A major trend in the computing industry is the move towards computers in portable products. This can be seen as the sales of laptops now surpass desktops, the emergence of a new class of devices known as Personal Digital Assistants (PDA's), and the growth in the sale of consumer electronics devices such as pagers, cellular phones, viewmen. For a large number of these products, battery life is an important consideration.

Often a major determinant of the battery life of these products is the energy consumption of the embedded computing system. Therefore it is important to devise low energy computing systems for these products. However, the computing system is often utilized to perform such functions as image processing, voice recognition, handwriting recognition, signal processing, data compression/decompression as well as more general purpose computing and execution of a non-trivial operating system. The sum total of the performance requirements can approach that of some modern desktops. This minitrack is intended to invite discussion of how techniques, appropriate to high performance ILP processors, can be used to reduce the energy consumption of such computing systems.

Basically, energy in CMOS-based computing systems is consumed primarily in logic transitions. During execution of a function on such a system a set of logic transitions occurs. The energy consumed in executing the function depends on the number and magnitude of the transitions. The energy consumed by a single transition varies as $C*V^2$, where $C$ is the capacitance being switched by the transition and $V$ is the supply voltage. Reducing the energy consumed in executing the function, then involves reducing the number of transitions, the capacitance, or the supply voltage.

The supply voltage has the most significant effect on the energy consumed. Its minimum value is often dictated by the maximum clock speed at which the system is to be run. The greater the clock speed, the greater the minimum supply voltage must be.

Time to execute a given function in a computing system, and therefore its performance, is often approximated by $I*CPI*T$, where $I$ is the instruction count needed to execute the function, CPI is the average number of clock cycles per instruction, and $T$ is the clock cycle time. In ILP computing systems these quantities are reduced by several means: use of superscalar techniques to reduce CPI, use of pipelining to reduce $T$, enhancing the locality of communications through the use of memory hierarchies to reduce $T$, use of optimizing compilers to reduce $I$ and CPI, etc.

Viewed from an energy consumption perspective, these same techniques can also reduce energy consumption. Techniques employing hardware parallelism allow the system to maintain constant performance while reducing the clock rate. A reduction in the clock
rate allows the system to be run at a reduced supply voltage. Memory hierarchies reduce the capacitance that must be switched when accessing information, and optimizing compilers reduce the number of transitions or more fully utilize the hardware parallelism. These techniques, even though they reduce N, C, or V, may also cause one or more of the other factors to increase. Thus their effectiveness in reducing energy consumption depends on the degree to which these two effects balance out.

Chandrakasan, et al. [1] investigated the issue of using parallelism to reduce energy consumption. In their paper they examine the energy consumption of a simple data path while varying its degree of parallelism and their estimate of the complexity of the associated control circuitry. Their results showed that for a 2μm technology, keeping performance of the data path constant by lowering the clock rate, energy consumption could be reduced by a factor of 9 with a degree of parallelism of 10 and a reduction of the supply voltage to 1.4V.

There is evidence in commercial products to support their general conclusions. Examination of several of the leading CPU's for moderate performance, low power applications [2],[3],[4], reveals that each employs a fair degree of pipelining and/or parallelism. These same processors, implemented using little or no parallelism, could easily run at a higher clock rate to achieve the same level of performance. The fact that this simpler implementation was not chosen suggests that parallelism does affect power consumption.

Chandrakasan, et al., appropriately note that to achieve the full energy savings there must be a commensurate degree of parallelism in the software being executed on the data path. Most applications will not have the degree of parallelism needed to fully utilize a data path with degree of parallelism of 10 and so the optimum degree of hardware parallelism will be less. The optimum degree of parallelism for any given application depends upon the degree of parallelism in the application and the degree to which that parallelism can be made manifest by the compiler.

Evaluation of ILP compiler techniques for their ability to reduce energy consumption by enhancing the exploitable parallelism in the application has not occurred. However, initial investigation has been done into the effects of rescheduling of instructions on the number of logic transitions that occur [5]. This investigation has revealed that by rescheduling of instructions to minimize the number of logic transitions within the control logic, switching activity can be reduced by 20-30%. The study did not consider the effects of rescheduling on the amount of switching activity on the CPU data bus caused by the alteration in the sequence of instructions that is fetched across it.

To achieve the high performance that ILP processors are capable of requires a memory subsystem of commensurate performance. Often memory hierarchies are employed to achieve this performance. Memory hierarchies also are capable of reducing energy consumption because they spatially localize communications, thereby reducing the capacitance that is switched.

Past research has investigated optimum cache design to support maximum performance. Bunda, et al. [6] have examined cache design for minimum energy consumption. Many of the standard cache design techniques still apply, however the optimum design points differ. For example, their results indicate that smaller line sizes are better. In addition, it was found that the addition of a small buffer between the cache and CPU to hold the most recently accessed line can significantly reduce energy consumption by reducing the frequency with which the full cache must be accessed.
Another approach to reducing energy consumption lies in the choice of the instruction set architecture (ISA) and encoding. RISC instruction sets have been shown to be more amenable to superscalar and pipelining techniques. This in itself suggests that they may be more energy efficient. However, most instruction sets tend to have 32 bit instructions. Work by Bunda, et al. [6], has shown that a RISC instruction set with 16 bit instructions can significantly reduce the memory traffic associated with fetching the instruction stream with little performance degradation and thus save energy. Specifically, the results show that on average instruction traffic is reduced by 35% while performance ranges from 13% slower with 0 wait state memory to 19% greater for 3 wait state memory on a system with no cache.

There is some question as to what style of architecture best exploits instruction level parallelism while minimizing power consumption. Processors such as the ARM [2], and Hitachi SH7xxx [3] utilize conventional architectures, whereas the Hobbit [4] employs a stack architecture. It is claimed in [4] that the stack architecture enables a more energy efficient memory system to be used, namely through a smaller data cache, that provides the throughput and latency required to support a machine pipelined to the degree of Hobbit.

In conclusion, recent research has shown that there are a variety of performance enhancing techniques used in ILP processors that have the potential to reduce energy consumption. They either reduce the amount of capacitance being switched or permit the system to maintain performance but operate at a reduced supply voltage. The key to their effectiveness lies in the balance between this energy savings and the energy increase that results from any added circuitry or increased activity in existing circuits.

Except for [1], none of the previous work considers the effect on energy consumption that improved performance has by permitting a lowering of the supply voltage. It can be reasonably asserted that not all increases in performance of a computing system will permit the system to maintain performance but run at a lower supply voltage. There are natural breakpoints in the supply voltage that are being standardized upon, such as 5V, 3.3V, 2.2V, etc.

Thus if the technique does not improve performance sufficiently to allow the system to be run at the next lowest breakpoint, then there will be no energy savings. However, one must view the problem in the context of combining several techniques together that give a sufficient performance boost to allow the next lower level of supply voltage to be used.

The challenge in this area is to not only devise techniques that save energy, but specifying the conditions under which energy is saved. The difficulty in determining the effectiveness of any technique lies in the implementation dependence of energy consumption and upon whether or not the system is capable of utilizing the improved performance to permit a reduction in the supply voltage.


