In the last years, several approaches were proposed to improve embedded systems performance by extending base processors to fit specific applications performance demands. Although the majority of the contributions focus specially on the architectural challenges, which range from completely reconfigurable hardware to custom ASICs, every work faces a common issue: the need of the extraction, analysis and transformation of code patterns from application source, in order to implement them in special hardware units. In this paper we present Pattlib, a library of C functions and a file format specifically designed to manipulate and store instruction patterns, binding software representations to hardware descriptions. As an intermediate pattern representation, Pattlib fits as a common denominator among a compiler, hardware generation tools and pattern manipulation tools, allowing for the highest modularization of the design flow of extensible processors. The paper also presents the results of our investigation of code patterns that are suitable for becoming new instructions, extracted from the Mediabench and MiBench benchmarks. Our approach was able to find patterns that occur in up to 15 applications, exposing code regularity.