1	for (i=0; i<N; i++) c[i] = c[i+1] + b[i]; Always vectorizable
2	for (i=0; i<N; i++) c[i] = c[i-1] + b[i]; NOT vectorizable (loop-carried dependency distance 1)
3	for (i=0; i<N; i++) if (b[i]!=0) c[i] = a[i] / b[i]; Vectorizable Requires support for conditional vector execution
4	while (b[i]!=0) {c[i] = a[i] / b[i]; i++} Partially vectorizable (vector search for zero, set vl, vector divide) Fully vectorizable with speculative vector support
5	for (i=0; i<N; i++) t += a[i] * b[i]; Partially vectorizable by using tree reduction (vector temporary) Fully vectorizable if reduction permutation supported
6	for (i=0; i<N; i++) c[i] = a[d[i]] + b[i]; Vectorizable Requires support for indexed instructions
7	for (i=0; i<N; i++) c[d[i]] = a[i] + b[i]; Vectorizable if - d[i] elements are disjoint OR - HW executes vector stores in order
8	for (i=0; i<N; i++) c[d[i]] += a[i] + b[i]; Vectorizable ONLY if d[i] elements are disjoint
9	for (i=0; i<N; i++) c[2*i] = a[2i+1]; Vectorizable with strided accesses

Comments: