Architecture and Operation of Vector Computer CONVEX 3820

SLIDES

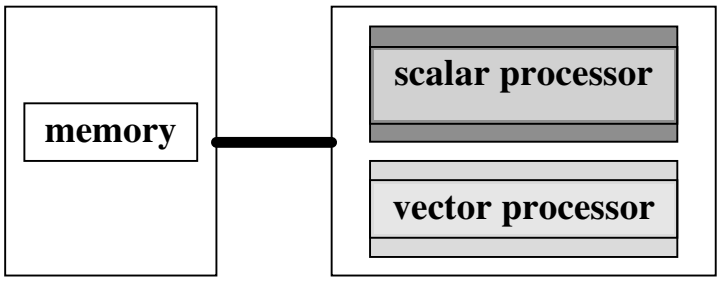
Kandó Kálmán Polytechnic
Institute of Computer Technology, Székesfehérvár
and
University of Athens

Etelka Kovács, Senior Lecturer
Ágnes Burián, Senior Lecturer

Vector computers

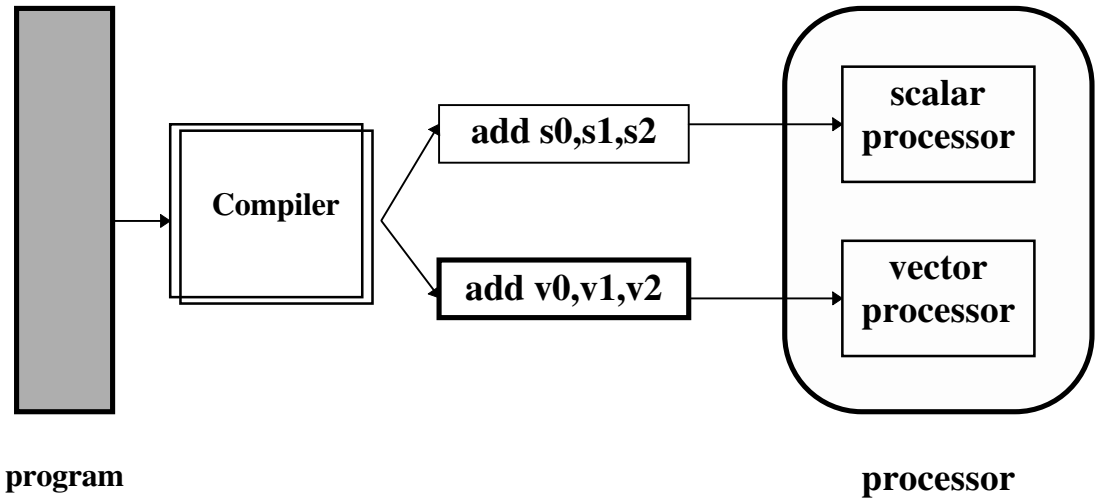
A vector computer generally consists of two different processors:

- **Scalar processor** **Traditional mainframe/mini like system**
- **Vector processor** **Fast system for specific operations**



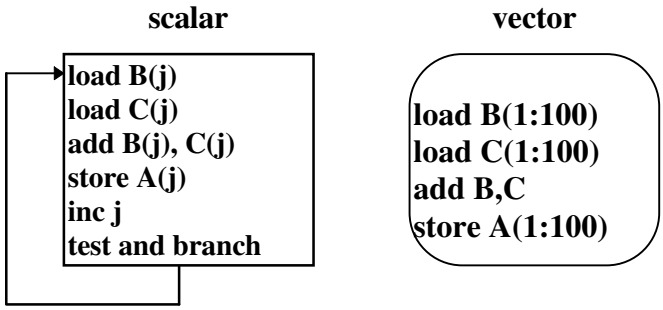
What is vectorization:

- Vectorization is the translation of a program (fragment) to instructions that will be executed in the vector unit


Compiler
add s0,s1,s2
add v0,v1,v2
scalar processor
vector processor
program
processor

Vectorization example:

```
DO J = 1, 100
    A(J) = B(J) + C(J)
END DO
```



600 instruction slow add

4 instruction fast add

BASICS VECTOR ARCHITECTURE

A vector processor typically consists of ordinary pipelined scalar unit plus a vector unit. All functional units within the vector unit have a latency of several clock cycles.

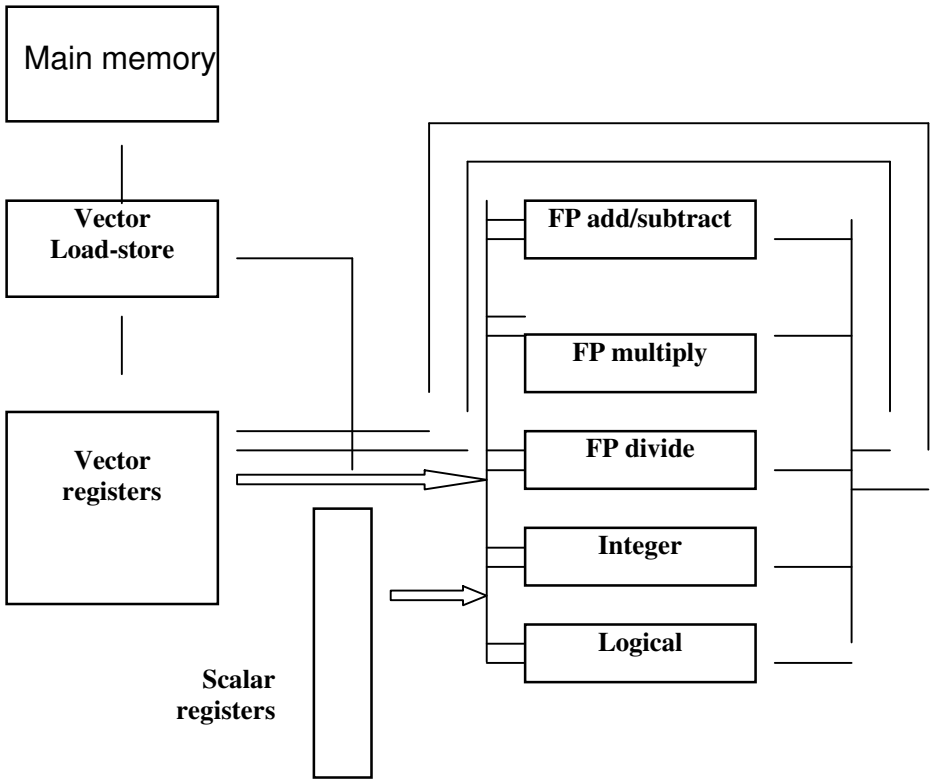
Two primary types of architectures for vector processors:

- **vector-register** processor,
- **memory-memory vector** processor.

In a **vector-register processor**, all vector operations –except load and store – are among the vector registers.
(for example **the mini-supercomputers Convex C-1 and C-2**)

In a **memory-memory vector** processor, all vector operations are memory to memory.

The basic structure of a vector-register architecture:



The primary components of instruction set architecture of vector processors:

A./ Vector registers:

- Each vector registers is a fixed-length bank holding a single vector
- The vector processor has eight vector registers
- Each vector register holds 64 elements
- Each vector register must have at least two read ports and one write ports

B./ Vector functional units:

- Each unit fully pipelined and can start a new operation on every clock cycle
- A control unit is needed to detect hazards, both from conflicts for the functional units (structural hazards) and from conflicts for register accesses (data hazards)
- This vector processor has five functional units

C./ Vector load-store unit:

- A vector memory unit loads or stores a vector to or from memory
- Fully pipelined
- This unit would also normally handle scalar loads and stores

D./ A set of scalar registers:

- Scalar registers can also provide data as input to the vector functional units,
- compute addresses to pass to the vector load-store unit.
- The normal 32 general-purpose registers and 32 floating-point registers

The characteristics some typical vector processors

Processor	Year announced	Clock rate (MHz)	Registers	Elements per register	Functional units	Load-store units
CRAY-1	1976	80	8	64	**6: add, multiply, reciprocal, integer add, logical, shift**	1
CRAY-X-MP **CRAY Y-MP**	1983 1988	120 166	8	64	**8: FP add, FP multiply, FP reciprocal, integer add, 2 logical, shift, population count/parity**	2 loads 1 store
CRAY-2	1985	166	8	64	**5: FP add, FP multiply, FP reciprocal/sqrt, integer (add, shift, population count), logical**	1
Fujitsu VP100/200	1982	133	8-256	32-1024	**3: FP or integer add/ logical multiply, divide**	2
Hitachi S810/820	1983	71	32	256	**3: 2 integer add/logical, 1 multiply-add, and 1 multiply/ divide-add unit**	4
CONVEX C-1	***1985***	***10***	***8***	***128***	***4: multiply, add, divide, integer/logical***	***1***
NEC SX/2	1984	160	8+8192	256 variable	**16: 4 integer add/logical, 4 FP multiply/ divide, 4 FP add, 4 shift**	8
DLXV	1990	200	8	64	**5: multiply, add, divide, integer add, logical**	1
Cray C-90	1991	240	8	128	**8: FP add, FP multiply, FP reciprocal, integer add, 2 logical shift, population count/parity**	4
CONVEX C-4	***1994***	***135***	***16***	***128***	***3: each is full integer, logical, and FP (including multiply add)***	
NEC SX/4	1995	400	8+8192	256 variable	**16: 4 integer add/logical, 4 FP multiply/ divide, 4 FP add, 4 shift**	8
CRAY J-90	1995	100	8	64	**4: FP add, FP multiply, FP reciprocal, integer/logical**	
CRAY T-90	1996	500	8	128	**8: FP add, FP multiply, FP reciprocal, integer add, 2 logical shift, population count/parity**	4

THE CONVEX C SERIES ARCHITECTURE

- Data representations
- Data operations
- Register sets
  - General registers
  - Communication registers
- Address space
  - Physical address space
  - I/O address space
  - Logical (virtual) address space
- Process structures
- Timers
- Memory management
  - Memory protection mechanisms
  - Caches
- Multiprocessing mechanisms
  - Automatic self-allocating processors
  - Parallel processing mechanisms
  - Forking and spawning mechanisms
- Exception system mechanisms
- Interrupt system mechanisms
- Instruction set

The CONVEX C Series architecture incorporates the following features:

- An integrated vector processor incorporated within the system for high speed operation
- A full range of fixed floating point data types
- A total capacity of four Gbytes of virtual memory. Two Gbytes are available to support large user programs and data, and two Gbytes are available to the operating system.
- Large, high-speed register sets (address, scalar, and vector) that support high-performance operation for address calculations in parallel with scalar and vector calculations.
- Communication registers and multiprocessing structures (in the multiprocessing C Series CPUs).
- Multilevel protection systems that support and separate users, thereby enhancing system reliability and increasing the performance of operating system functions.

Brief summaries about CONVEX CPUs

❶ Data representations and operations

❷ General registers

❸ Physical address space

❹ Logical (virtual) address space

❺ Communication registers

❻ Process structures

❼ Memory and cache management

❽ Multiprocessor management

❾ Timers

❿ Operating system interrupts

❶❶ Operating system exceptions

❶ Data representations and operations

There are three binary numeric data representations:
- Signed fixed point integer
- Unsigned fixed point integer
- Floating point

- Byte – 8 bits
- Halfword – 16 bits
- Word – 32 bits
- Longword – 64 bits

The CONVEX CPUs support both native and LEEE standard floating point number representations in two formats:
- Single-precision word (32 bits)
- Double-precision longword (64 bits)

❷ General registers

- **Address registers** (eight 32-bit registers: A0-A7);
  - A0 is the stack pointer /SP/
  - A3, A4, and A5 are implicitly used by some trap handlers (for example, page fault, system exception, and so forth)
  - A5 are implicitly used by some additional instructions
  - A6 is the argument pointer /AP/
  - A7 is the frame pointer /FP/
- **Scalar registers** (eight 64-bit registers: S0-S7);
  The S register can contain logical, fixed point integer, or floating point operands.
- **Vector registers** (8 vectors; each containing 128 64-bit register elements)
  (see in the following section)

The **two status registers** are:
- Program counter (PC)
- Processor status word (PSW)

The **three privileged flags** are
- Interrupt on (ION)
- Real-time interrupt on (RT_ION)
- Vector valid flag (VV)

❸ Physical address space

The physical address space in all C Series CPUs spans physical memory. Data access at the program level uses virtual addresses, however, these virtual addresses are translated to physical addresses in hardware.

❹ Logical (virtual) address space

- The memory management unit (MMU) supports the operating system in providing a versatile and reliable virtual memory programming environment..
- The CONVEX C Series architecture provides 4 Gbytes of virtual memory in its virtual address space partitioned into eight 512-Mbyte segments.
- Four segments are allocated to the operating system and four segments to the user. The maximum size of a user program (instruction and data) is limited to 2 Gbytes. The operating system data structures and instructions necessary to manage the user program occupy the remaining 2 Gbytes of virtual storage.

To manage memory, the CONVEX system architecture defines and supports these features:

- **Segment** – A logically contiguous 512-Mbyte block of memory.
- **Segment descriptor register (SDR)** – A 32-bit register that contains the pointer to the first-level page table.
- **Page** – A 4-kbyte contiguous block of memory. The bytes within a page are both virtually and physically contiguous.
- **Page frame** – A page that is stored in physical memory.
- **Page table entry (PTE)** – 32 bit entry interpreted by the memory management unit (MMU) or operating system software to determine whether or not a page is resident in main memory.
- **Page table** – A page that contains page table entries (PTEs). A page table begins on an integral page boundary and is contained in one page frame or less. First-level page tables contain PTEs that have pointers to second-level page tables. Second-level page tables contain pointers to physical page frames or thread-level pages.
- **Address translation units (ATU)** – A programmer-invisible address cache that is maintained in hardware. It contains the most recently used virtual-to-physical address translations.

❺ Communication registers

The communication registers set implements CPU communication and synchronization for the multiprocessing C Series architecture:

- 1 024 locations by 64 bits for C200/C3200 Series CPUs
- 1 024 locations by 64 bits for C3400 Series CPUs
- 4 096 locations by 64 bits for C3800 Series CPUs

❻ Process structures

A CONVEX C Series process consists of one or more threads. A process is controlled by maintaining a process stack. These stack entries are called return blocks and contain vital information for controlling the execution of a process.

❼ Memory and cache management

C Series CPUs have implementation-specific cache management mechanisms. These mechanisms purge the instruction, data, ATU, and PTE caches under certain conditions.

Memory interleave is the process of swapping address bits to determine which physical memory bank to use.

❽ Multiprocessor management

Multiprocessing is defined as the creation and scheduling of individual processes on any subcomplex. The multiprocessor management hardware incorporated in each C Series architecture CPU provides the operating system and user with a simple and flexible set of instructions for dynamic CPU allocation, deallocation, and communication.

In order to manage multiple CPUs, the CONVEX C Series architecture defines and supports several attributes:
CPU – One physical central processing unit
CPU complex – The entire set of physical CPUs in a configuration
CPU subcomplex – Any subset of a complex
Thread – Any single instruction stream executing within a process
Process – One or more threads executing within a single logical address space

❾ Timers

The multiprocessing C Series architecture provides four timers to permit accurate accounting of CPU execution time and to assist in process scheduling:

- Interval timer – Used interrupt the CPU at a programmable rate (count-down timer)
- Time of century clock – This clock keeps „wall clock time", not user time.
- CPU execution timer – Timer for each process executing.
- Thread timer – Timer for each thread executing.

❿ Operating system interrupts

Interrupts are results of events that occur asynchronously and belong to the system, not to the executing process. When an interrupt occurs, the processor jumps to a particular interrupt handler determined by the interrupt source.

❶❶ Operating system exceptions

Exceptions are invoked when problems occur in a currently executing program, such as arithmetic inconsistencies, address translation faults, or some asynchronous event (such as an interrupt). When an exception occurs, control is transferred to a predetermined address whose value is a function of the exception.

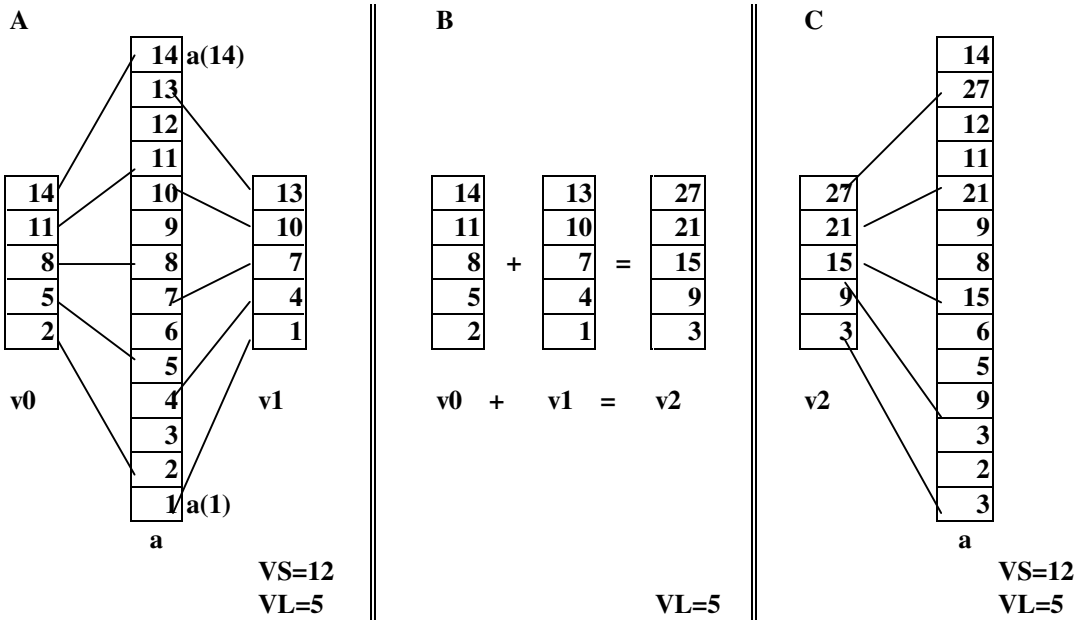
VECTOR HARDWARE

Four types of register are used in vector operations:

- Vector-accumulator (V) register
- Vector-length (VL) register
- Vector-stride (VS) register
- Vector-merge (VM) register

How the CONVEX architecture works

Example:

```
INTEGER A(14), I
DO I=1, 14, 3
   A(I)=A(I+1)+A(I)
ENDDO
```

The code increments every third elements of array and uses the VS, VL and V registers. The next figure shows the vector operations on array A.