

Final Presentation:

Dynamic Speed & Voltage Scaling for GALS Processors

Shelley Chen Anand Eswaran

{schen1, aeswaran}@andrew.cmu.edu

December 5, 2002



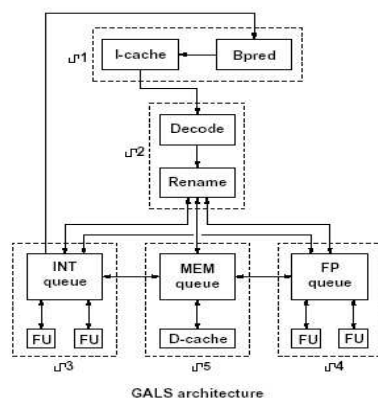
Outline

- Motivation
- Recap
- Final Results
- Conclusion
- Future Work

Motivation

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

Solution: GALS Processors



- Compromise between high performance and low power processors
- Allow for voltage adjustments on a finer scale than system-level



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

```
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  
}
```

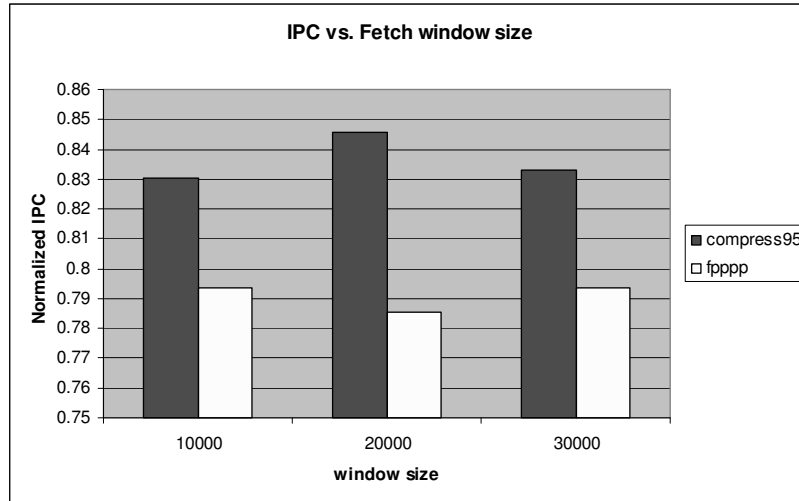


Dependence based adjustments

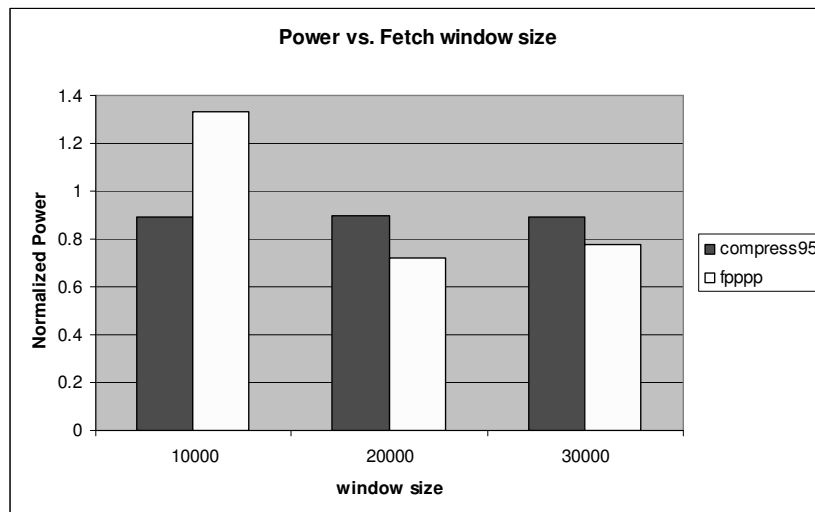
- Functional unit clock scaling usually based solely on queue lengths
- Number of ready instructions in queue (no dependency associated with them) is a better estimator.
- This prevents unnecessary speed-ups of functional units.

```
If (state == HIGH) {  
    if (idep < low_threshold) count++;  
    if (count > CLK_INT) state = LOW;  
} else {  
    if (idep > high_threshold || dep > dep_threshold)  
        count++;  
    if (count > CLK_INT) state = HIGH;  
}
```

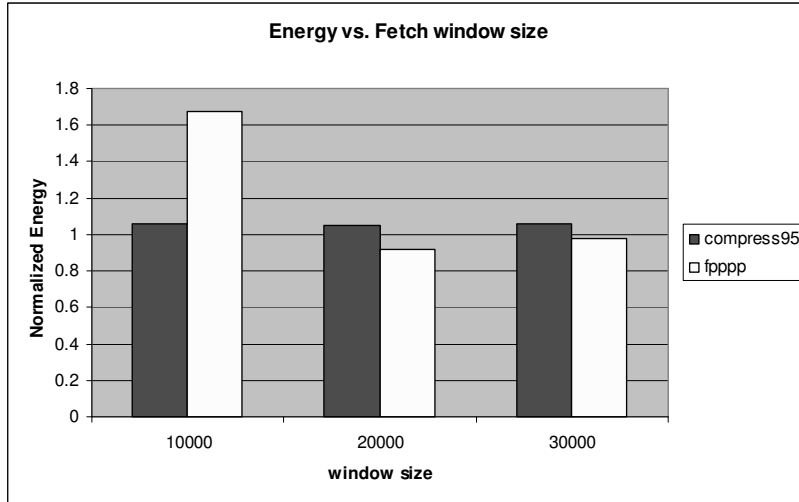
Results of Fetch Stage: IPC



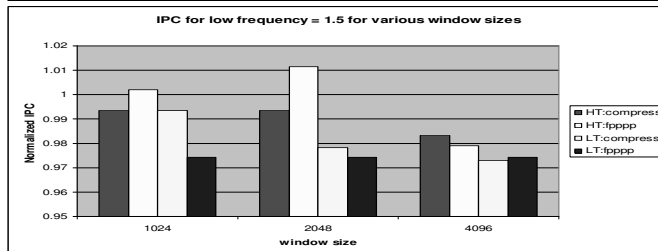
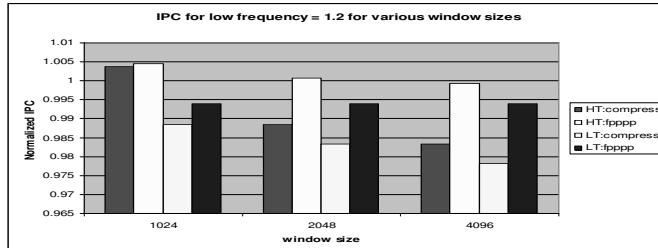
Results of Fetch Stage: Power



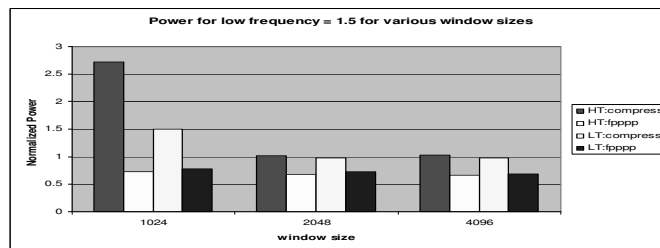
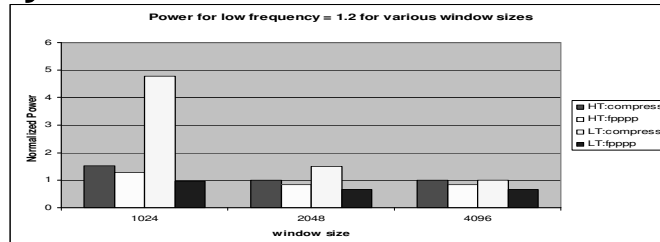
Results of Fetch Stage: Energy



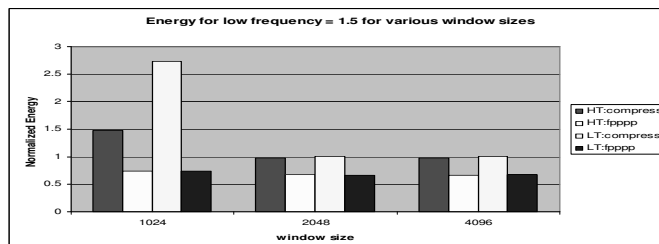
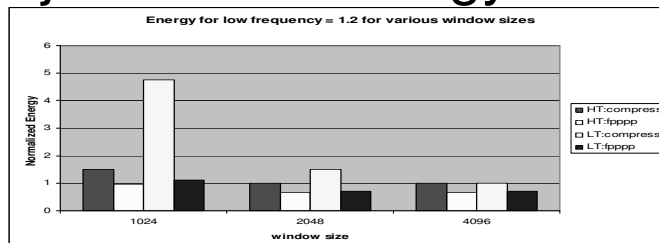
Results of Dependence Based Adjustments: IPC



Results of Dependence Based Adjustments: Power



Results of Dependence Based Adjustments: Energy






Conclusion

- Proposed scheme is definitely feasible as shown in the results
- Modified Simulator performs better in scenarios where there are irregularities in program behavior
- High frequency variation is possible without considerably effecting IPC



Future Work

- Test simulator across wider range of benchmarks
- Explore having dynamic thresholds



Questions?