
EXPLOITING BIT-SLICE INACTIVITIES FOR REDUCING ENERGY REQUIREMENTS OF SUPERSCALAR PROCESSORS

Kanad Ghose, Dmitry Ponomarev, Gurhan Kucuk, Andrew Flinders
Department of Computer Science
State University of New York
Binghamton, NY 13902-6000

Peter M. Kogge
Dept. of Computer Science and Eng., University of Notre Dame, Notre
Dame, IN, 46656

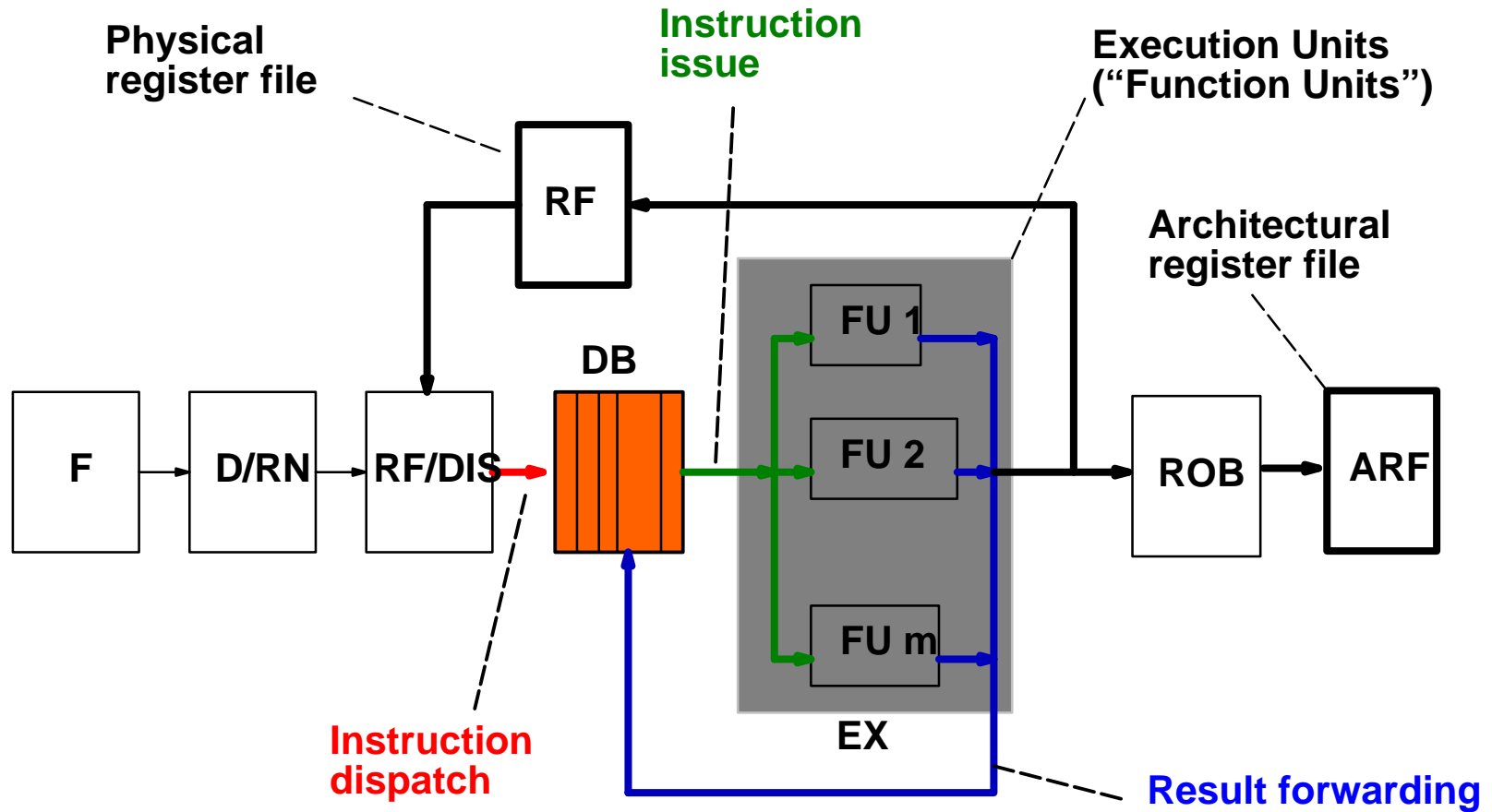
Nikzad (Benny) Toomarian
CISM, Jet Propulsion Lab, Pasadena, CA, 91109

Koolchips Workshop, December 10, 2000

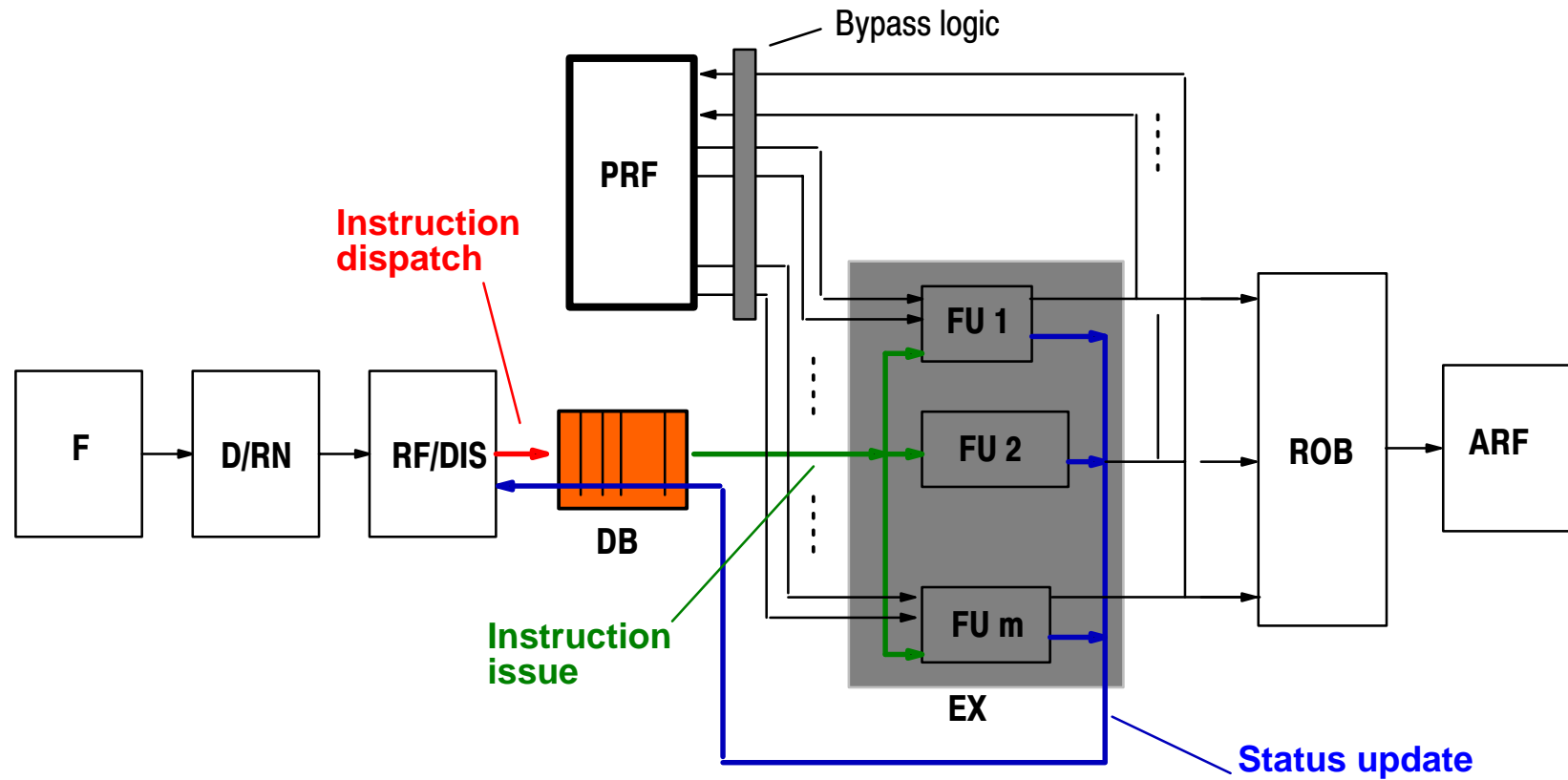
Exploiting Bit-Slice Inactivities in Datapaths

- **Expectation:** Significant number of **BYTES** within the operands driven across the on-chip interconnections and stored in the RFs, ROB and DB contain **ALL ZEROS**.
- **Reasons:**
 - Small integer literals (address offsets, literal operands, flags, byte ops, ...)
 - Consequence of byte packing and unpacking operations and usage of the bit or byte masks to isolate parts of the operands.
 - Floating point operands may not use all of the bits allowed in the mantissa field
 - Lower-precision data may not make use of full datapath width
- **Observation:** within the non-zero part of an operand, about 70% of the bits are identical to what was driven immediately before on the data flow path.

Where Bit-Slice Inactivity Can Be Exploited: Superscalar Datapath A



Where Bit-Slice Inactivity Can Be Exploited: Superscalar Datapath B

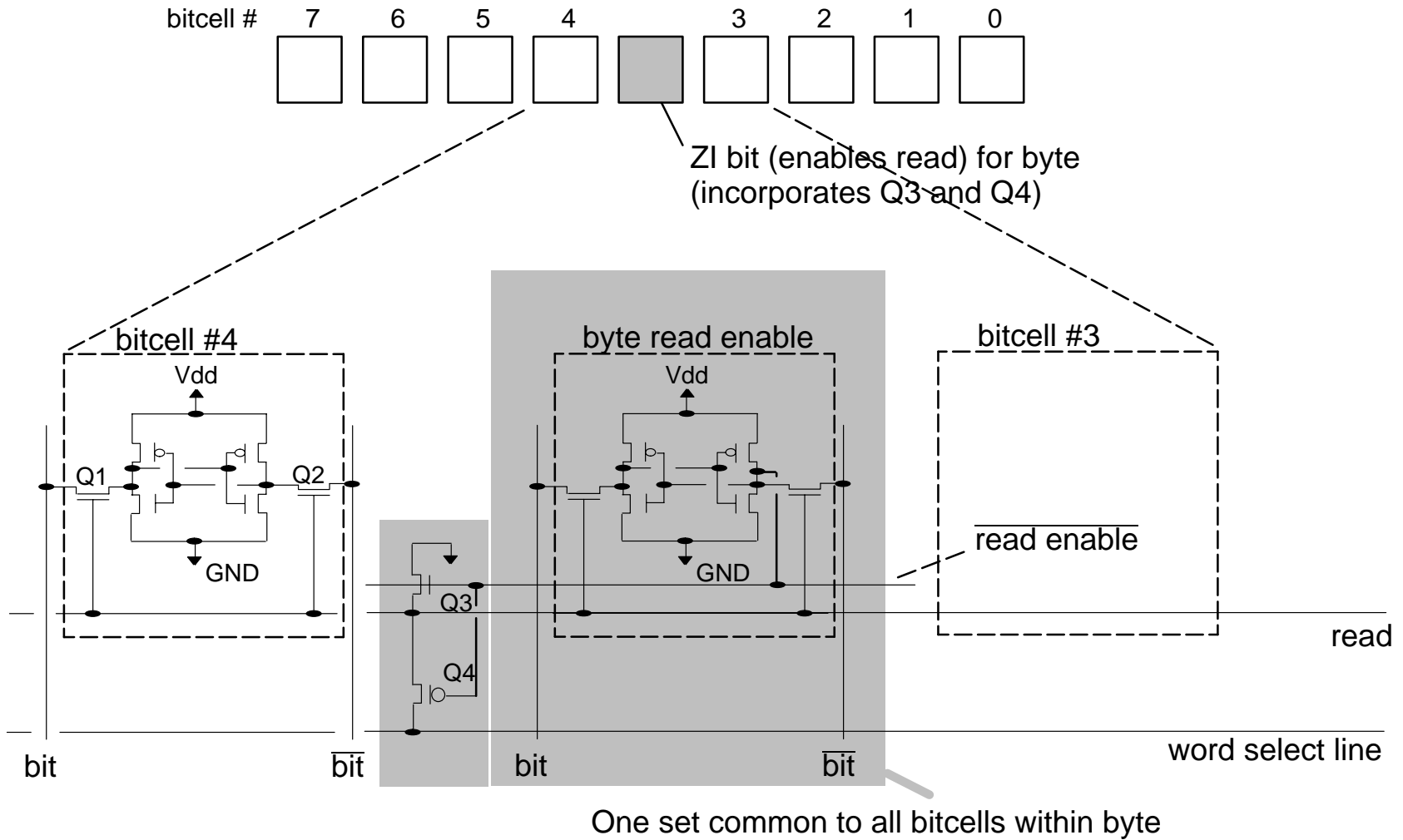


Encoding the bytes containing all zeroes (ZE – zero encoding)

Idea: Instead of driving byte with all zeroes, encode it using the ZI (Zero Indicator) bit and only drive this bit, thus achieving power savings. In addition to ZE, on non-zero part of the operand, do not drive bit lines whose values had not changed since the time they were last driven.

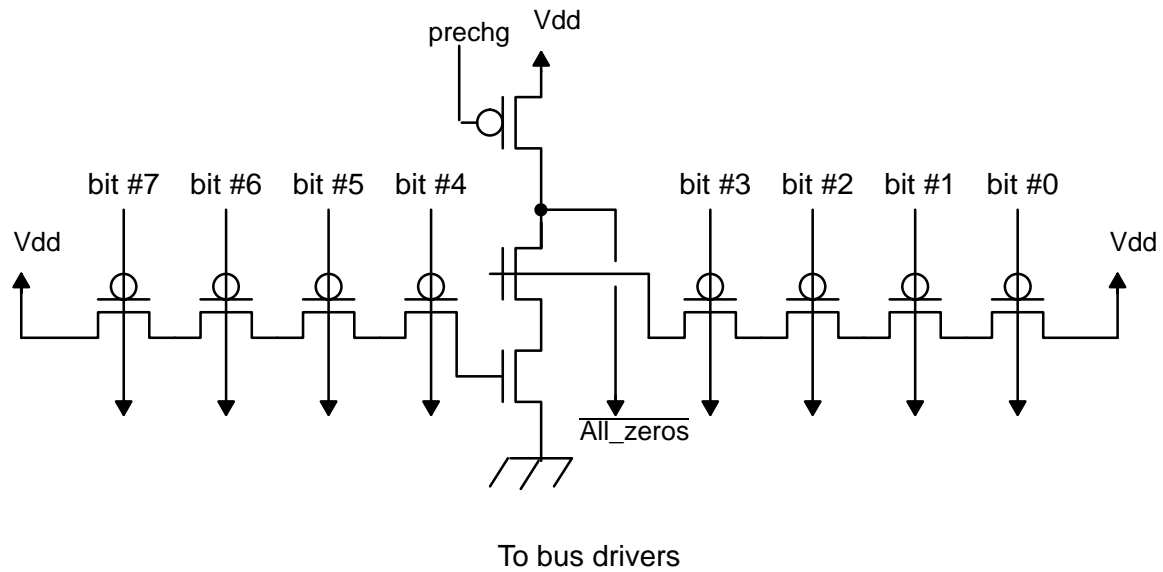
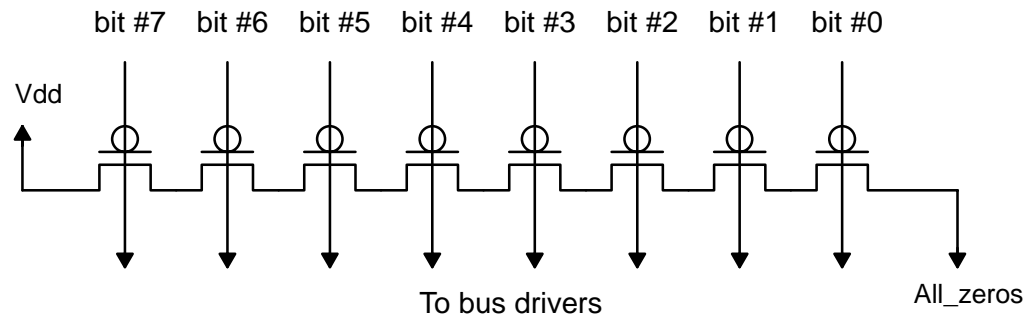
- [E]ncoding can be done by Function Units:
 - Either consciously as part of the logic
 - OR mechanically: by detecting zero bytes in results using fast pull-down logic
- [C]an use encoded data for writing register files, ROB and DB; need to store encoding as well
- [H]arder to do: avoid energy dissipation within RFs, ROB and DB in the course of reading stored data out
 - this requires circuit-level facilities

Associated Circuit Techniques: Readout Logic

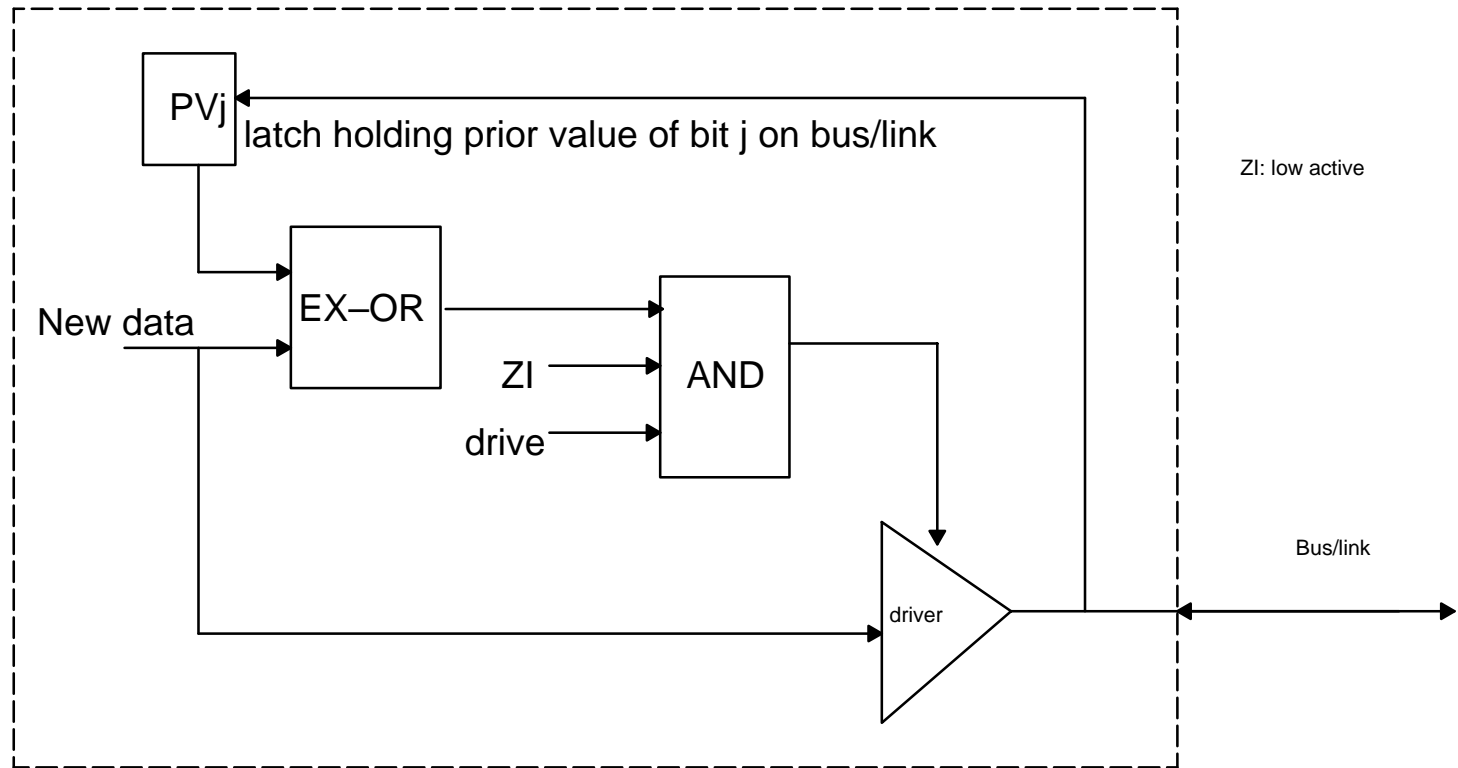


Associated Circuit Techniques: Encoding Logic for bytes of all zeroes

From latch, FU



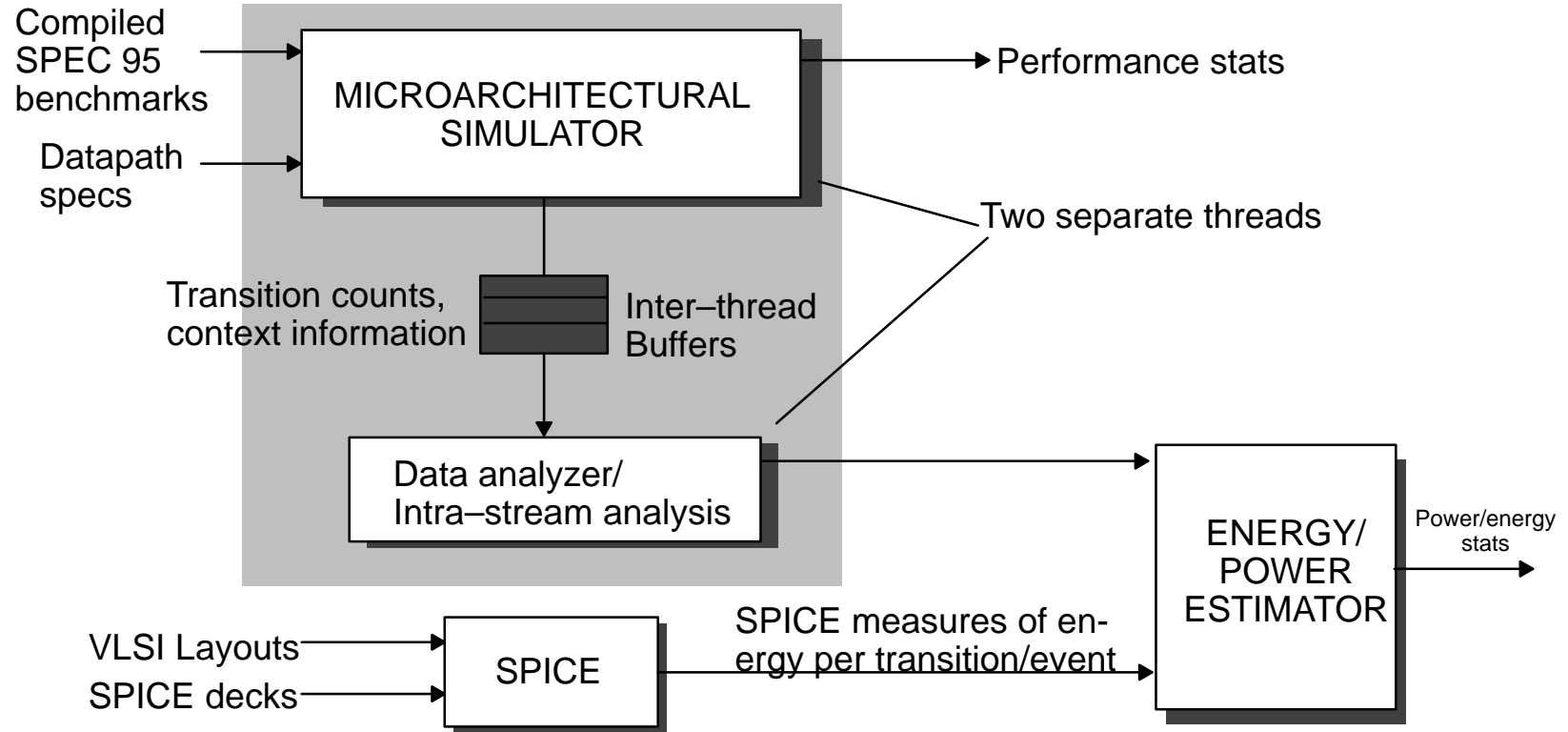
Associated Circuit Techniques: circuit for driving only the bit lines of non-zero data slices that changed since the last transfer



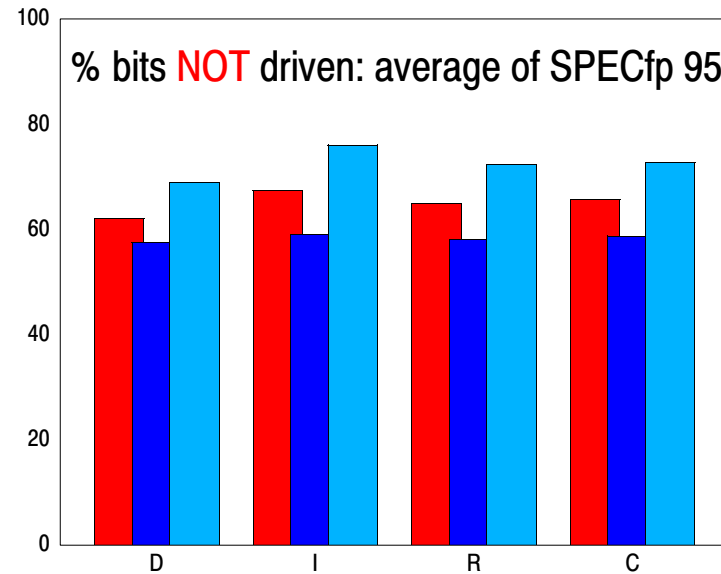
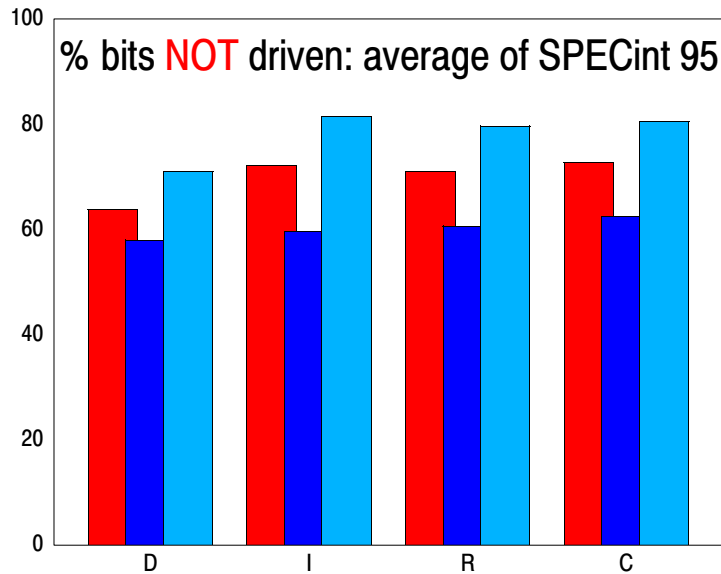
Approach

- Used a **true cycle-by-cycle register-level simulator for a typical superscalar pipeline**. SimpleScalar has been substantially modified for this purpose
- Simulated the execution of **SPEC 95 benchmarks**, and gathered the statistics about percentage of bytes that are all zeroes on various on-chip interconnections.
- Collected transition counts for each major datapath component
- Used **SPICE measurements for the VLSI layout** of a register file, dispatch buffer and reorder buffer in a 0.5 micron, 4-metal layer process to estimate the power dissipated for each type of transition within each major component
- Computed total energy dissipation within register files, ROB and DB based on transition counts and energy dissipated per transition for each component, for each benchmark
- **Overall accuracy is as good as it gets short of an implementation**

Datapath Power Estimator



Results – potential for power savings on the interconnections (Datapath A)



- Not driving unchanged bits across entire data width including all zero bytes
- Not driving all zero bytes, driving ZI bits
- Driving only bits that change within non-zero bytes and ZI bits, not driving zero byte fields

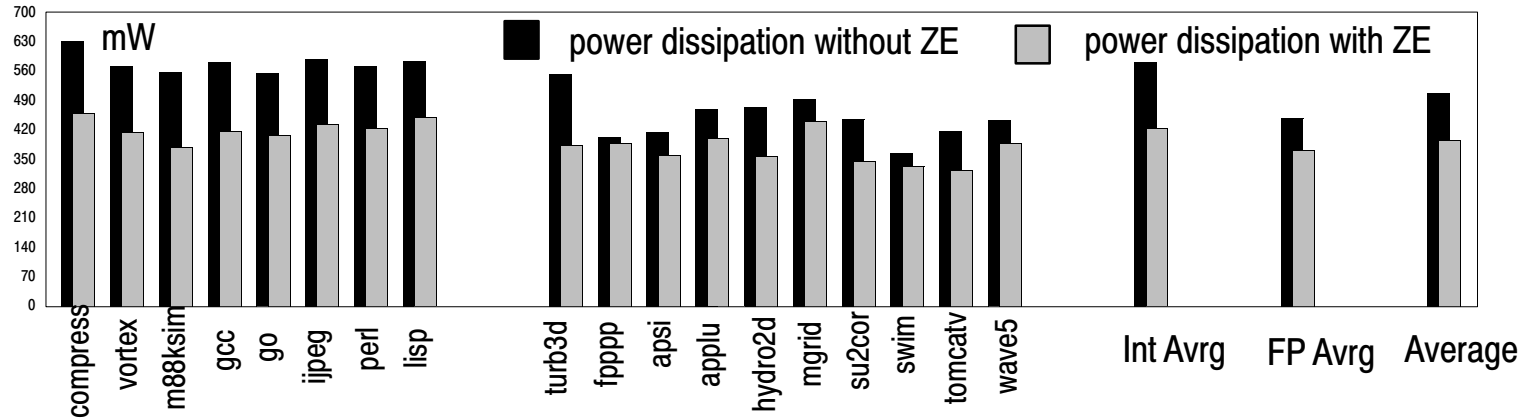
D: dispatch buses I: issue buses R: Result buses C: commit buses

Main Sources of Energy Dissipation in Dispatch Buffer (DB)

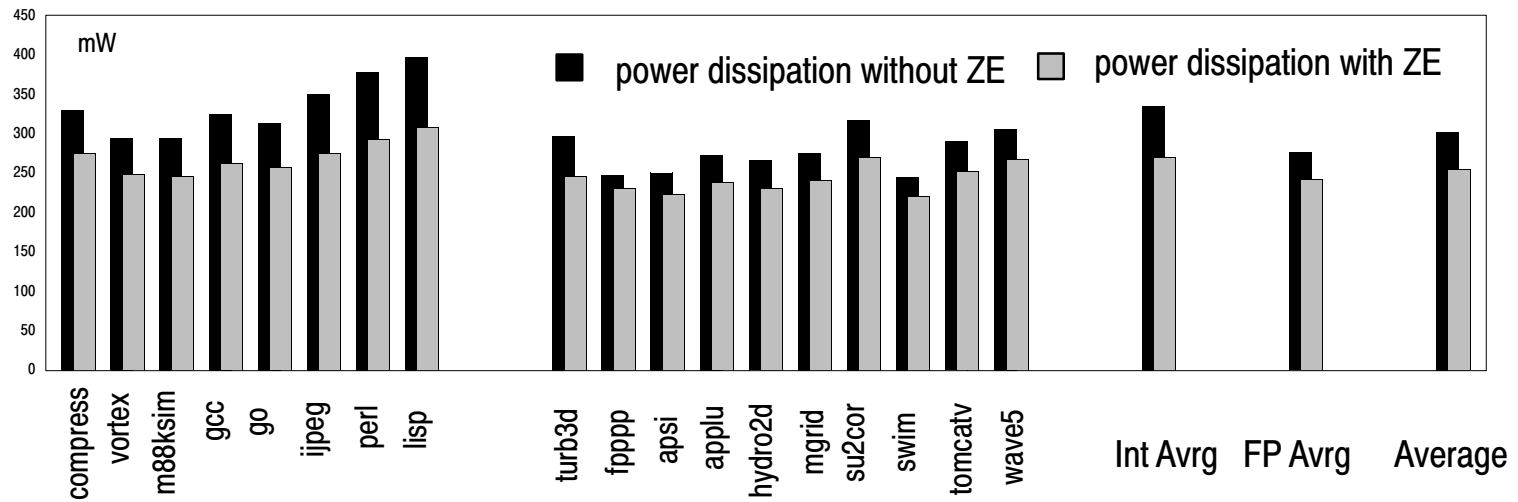
- Energy dissipated in the DB in the process of establishing DB entries for dispatched instructions.
- Energy dissipated in the DB when FUs complete and forward the results (Datapath A) and/or status information (Datapath B) to the DB entries.
-
- Dissipations in the DB at the time of issuing instructions to the FUs.

Results – power savings within the DB

DATAPATH A



DATAPATH B

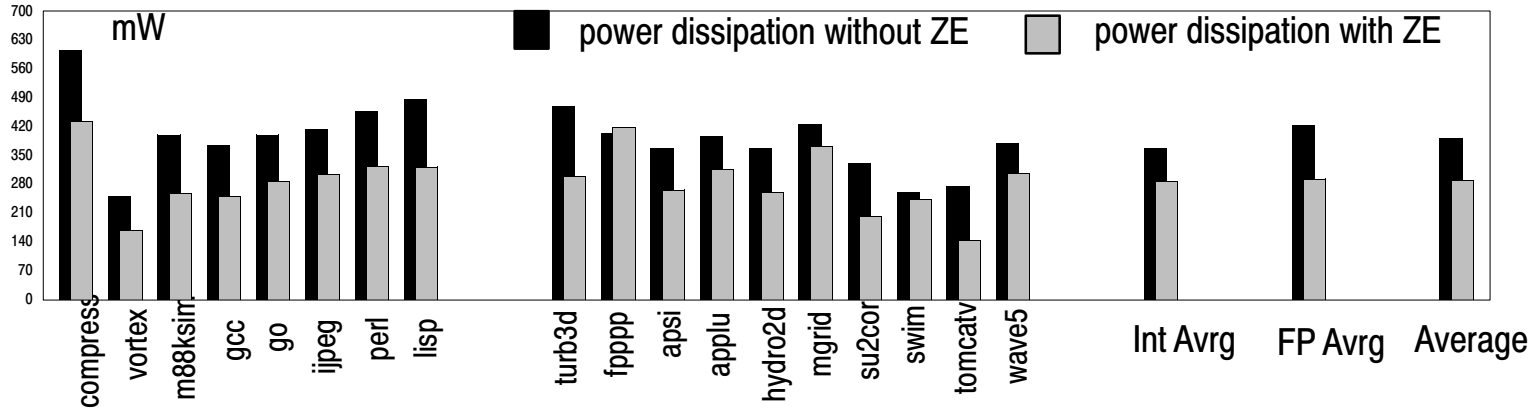


Main Sources of Energy Dissipation in Physical Register Files (PRF) and Architectural Register Files (ARF)

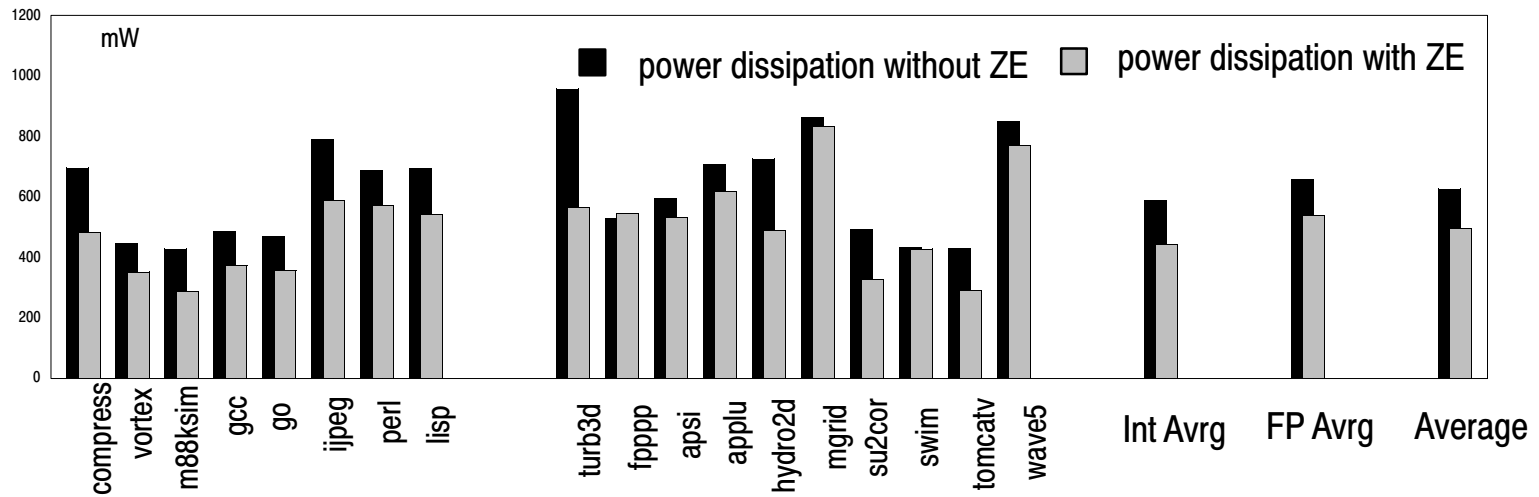
- Energy dissipated in the PRF in the process of reading the source operands at the dispatch time.
- Energy dissipated in the PRF when FUs complete and write the results and/or status information into the registers within the PRF.
- Energy dissipated in the ARF in the process of committing instruction from the ROB.

Results – power savings within the PRF

DATAPATH A



DATAPATH B

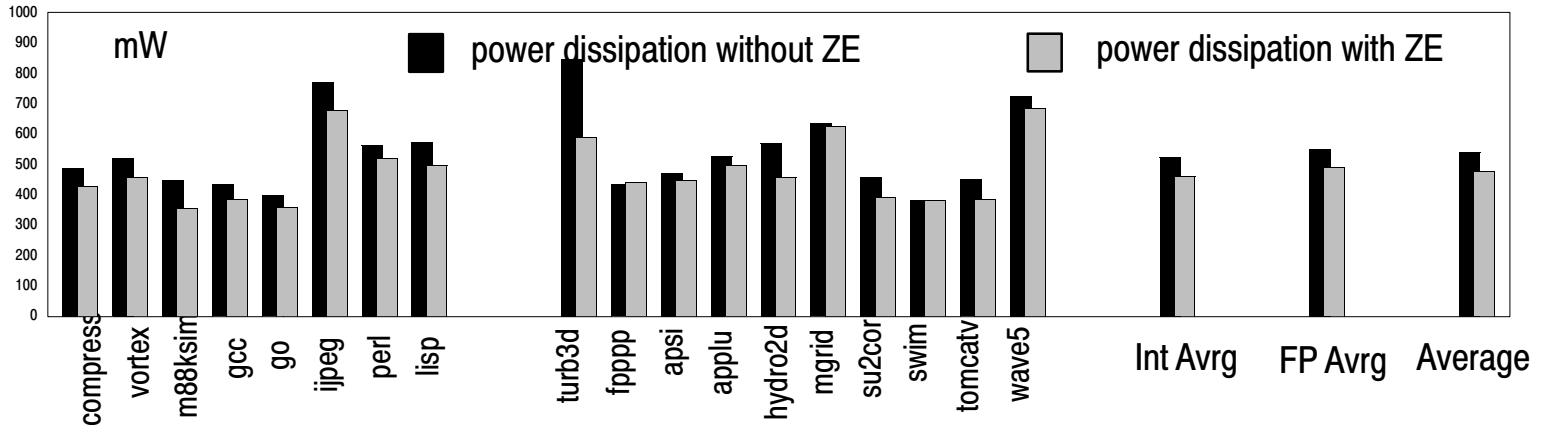


Main Sources of Energy Dissipation in Reorder Buffer (ROB)

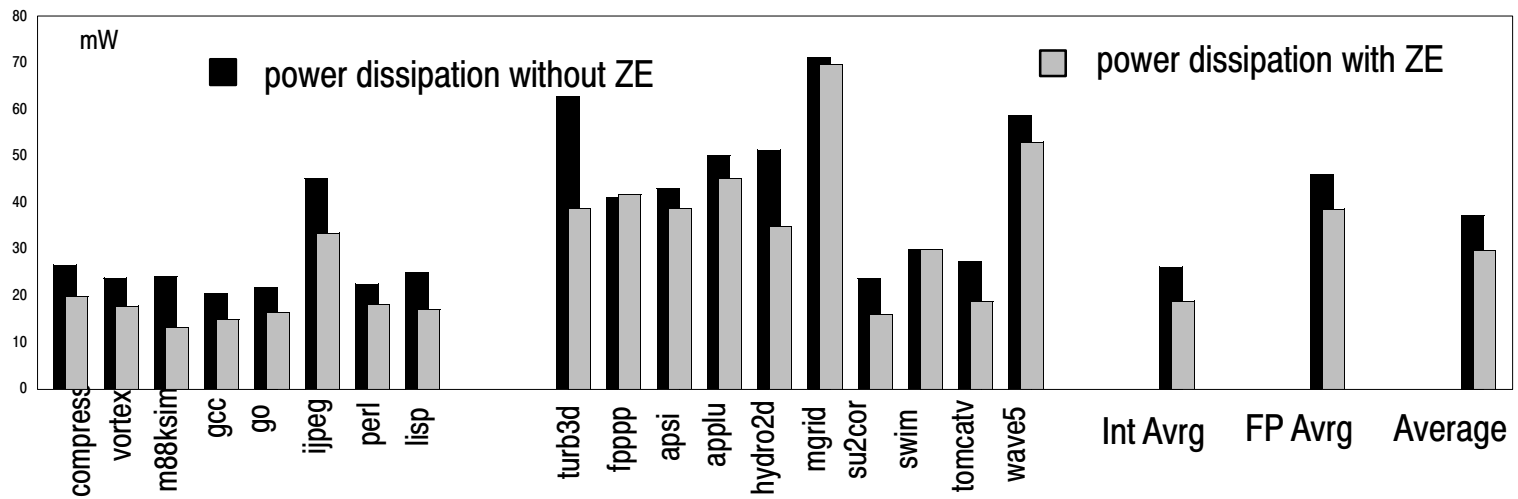
- Dissipations that occur in the ROB at the time of setting up the entry for dispatched instructions.
 - Dissipations that take place when results are written to the ROB from the function units.
 - Dissipations that take place when ROB entries are committed; this dissipation is the energy spent in reading the register file that implements the ROB.
-
- Dissipations that occur in clearing the ROB on mispredictions – this involves clearing the valid bit of all ROB entries. Compared to all of the other ROB dissipation components, this is quite small.

Results – power savings within the ROB and ARF

ROB – DATAPATHS A and B



ARF – DATAPATHS A and B

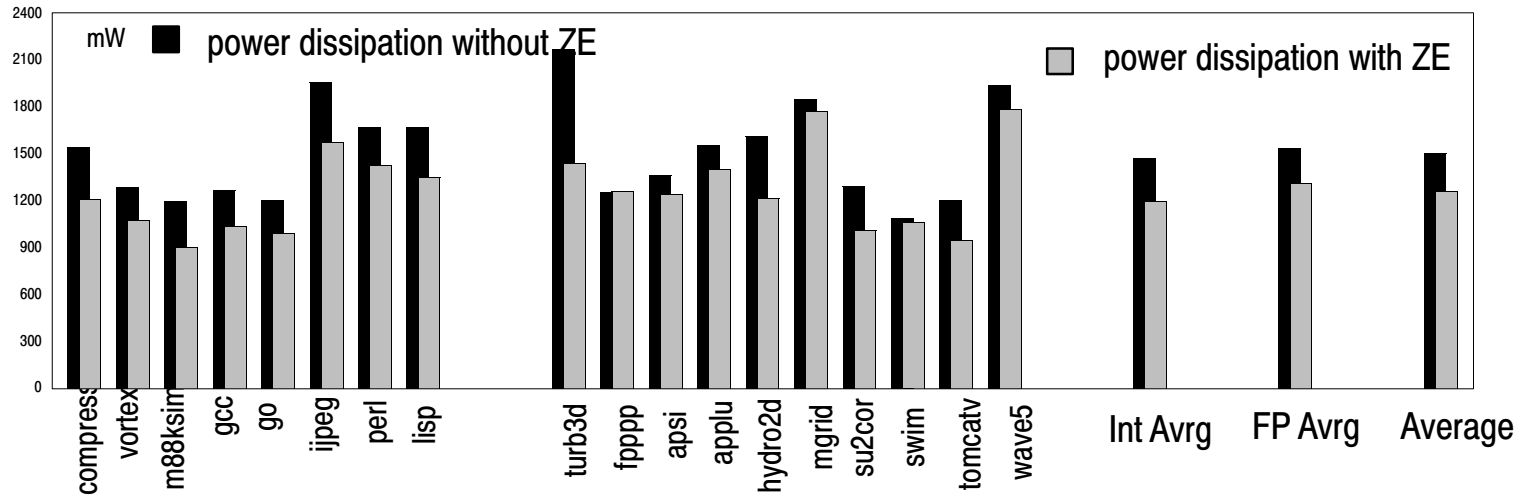


Results – total power savings

DATAPATH A



DATAPATH B



Conclusions

- Significant power savings in the superscalar processor can be realized using the dynamic activation of datapath slices.
- This technique exploits the lack of entropy in the data streams.
- Potential for power savings in the on-chip interconnections is in excess of 80%. As feature sizes decrease, this component of power dissipation becomes more dominant as wire capacitances increase.

Conclusions (continued)

- Total power savings realized within the DB, the ROB, PRF and ARF:

Datapath A:

SPECInt95	23%
SPECfp95	15%
Average	18%

Datapath B:

SPECInt95	18%
SPECfp95	14,3%
Average	16,3%