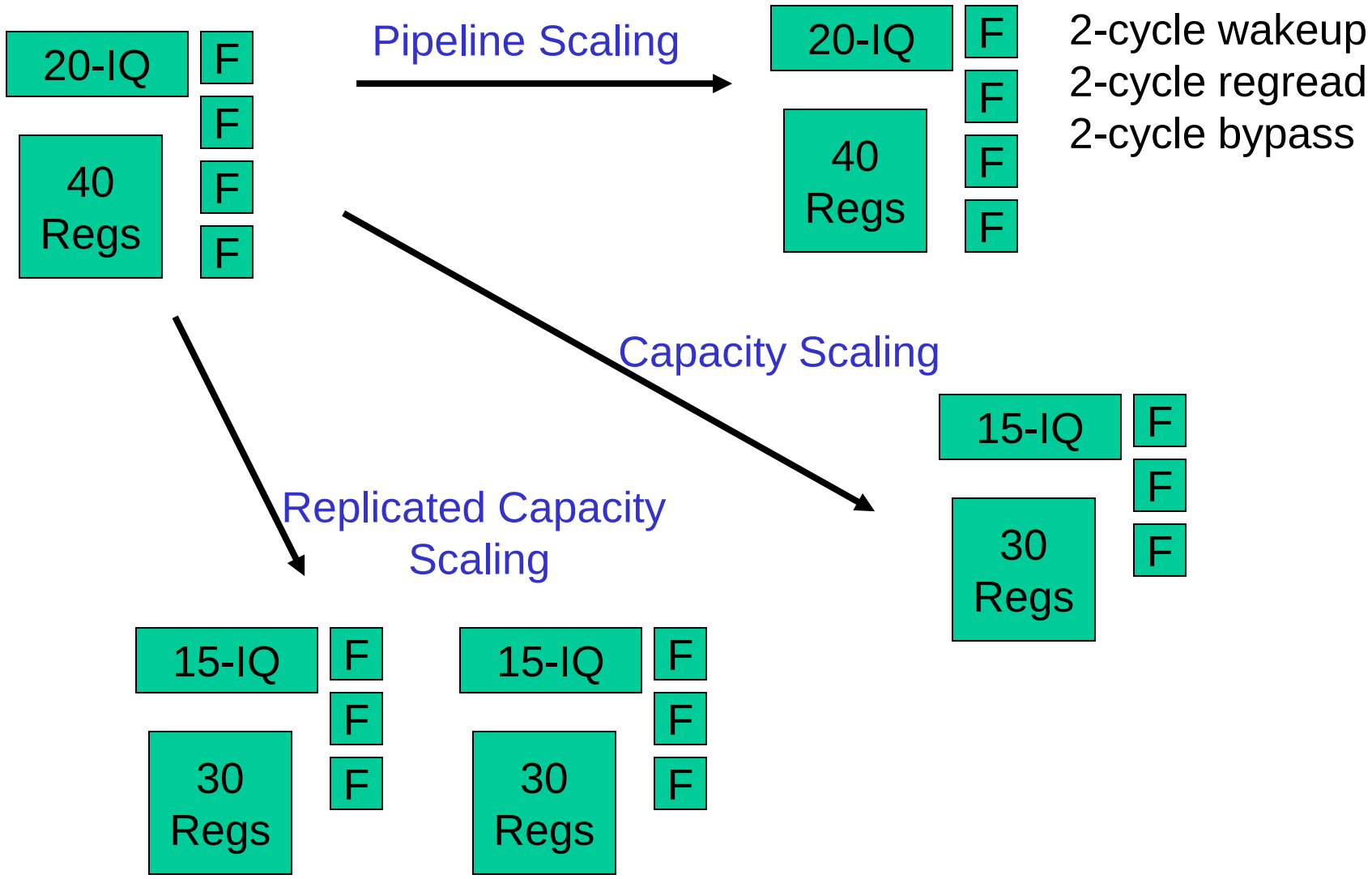
- Conflict between the desire to increase IPC and clock speed

- Can achieve both if we use large structures and deep pipelining; but, some structures can't be easily pipelined and long-latency structures can also hurt IPC

Deep Pipelines

- What does it mean to have
  - → 2-cycle wakeup
  - → 2-cycle bypass
  - → 2-cycle regread

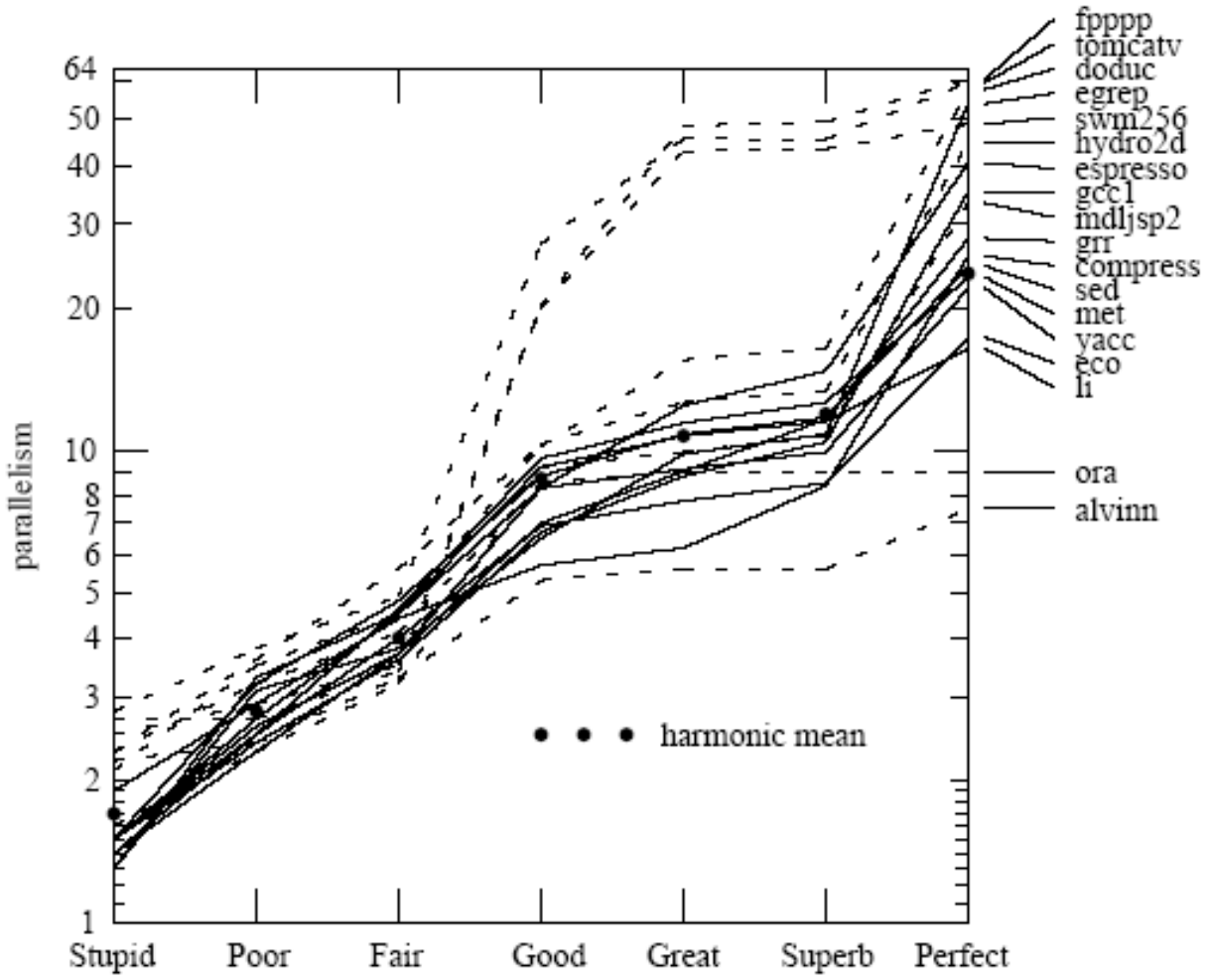
Frequency Scaling Options



Recent Trends

- Not much change in structure capacities
- Not much change in cycle time
- Pipeline depths have become shorter (circuit delays have reduced); this is good for energy efficiency
- Optimal performance is observed at about 50 pipeline stages (we are currently at ~20 stages for energy reasons)
- Deep pipelines improve parallelism (helps if there's ILP); Deep pipelines increase the gap between dependent instructions (hurts when there is little ILP)

ILP Limits Wall 1993

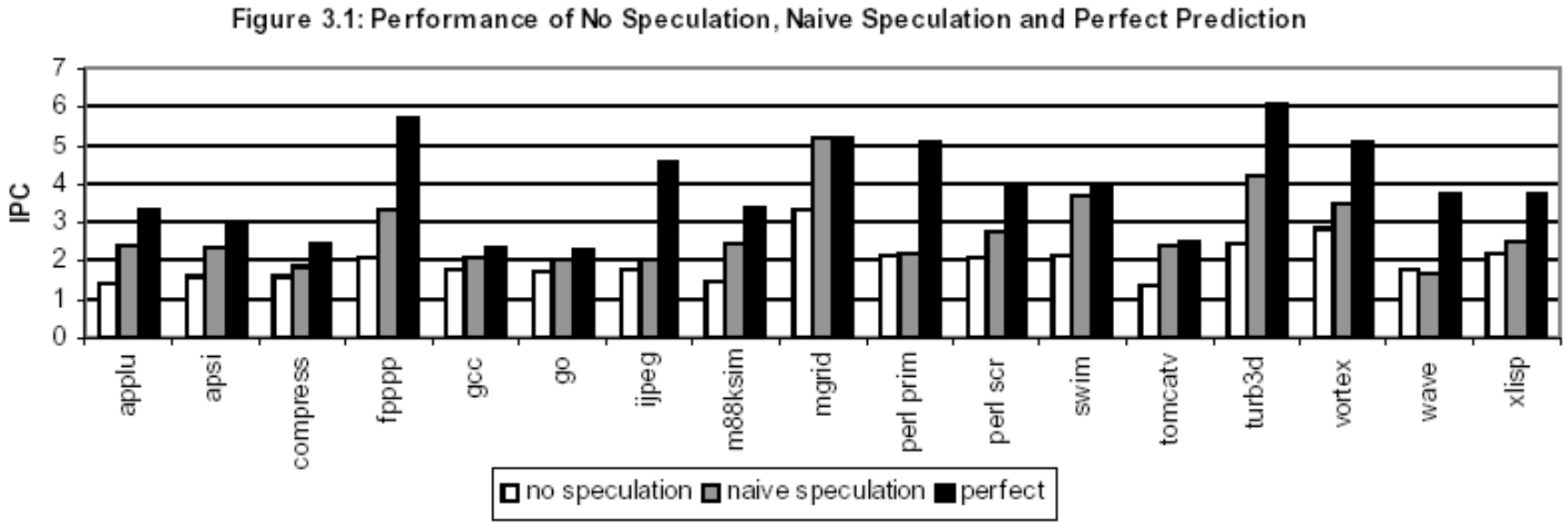


Techniques for High ILP

- Better branch prediction and fetch (trace cache)
  → cascading branch predictors?
- More physical registers, ROB, issue queue, LSQ
  → two-level regfile/IQ?
- Higher issue width
  → clustering?
- Lower average cache hierarchy access time
- Memory dependence prediction
- Latency tolerance techniques: ILP, MLP, prefetch, runahead, multi-threading

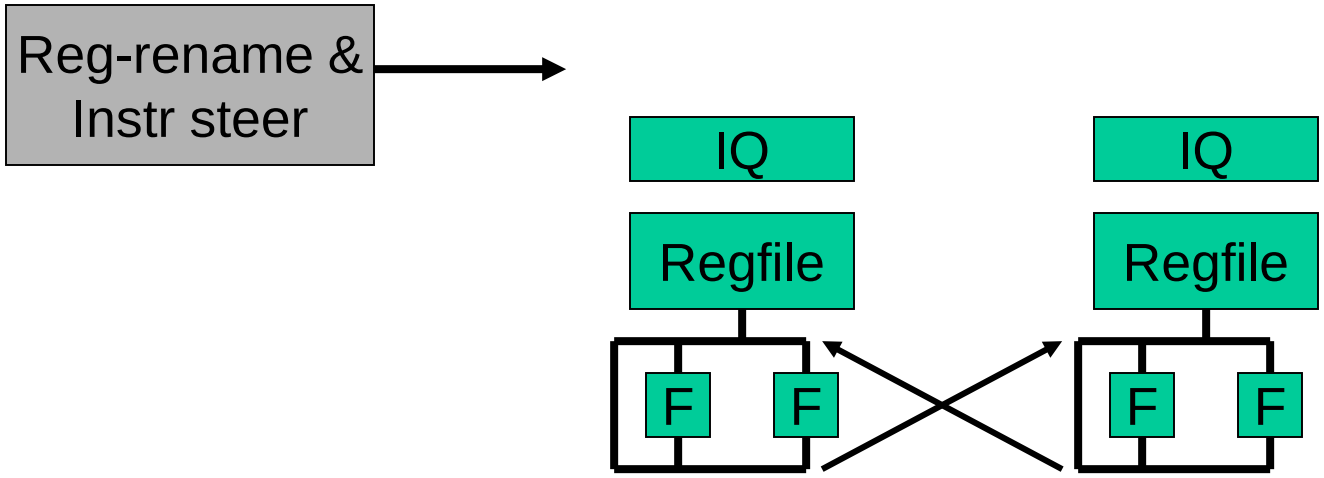
Impact of Mem-Dep Prediction

- In the perfect model, loads only wait for conflicting stores; in naïve model, loads issue speculatively and must be squashed if a dependence is later discovered



From Chrysos and Emer, ISCA'98

Clustering



40 regs in each cluster

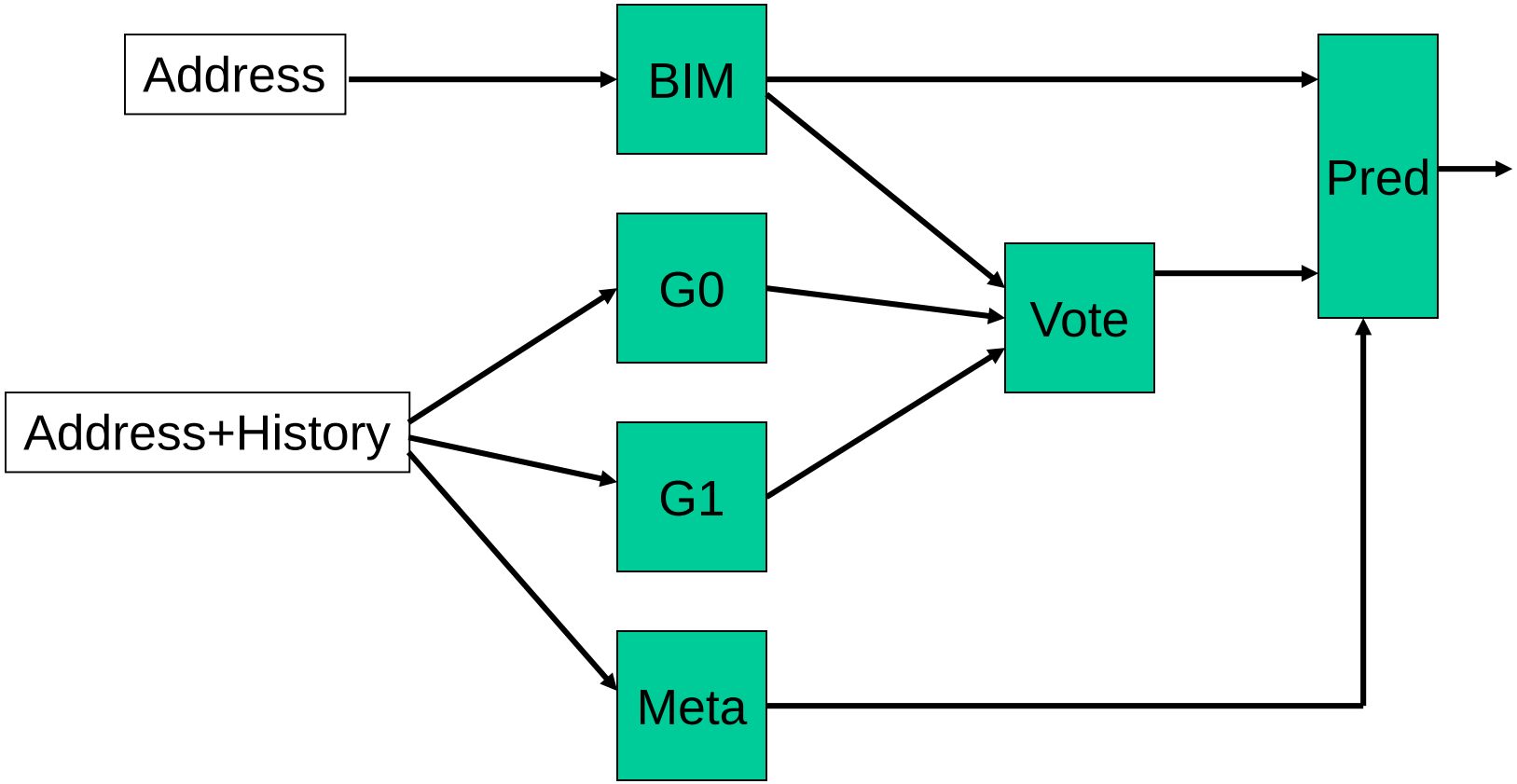
r1 ← r2 + r3
r4 ← r1 + r2
r5 ← r6 + r7
r8 ← r1 + r5

p21 ← p2 + p3
p22 ← p21 + p2
p42 ← p21

p41 ← p56 + p57
p43 ← p42 + p41

r1 is mapped to p21 and p42 – will influence steering and instr commit – on average, only 8 replicated regs

2Bc-gskew Branch Predictor



44 KB; 2-cycle access; used in the Alpha 21464

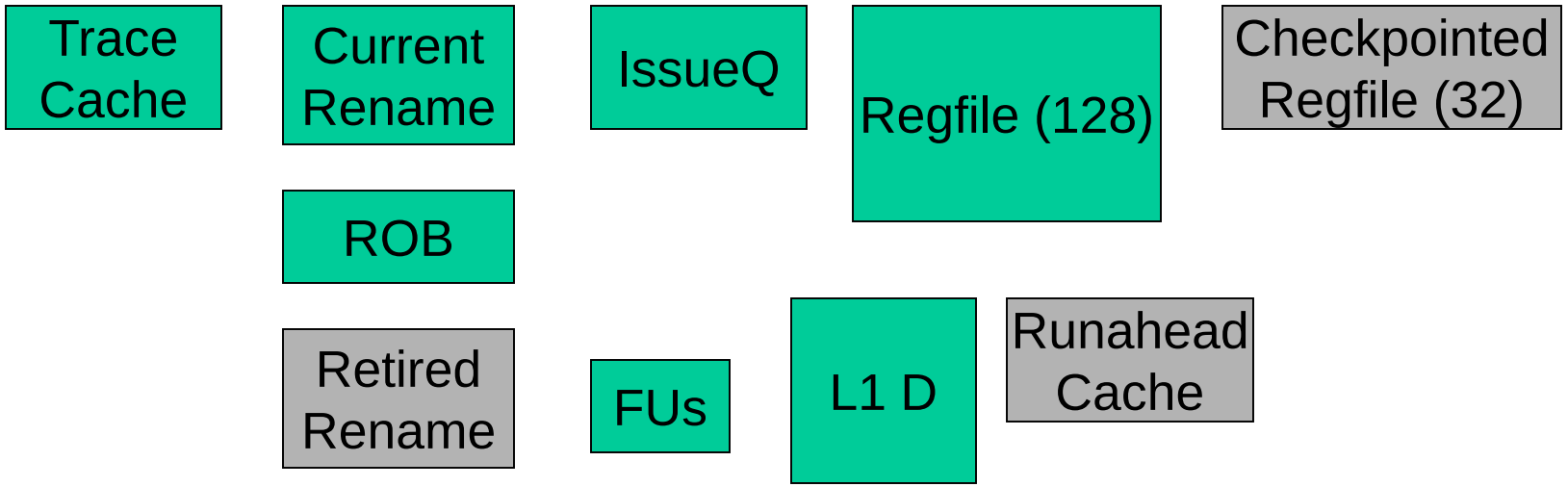
Rules

- On a correct prediction
  - if all agree, no update
  - if they disagree, strengthen correct preds and chooser

- On a misprediction
  - update chooser and recompute the prediction
    - on a correct prediction, strengthen correct preds
    - on a misprediction, update all preds

Runahead

Mutlu et al., HPCA'03



When the oldest instruction is a cache miss, behave like it causes a context-switch:

- checkpoint the committed registers, rename table, return address stack, and branch history register
- assume a bogus value and start a new thread
- this thread cannot modify program state, but can prefetch

Memory Bottlenecks

- 128-entry window, real L2 → 0.77 IPC
- 128-entry window, perfect L2 → 1.69
- 2048-entry window, real L2 → 1.15
- 2048-entry window, perfect L2 → 2.02
- 128-entry window, real L2, runahead → 0.94

Title

- Bullet