Vector Processor : Multiple Lanes

- A critical advantage of.a vector instruction set is that it allows software to pass a large amount of parallel work to hardware using only a single short instruction. A single vector instruction can include scores of independent operations yet be encoded in the same number of bits as a conventional scalar instruction.

- The parallel semantics of a vector instruction allow an implementation to execute these elemental operations using a deeply pipelined functional unit, as in the VMIPS implementation we've studied so far; an array of parallel functional units; or a combination of parallel and pipelined functional units.

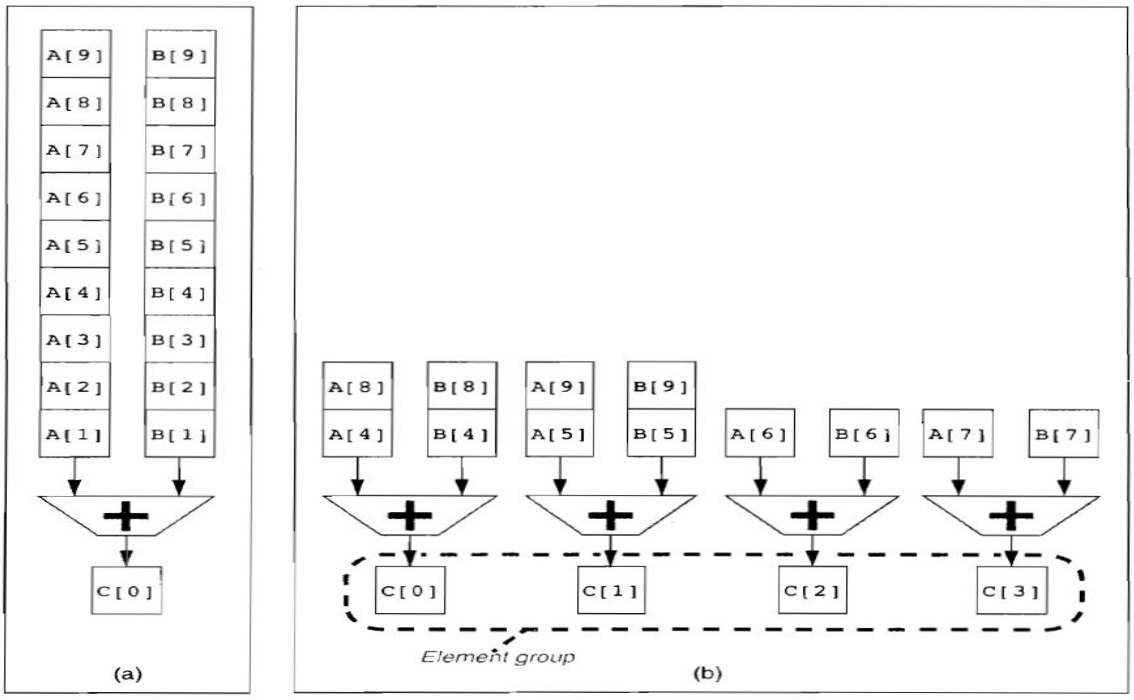

Fig:Using multiple functional units to improve the performance of a single vector add instruction,

C = A + B.

- The vector processor (a) on the left has a single add pipeline and can complete one addition per cycle. The vector processor (b) on the right

has four add pipelines and can complete four additions per cycle. The elements within a single vector add instruction are interleaved across the four pipelines. The set of elements that move through the pipelines together is termed an element group

- Going to four lanes from one. lane reduces the number of clocks for a chime from 64 to 16. For multiple lanes to be advantageous, both the applications and the architecture must support long vectors; otherwise, they will execute so quickly that you'll run out of instruction bandwidth, , requiring ILP techniques (see Chapter 3) to supply enough vector instructions.
- Each lane contains one portion of the vector register file and one execution pipeline from each vector functional unit. Each vector functional unit executes vector instructions at the rate of one element group per cycle using multiple pipelines, one per lane. The first lane holds the first element (element 0) for all vector registers, and so the first element in any vector instruction will have its source and destination operands located in the first lane.
-

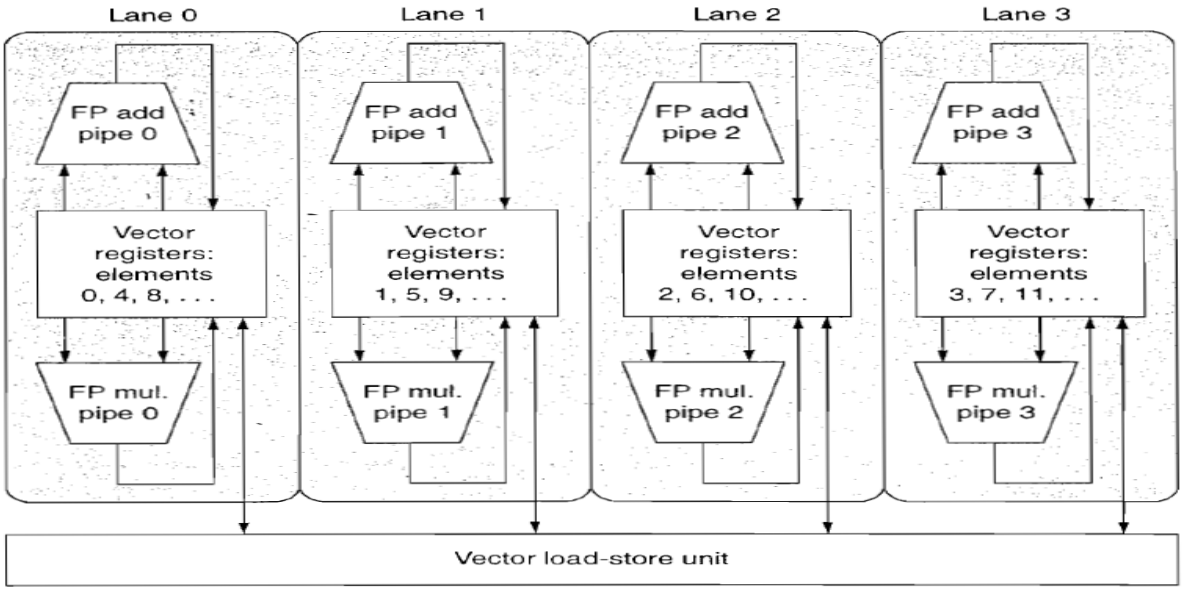

Fig: Structure of a vector unit containing four lanes

- The vector register storage is divided across the lanes, with each lane holding every fourth element of each vector register. The figure shows three vector functional units: an FP add, an FP multiply, and a load-store unit. Each of the vector arithmetic units contains four execution pipelines, one per lane, which act in concert to complete a single vector instruction. Note how each section of the vector register file only needs to provide enough ports for pipelines local to its lane.
- This allocation allows the arithmetic pipeline local to the lane to complete the operation without communicating with other lanes. Accessing main memory also requires only intralane wiring. Avoiding interlane communication reduces the wiring cost and register file ports required to build a highly parallel execution unit, and helps explain why vector computers can complete up to 64 operations per clock cycle (2 arithmetic units and 2 load/store units across 16 lanes).

- Adding multiple lanes is a popular technique to improve vector performance as it requires little increase in control complexity and does not require changes to existing machine code. It also allows designers to trade off die area, clock rate, voltage, and energy without sacrificing peak performance. If the clock rate of a vector processor is halved, doubling the number of lanes will retain the same potential performance.