Reconfigurable computing systems can change the functionality and structure of their components to improve the resource efficiency. Many existing architectures must be programmed in assembly, or a compiler does not provide full automation. Usually, a compiler is customized to a specific reconfigurable system developed for a certain application domain.

This thesis presents a unified hardware/software approach, where compiler-driven dynamic reconfiguration selects from a fixed set of modes known to the compiler. Such modes are denoted as *reconfigurable architectural variants* or briefly *variants*. Each variant relies on matching program analysis and represents optimal machine configurations for certain application domains.

A prominent example is to reconfigure between different parallelization paradigms like SIMD or MIMD. Given a program that exhibits both regular and non-regular structures, the compiler can determine the best execution mode by analyzing the parallelism.

Reconfiguring the connections between ALUs and register banks allows to exploit more physical registers than architecturally available. In a multi-core, a processor can use registers of other processors temporarily to avoid spilling or communicate efficiently employing some registers in a shared manner.

This thesis concentrates on switching between SIMD and MIMD execution as well as reconfiguring register connections. Both opportunities have been evaluated using a multi-core of four-tightly coupled processors.