# Vector Architecture

- The most efficient way to execute a vectorizable application is a vector processor. → Jim Smith
- Vector architectures grab sets of data elements scattered about memory, place them into large, sequential register files, operate on data in those register files, and then disperse the results back into memory. A single instruction operates on vectors of data, which results in dozens of register-register operations on independent data elements.
- These large register files act as compiler-controlled buffers, both to hide memory latency and to leverage memory bandwidth. Since vector loads and stores are deeply pipelined, the program pays the long memory latency only once per vector load or store versus once per element, thus amortizing the latency over, say, 64 elements.

## VMIPS

- This processor, which is loosely based on the Cray-1. This instruction set architecture VMIPS; its scalar portion is MIPS, and its vector portion is the logical vector extension of MIPS. The primary components of the instruction set architecture of VMIPS are the following:
- Vector registers-Each vector register is a fixed-length bank holding a single vector. VMIPS has eight vector registers, and each vector register holds 64 elements, each 64 bits wide. The vector register file needs to provide enough ports to feed all the vector functional units. These ports will allow a high degree of overlap among vector operations to different vector registers. The read and write ports, which total at least 16 read ports and 8 write ports, are connected to the functional unit inputs or outputs by a pair of crossbar switches.
- Vector functional units-Each unit is fully pipelined, and it can start a new operation on every clock cycle. A control unit is needed to detect hazards, both structural hazards for functional units and data hazards on register accesses. Figure below shows that VMIPS has five functional units.
- For simplicity, we focus exclusively on the floating-point functional units

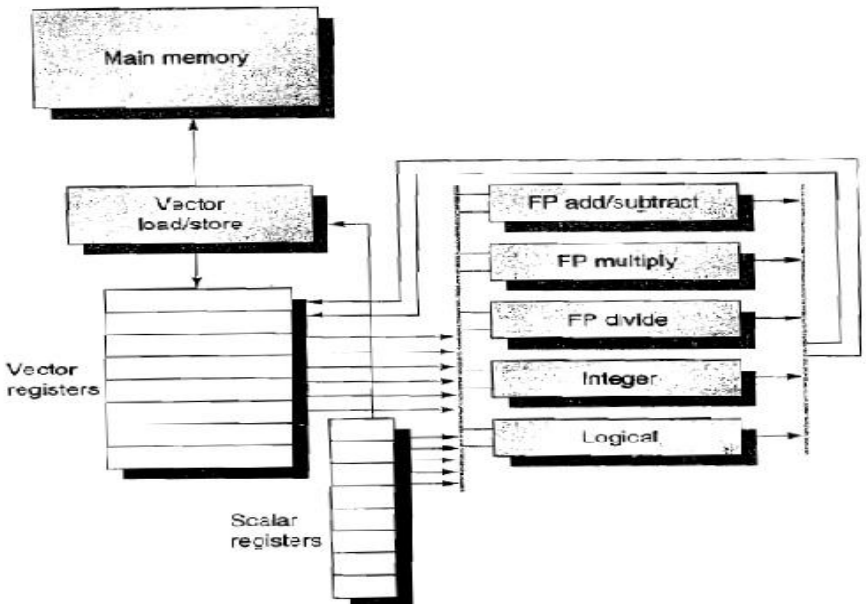


Fig: Basic structure of Vector Architecture VMIPS

- This processor has a scalar architecture just like MIPS. There are also eight 64-element vector registers, and all the functional units are vector functional units. . The figure shows vector units for logical and integer operations. The vector and scalar registers have a significant number of read and write ports to allow multiple simultaneous vector operations. A set of crossbar switches (thick gray lines) connects these ports to the inputs and outputs of the vector functional units.

- Vector load/store unit-The vector memory unit loads or stores a vector to or from memory. The VMIPS vector loads and stores are fully pipelined, so that words can be moved between the vector registers and memory with a band width of one word per clock cycle, after an initial latency. This unit would also normally handle scalar loads and stores.

- A set of scalar registers-Scalar registers can also provide data as input to the vector functional units, as well as compute addresses to pass to the vector load/store unit. These are the normal 32 general-purpose registers and 32 floating-point registers of MIPS. One input of the vector functional units latches scalar values as they are read out of the scalar register file.