The recent times are known as the dark silicon era. Memory subsystem consumes a major part of energy and so it is imperative to evolve them into energy efficient memories. In the past few years, new memories such as resistive memories or nonvolatile memories have emerged. They are inherently energy efficient and are promising candidates for the future memory devices. However, the application and program layer is not aware of the new memory and is not specifically optimized for energy efficiency. In this thesis, we propose compiler optimization and software testing methods to optimize programs for energy efficiency. Our techniques provide cross layer support to fully utilize the advantages of the energy efficient memories. In most of our works, we assume a resistive technology based hybrid memories as L1 data cache, L2, L3 and main memory level. We propose a new virtual memory design (EnVM) that is aware of resistive technology based hybrid caches. EnVM is based on the memory access behavior of a program and can control the data allocation to the caches. The merits of EnVM diminish at the main memory level, as the size of basic data unit differs from caches. Caches address cache line size data whereas main memory addresses a page which is much larger. We propose a new operating system assisted page addressing mechanism that accounts for cache line size data even in the main memory level. Thus, we magnify the effects of hybrid memory at the main memory level. The next challenge is a characteristic of the energy efficient memories that makes them prone to errors (bitflips). This is not only true for the resistive memories, undervolted memories also exhibit such characteristics. Adapting error detection and correction mechanisms often offsets the gain in power consumption. We propose a framework that exploits the inherent error resiliency of some application to solve this issue. Instead of mitigating, it allows errors if the final output is within a given Quality of Service (QoS) range. Thus, it is possible to run such applications on the energy efficient memories without having to provide error correction support. The above framework, based on a dynamic program testing accrues a large search space to find an optimal approximation configuration for a given program. The running time of the analysis and bookkeeping overheads of such techniques scales linearly with increase in program size (lines of code), so we also propose a static code analysis which deduces
accuracy measures for program variables to achieve a given QoS. This compile time framework complements the dynamic testing schemes and can improve their efficiency by reducing the search space. Overall, in this thesis, we show that with proper support from the software stack, it is possible deploy energy efficient memories in the current memory hierarchy and achieve remarkable reduction in power consumption without compromising performance.