SIMD systems

- Single instruction, multiple data, or SIMD, systems are parallel systems. As
  the name suggests, SIMD systems operate on multiple data streams by applying thesame instruction to multiple data items.
- SIMD system can be thoughtof as having a single control unit and multiple ALUs. An instruction is broadcastfrom the control unit to the ALUs, and each ALU either applies the instruction to thecurrent data item, or it is idle. As an example, suppose we want to carry out a "vectoraddition."
- That is, suppose we have two arrays x and y, each with n elements, and wewant to add the elements of y to the elements of x:

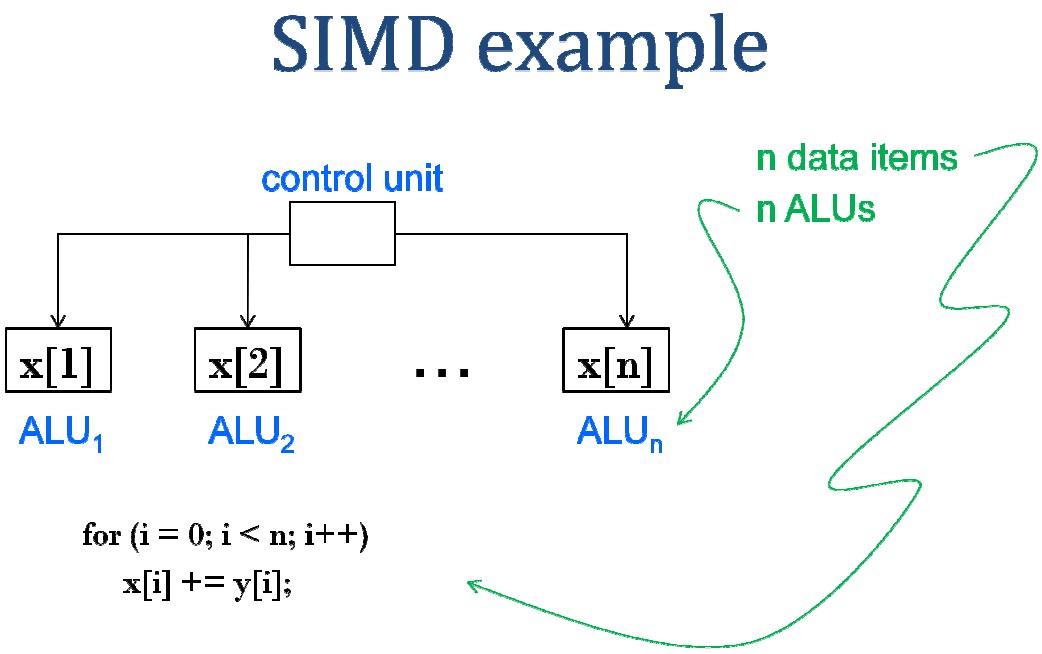
```
for (i = 0; i < n; i++)
        x[i] += y[i];
```

- Suppose SIMD system has n ALUs. Then we could load x[i] and y[i] into the ith ALU, have the ith ALU add y[i] to x[i], and store the result inx[i]. If the system has m ALUs and m < n, we can simply execute the additions inblocks of m elements at a time.
  - For example, if m D 4 and n D 15, we can first add elements 0 to 3, then elements 4 to 7, then elements 8 to 11, and finally elements 12 to 14.Note that in the last group of elements in our example—elements 12 to 14—we'reonly operating on three elements of x and y, so one of the four ALUs will be idle.The requirement that all the ALUs execute the same instruction or are idle canseriously degrade the overall performance of a SIMD system.

- SIMD Arch have significant DLP. Single Instruction can launch many data opns
- SIMD is more energy efficient than MIMD
  - MIMD needs to fetch and execute one instruction per data opns.
  - SIMD is more attractive for PMDs.
  - Advantage of SIMD over MIMD

- Programmer thinks sequential execution yet achieves parallel speedup by having parallel data operations

- Parallelism achieved by dividing data among the processors.
- Applies the same instruction to multiple data items.
- Called data parallelism.
- SIMD has 3 variations:
- Vector Architectures
- Allows pipelined execution of many data operations
- SIMD MMX
- Allows simultaneous parallel data operations that support Multimedia applications.
- GPU Architectures
- They offer higher performance than traditional multicore
- They have system processor, system memory & graphics memory.



- What if we don't have as many ALUs as data items?
- Divide the work and process iteratively.
- Ex. m = 4 ALUs and n = 15 data items.

Round3	ALU_1	ALU_2	ALU_3	ALU_4
1	X[0]	X[1]	X[2]	X[3]
2	X[4]	X[5]	X[6]	X[7]
3	X[8]	X[9]	X[10]	X[11]
4	X[12]	X[13]	X[14]	

- All ALUs are required to execute the same instruction, or remain idle.
- In classic design, they must also operate synchronously.
- The ALUs have no instruction storage.
- Efficient for large data parallel problems, but not other types of more complex parallel problems.
- SIMD drawbacks