A popular software attack on a program is by transferring the program control to malicious code inserted into the program. Control Flow Integrity (CFI) check has been proposed as a detection mechanism for control flow deviation. In the context of embedded processors, we propose a novel approach to implement CFI to detect and stall under a control flow attack. We exploit the unused bits in an instruction word to embed a label that can be used to check CFI during runtime. Given a control flow graph, we embed a unique label in each instruction in a basic block such that a given property is satisfied by labels along a valid control flow edge. For example, the hamming distance between any two basic blocks in a legal path is less than 5 and in illegal paths, it is greater than 5. In a five stage processor pipeline, when an instruction is fetched, its label is checked against prior instruction’s label for the known property (i.e., hamming distance of 5). We implemented the proposed approach in the SimpleScalar toolset and validated on seven (7) embedded application benchmarks chosen from MiBench benchmark suite.