EE Quals 2005 Name: Christoforos Kozyrakis

Vector architectures use instructions (load/store and arithmetic) that define operations on arrays of numbers (vectors). The following vectorizable loop:

for (i=0; i<64; i++) c[i] = a[i] + b[i];

can be expressed with two vector loads for a and b, an element-wise vector add, and a vector store for c. Each instruction specifies 64-bit independent element operations. A vector processor typically executes multiple of these operations per cycle (e.g. 8 elements adds per cycle for the vector add in our example).

For vector processors to be useful, the loop must be vectorizable. For the following loops:

- Identify if they are vectorizable: yes/no, why, how, under which conditions, or with what hardware or instruction set extensions/requirements…

You can discuss the loops in any order you want. Assume N is large (100s or 1000s).

1	for (i=0; i<N; i++)	c[i] = c[i+1] + b[i];
2	for (i=0; i<N; i++)	c[i] = c[i-1] + b[i];
3	for (i=0; i<N; i++)	if (b[i]!=0) c[i] = a[i] / b[i];
4	while (b[i]!=0)	{ c[i] = a[i] / b[i]; i++; }
5	for (i=0; i<N; i++)	t += a[i] * b[i];
6	for (i=0; i<N; i++)	c[i] = a[d[i]] + b[i];
7	for (i=0; i<N; i++)	c[d[i]] = a[i] + b[i];
8	for (i=0; i<N; i++)	c[d[i]] += a[i] + b[i];
9	for (i=0; i<N; i++)	c[2*i] = a[2i+1];