Dynamic Speed & Voltage Scaling for GALS Processors

Shelley Chen Anand Eswaran

{schen1, aeswaran}@andrew.cmu.edu

Carnegie Mellon University
Department of Electrical and Computer Engineering
Pittsburgh, PA 15213

Outline

- Motivation
- Problem
- Solution
- Milestone Update

Motivation

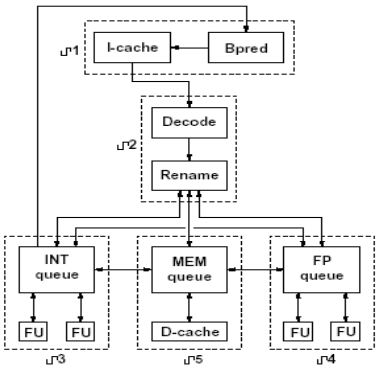
- Dynamic Voltage Scaling in GALS processors
- Smooth control flow would be ideal
- Minimize ineffective speed adjustments

Problem

- On-chip power dissipation soaring with increasing transistor density
- Need for micro-architecture level solutions
- Synchronous processors : speed/voltage adjustments not effective.

Front End Scaling Solution

- Front end throttling based upon commit rate
- Ideally, fetch rate should match commit rate
- Challenges: stability of system, optimal window size, thresholds



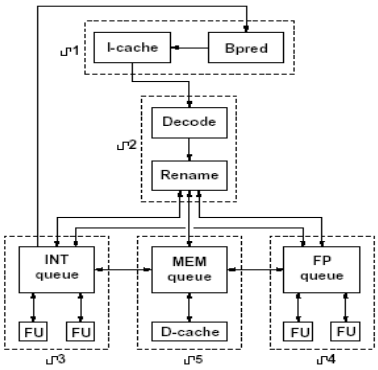
GALS architecture

Fetch Stage Scaling Algorithm

```
if (num_instructions >= window_size) {

  if (commit_rate - fetch_rate > threshold_high)
      clock_rate_fetch = HIGH_MODE;

  if (commit_rate - fetch_rate < threshold_low)
      clock_rate_fetch = LOW_MODE;

  // otherwise, leave it as it is
}
```

Dependence based adjustments

- Functional unit clock scaling usually based solely on queue lengths
- Number of ready instructions in queue (no dependency associated with them) is a better estimator.
- This prevents unnecessary speed-ups of functional units.



GALS architecture

Milestone #1 Update

- Familiar with sim-GALS
- Implemented module for calculating the fetch and commit rates
- Implementing the code for Fetch stage slowdown
- Determined inter-functional unit dependency issues