1. The need: Why this research was needed

1.1 Requirements, design and fine, dynamic, fine-grained, source-level techniques to estimate the energy consumed by their software.

1.2 Focus: this is on the core of single-issue CPUs (no memory hierarchy, no VLIW, ...)

1.3 State of the Art: current techniques do not satisfy the above requirements.

2. Theory: how this technique works

2.1 Divide and conquer: $\mathbb{C} = \mathbb{n} + \mathbb{c}$

2.2 Determine single-execution costs (need a cost model, have to know the code structure!)

2.3 Determine execution counts (.profile the program to allow for cost counting)

3. Results: The technique is accurate and fast

3.1 ANSI-C compliant flow implementation available

3.2 New experiments – Setup:

3.3 New experiments – Results: accuracy model shows slight correlation with reference (up to 2% performance overhead in the worst case, only a dozen cases that normal)

4. Uses and developments

4.1 Automated source-code optimization

4.2 Support for VWR

4.3 Support for VLIW

4.4 Extension to C++

Selected Publications

- Book chapters:
  - "Energy estimation in the microprocessor design process" in [Etienne C., Carre, J., Tournier, S., Miracle, P., Ma, P., Marzullo, P., Sarp, C., Sengupta, P. 1998].
  - "Energy models and techniques for evaluating energy consumption in software systems" in [Chung, D., E. 2002].

- Conference papers:
  - [Carre, E., Carre, J., Tournier, S., Miracle, P., Marzullo, P., Sarp, C., Sengupta, P. 2000].
  - [Carre, E., Carre, J., Tournier, S., Miracle, P., Marzullo, P., Sarp, C., Sengupta, P. 2001].