

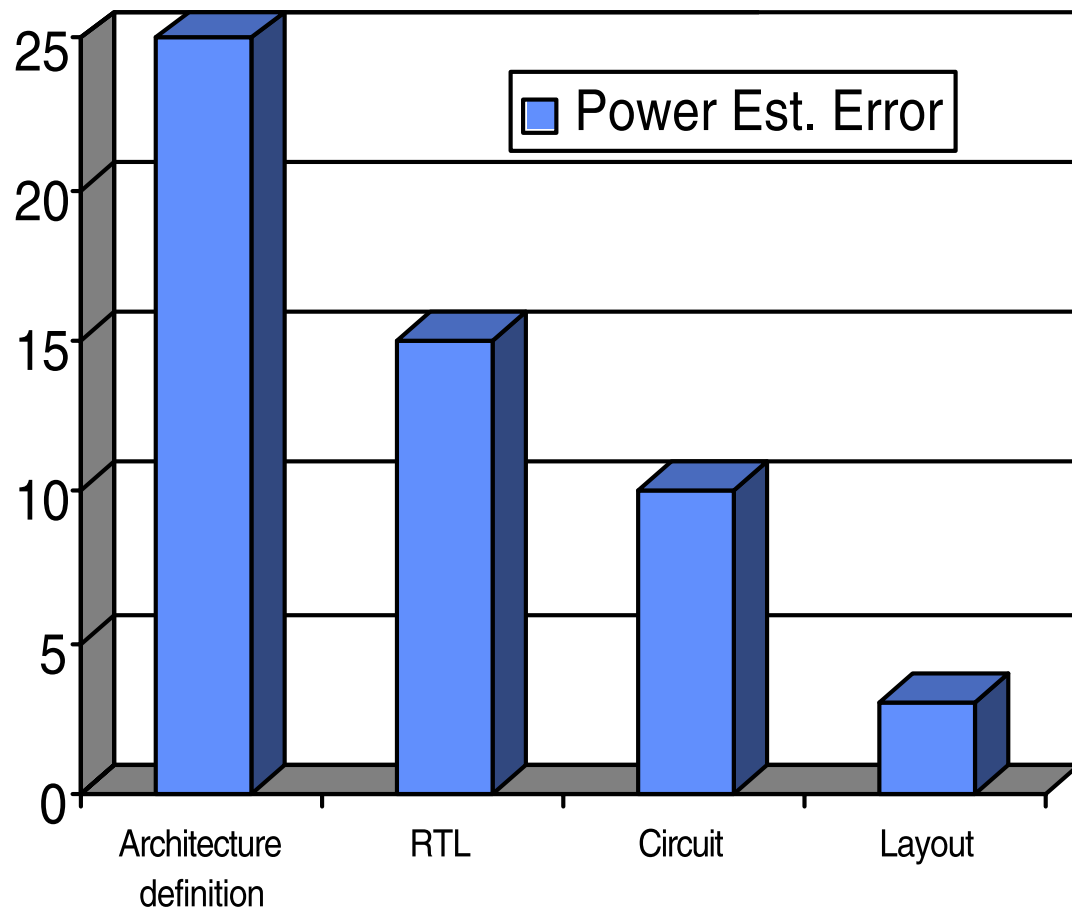


Component Design



- Refine mapping between micro-architectural components and activity model used in Intel's power analyzer
- Provide generalized set of relative and absolute characterizations of commonly used, specific micro-architectural components
- Accurate power model of SA-1100 as early in design process as possible

Power Estimation Errors vs. Microprocessor Design Phases



Two architectural definition stage power models are currently available

- Cai-Lim model (Intel)
- Wattch model (Princeton)



Activity-based Power Estimation



- **Break micro-architecture into components**
 - Assign an average cost per use of the component
 - Keep track of component use
- **Faster than lower level power modeling tools**
- **25% error in power estimation**



- **Active power density-based power costs**
- **.25 μm process**
- **Conditional clocking, inactive = 10% active**
- **Advantages**
 - Detailed activity counters, closely tied to simulation architecture
 - Integrated clock distribution model
- **Disadvantages**
 - Design details (structure sizes, component designs) hidden
 - Numbers hard-coded, difficult to scale and extend



- **Capacitance-based power costs**
- **.10 μm - .80 μm processes available**
- **Conditional clocking, inactive = 10% active**
- **Advantages**
 - Initialization of component costs using simulation configuration
 - Many design details exposed
- **Disadvantages**
 - Lack of fine-grained activity counters
 - Power for clock distribution separate
- **Accurate to within ~10% of layout level tools**



Comparing the Models



- **SimpleScalar 3.0 sim-outorder simulator containing both Cai-Lim and Wattch**
- **Use Wattch's estimate of accurate to within 10% of layout**
- **Look for statistically significant results**
 - **Power and energy reductions relative to base case**
 - **Differences between models**



Styles of Power Model Usage



- **Create a new processor design**
 - No functional or circuit models exist yet
 - Policies, such as clock distribution, may not be decided
 - Less accurate

- **Extend an existing processor design**
 - Exploit detailed functional and circuit models
 - More accurate
 - Easier to validate



Lower Power Issue Mode



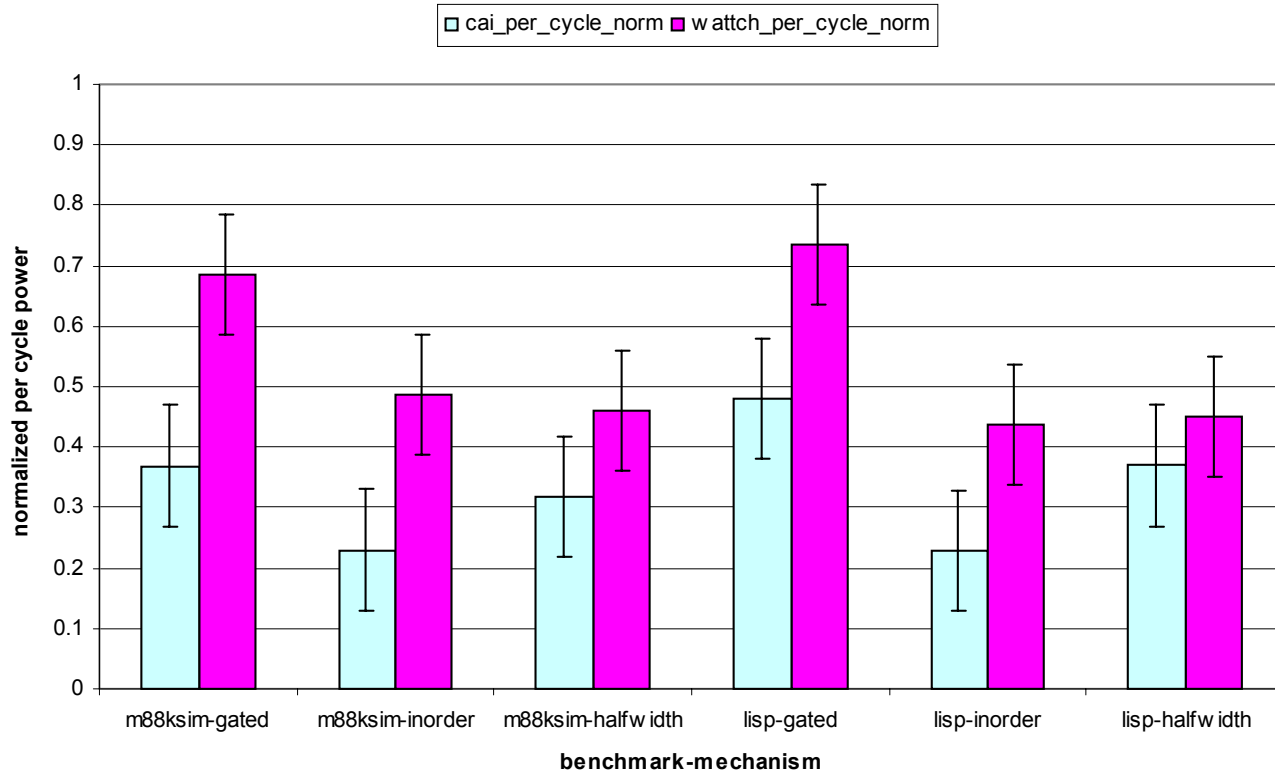
- Explore a possible new processor design
- Dynamically match issue mode of micro-architecture to current IPC requirements of workload
- Low IPC requirements can be satisfied with lower power issue modes
- Most interested in power reduction
- Power reduction from activity reduction



IPC Matching Power

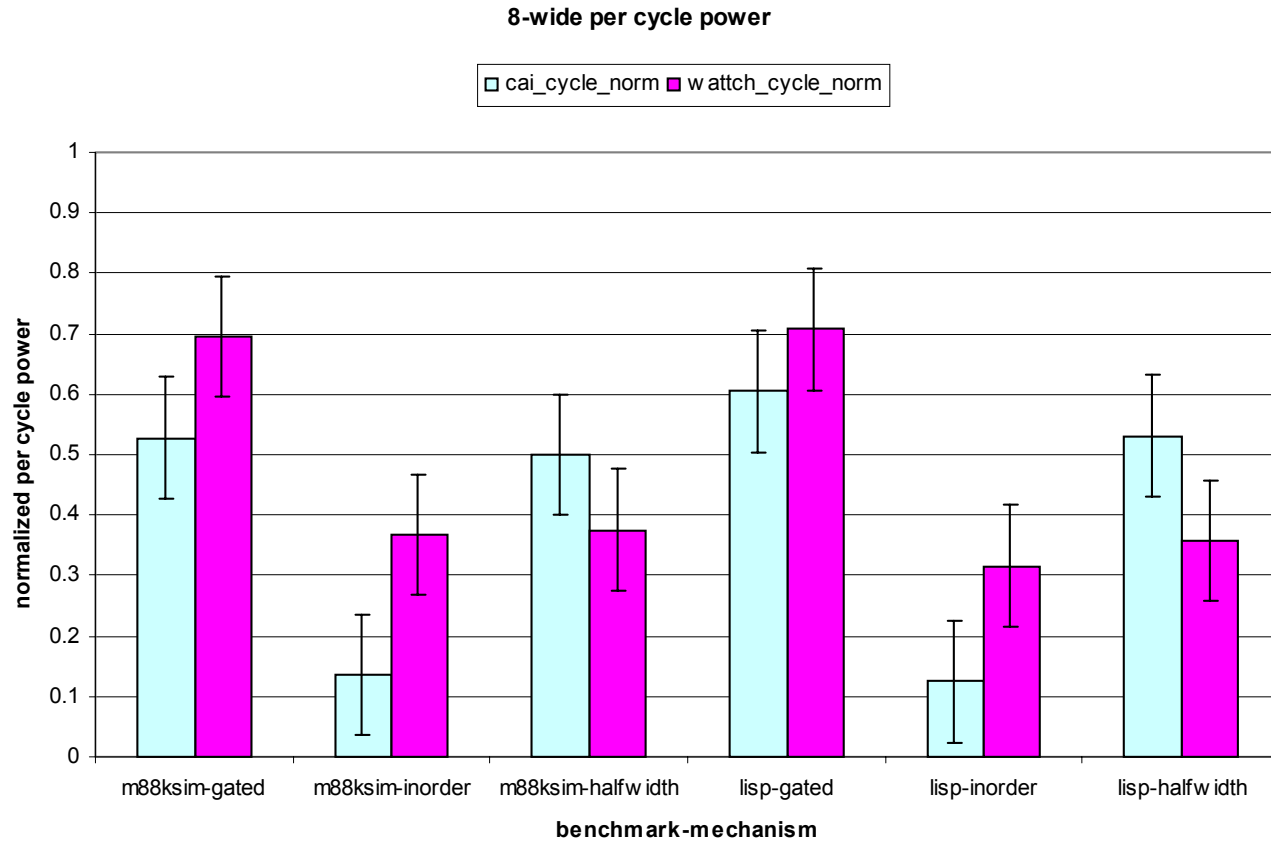


4-wide power per cycle





IPC Matching Power





IPC Matching Power Conclusions

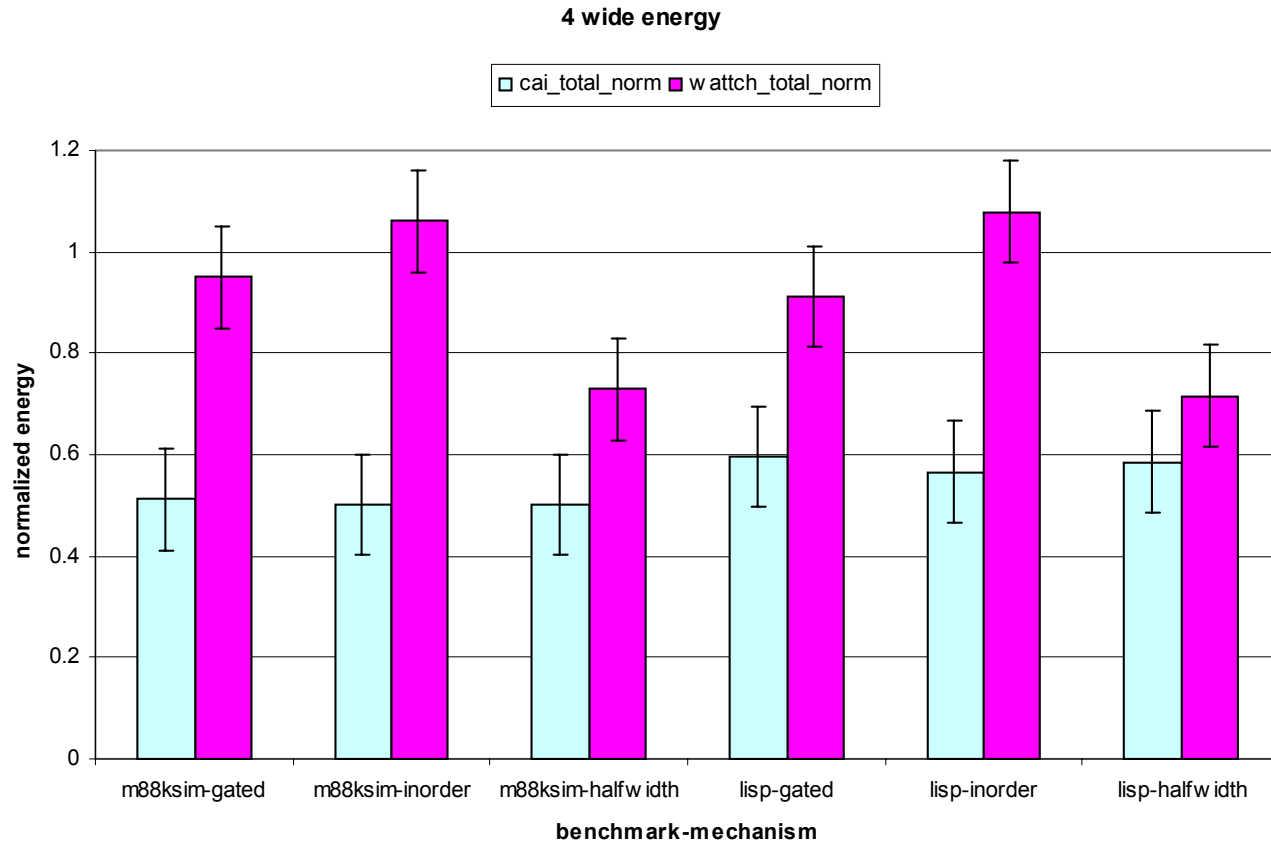


- **Which mode to select?**
 - Cai-Lim: inorder issue
 - Wattach: inorder issue or half-width machine

- **Statistical significance**
 - All differ from base case
 - Also differ from each other in some cases

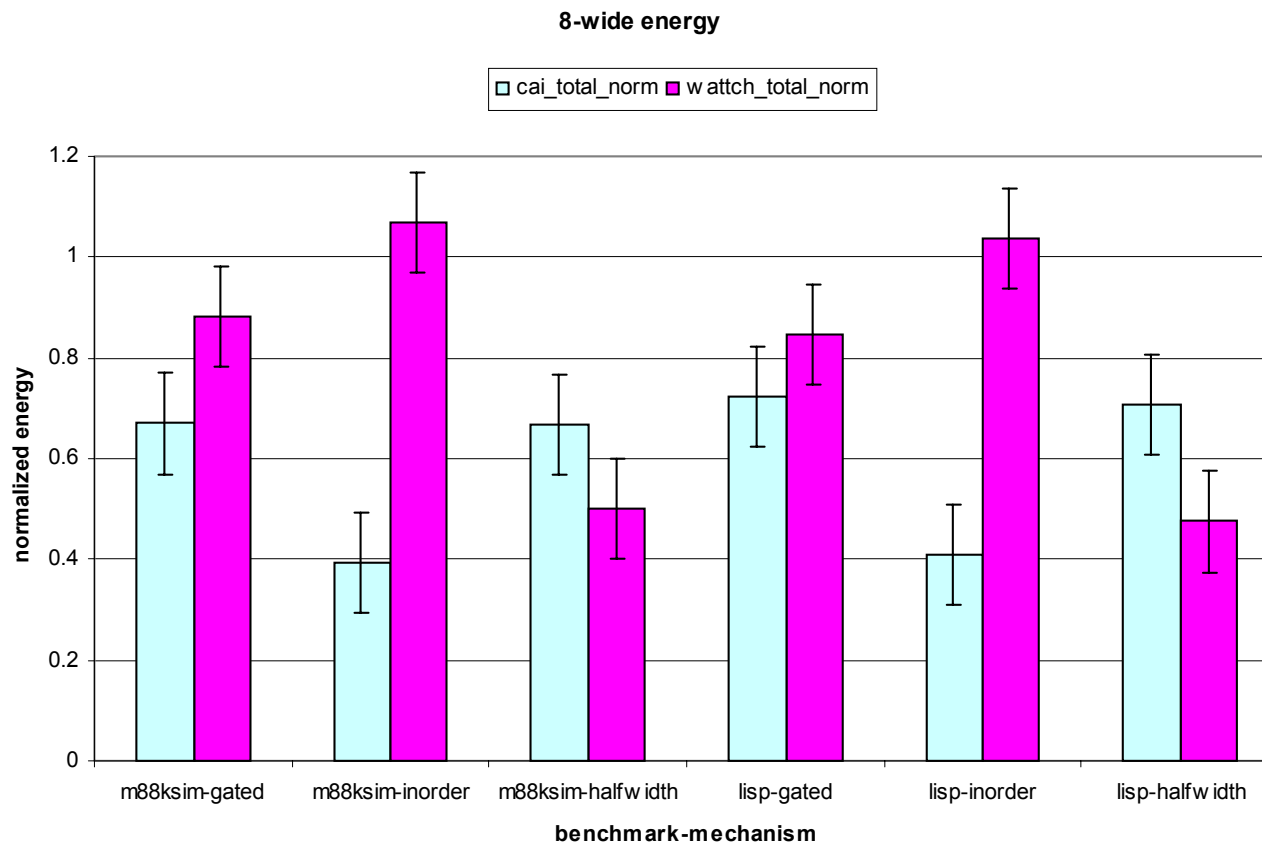


IPC Matching Energy





IPC Matching Energy





IPC Matching Energy Conclusions



■ Which mode to select?

□ Cai-Lim

- 4-wide: all low power modes equivalent
- 8-wide: inorder

□ Wattch: half-width machine

- Inorder issue uses the most energy

■ Statistical significance

□ Wattch doesn't always differ from base case

□ Cai-Lim always differs from base case

□ Also differ from each other in almost all cases



IPC Matching Differences



■ Cai-Lim

- More fine grained architectural-level activity counters
- More closely matches simulation architecture

■ Wattch

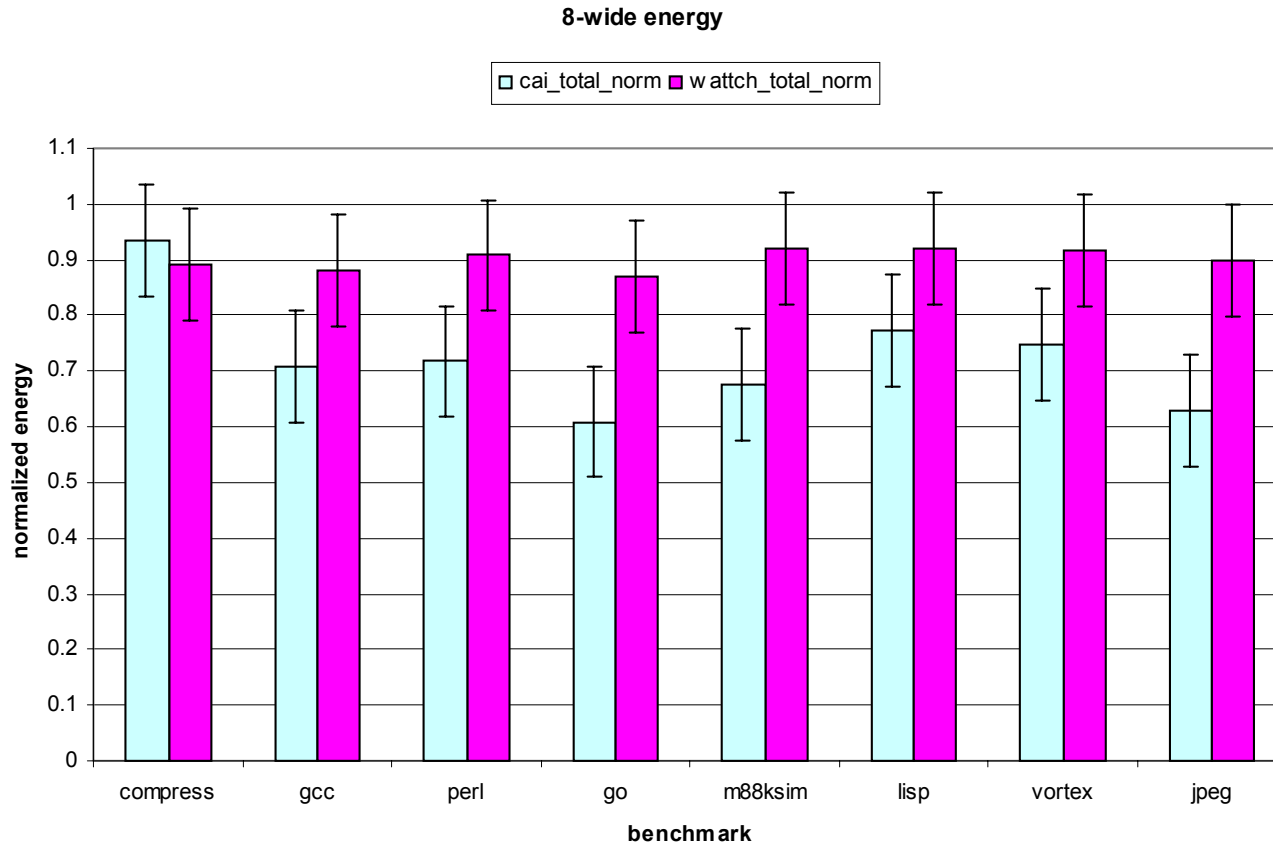
- Misses some architectural activities (e.g. write from fetch to dispatch queue)
- Fetch model differs from simulation architecture



- Explore an existing processor design
- Dynamically size the instruction queue based on where committed instructions are issued from
- Issue requirements can sometimes be met with a smaller queue size
- Interested only in energy reduction
- Power reduction primarily from additional conditional clocking
- Verification of prior work by Folegnani et al



Dynamic Issue Window Resizing





Instruction Queue Resizing Energy Conclusions



■ Is it worthwhile?

□ Cai-Lim

- 4-wide: almost never
- 8-wide: yes

□ Wattch: almost never

■ Statistical significance

□ Wattch almost never differs from base case

□ Cai-Lim 8-wide differs from base case

□ Models differ from each other in some 8-wide benchmarks



Instruction Queue Resizing Differences



■ Cai-Lim

- Linear scaling applied, perhaps erroneously
- Clock is scaled by conditionally clocking the unused segments of queue

■ Wattch

- Log N and N Log N scaling applied to different components
- Unable to scale clock power because it is a separate, pre-calculated component



Current Power Model Results



- **Current models are not sufficient**
 - Cai-Lim details are hidden
 - Watch not detailed enough
 - Need correlation studies to show contribution to error from different components

- **Significance of results**
 - Models give contradictory results in some cases



Next Generation Power Model?



- **Access to details for accurate scaling of components**
- **Fine grained activity counters**
- **Clock distribution options**
 - Integrate into each component
 - Parameterized, but separate with component level contributions



The Next Step



- **Build a new power model that is closely coupled to micro-architecture**
- **Develop infrastructure for correlation studies**
- **Whitebox calibration**
 - Perform validation and verification