In this paper we investigate the problem of code generation for address computation for DSP processors. This work is divided into four parts. First, we propose a branch instruction design which can guarantee minimum overhead for programs that make use of implicit indirect addressing. Second, we give a formulation and propose a solution for the problem of allocating address registers (ARs) for array accesses within loop constructs. Third, we describe retargetable approaches for auto-increment (decrement) optimizations of pointer variables, and loop induction variables. Finally, we use a graph coloring technique to allocate physical ARs to the virtual ARs used in the previous phases. The results show that the combination of the above techniques considerably improves the final code quality for benchmark DSP programs.