Notes-10

VECTOR PROCESSORS

Informal: Consider a pipelined functional unit, e.g. an adder with a 3-stage pipeline. Observer that we are NOT talking about the instruction pipeline; rather, we are considering a pipelined execution unit.

From our earlier discussion (Notes 1-4), we know that a long sequence of identical instructions (implying the same functional unit), with **no data dependence**, executes very efficiently, in a pipeline, with one result emerging every clock cycle. **Vector processors** exploit this feature. A vector is essentially a one-dimensional array of data. A **Vector Instruction** is actually equivalent to a sequence of identical instructions operating on the consecutive elements of the data. Thus, the vector instruction

V4 V2 + V3

Is equivalent to:

V4[1] := V2[1] + V3[1]

V4[2] := V2[2] + V3[2]

V4[3] := V2[3] + V3[3]

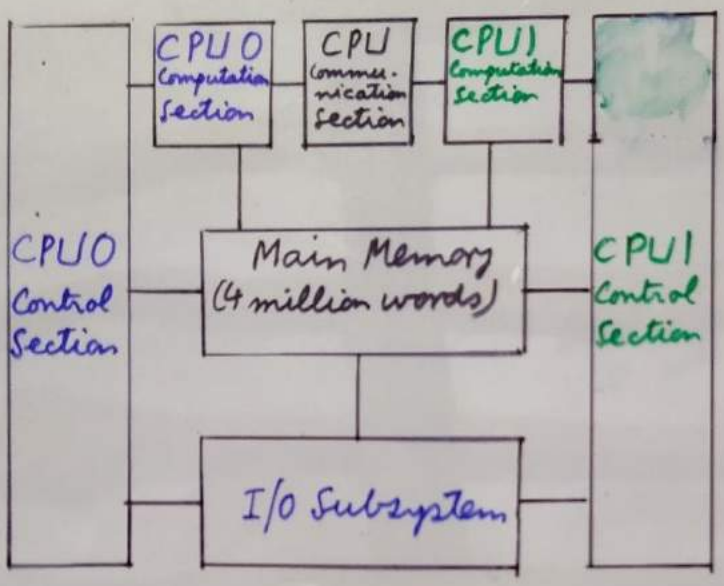
……

V4[n] := V2[n] + V3[n]

Where n is the length (number of elements) of the vectors.

The consecutive elements of V2 and V3 are fed to the vector adder and the result (output) of the adder are the consecutive elements of V4. Evidently, there is **no dependence** between consecutive instructions.

The CRAY X-MP/Model 24 Architecture



- Two identical CPUs, each with a 9.5 ns cycle time.
- The processors share a common main memory and I/O section.
- The control section for each processor manages the instruction buffers, issues instructions, and controls the flow of information within other sections.
- Each CPU has its own computation section consisting of registers and functional units. The registers include address registers, scalar registers, and vector registers. There are 13 pipelined functional units dedicated to performing specific integer and floating point operations.

The Cray X-MP/Model 24 Architecture (cont'd.)

- The communication section includes 3 clusters of shared registers and semaphores used to arbitrate interprocessor communication and shared memory.

- The main memory is shared by the two CPUs. It consists of 4 million words organized into 32 banks. Each word includes 64 data bits and 8 check bits.

- There are between two and four I/O processors included in the system:

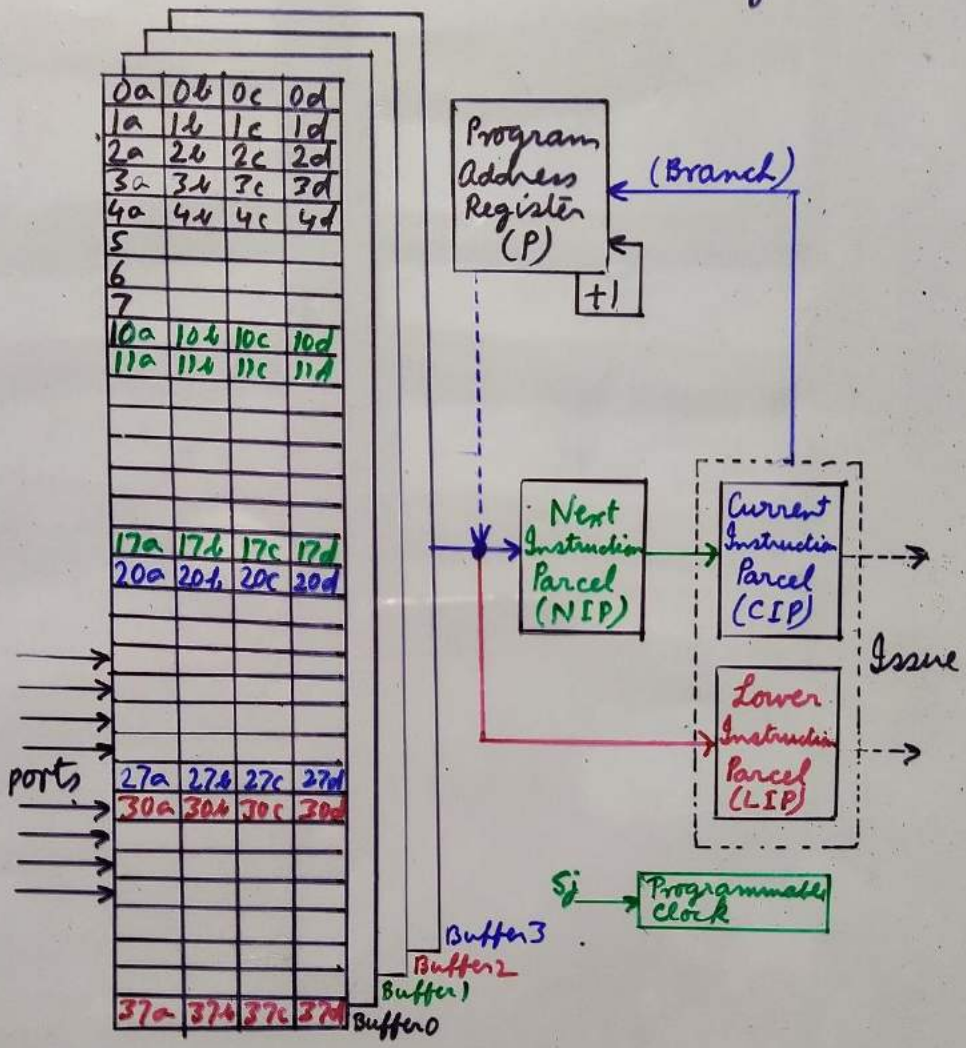
  * MIOP → required in all configurations. manages the front-end interfaces and console.

  * BIOP → handles transfers between main memory and secondary storage devices.

  * DIOP → manages additional (optional) disk controllers.

  * XIOP → controls block multiplexor (optional) channels.

THE CONTROL SECTION of the Gray X-MP



Cray X-MP: Instruction Issue Phase.

- Load and store architecture.
  Functional units require register operands.
  Most instructions do not experience a delay due to operand fetches from memory.
- Instructions are either 16 bits (1 parcel) or 32 bits (2 parcels) long.

1-parcel instructions

- During the instruction fetch, the first parcel of an instruction is transferred from the instruction buffer to the NIP (Next Instruction Parcel Register). This process takes one clock cycle.
- On the next clock cycle the parcel is moved to the CIP (Current Instruction Parcel Register) where it is decoded. The instruction is held in the CIP until it can be issued, that is, until the processor is ready to execute it.
- A one-parcel instruction will need only one clock cycle in the CIP unless there is a conflict because a previously issued instruction is using resources which will be required during the execution of the current instruction. This situation is called a hold condition.

Cray X-MP : Instruction Issue Phase (contd.)

Most of the Cray X-MP instructions are one parcel long. Under optimal conditions, each 1-parcel instruction will spend one clock cycle in the NIP and one clock cycle in the CIP. These operations can overlap so that the next instruction can be brought to the NIP on the same cycle as the previous instruction is brought to the CIP. This works as long as the next instruction is in the current instruction buffer and there are no branches. Under these optimal circumstances, an instruction issues on each clock cycle.

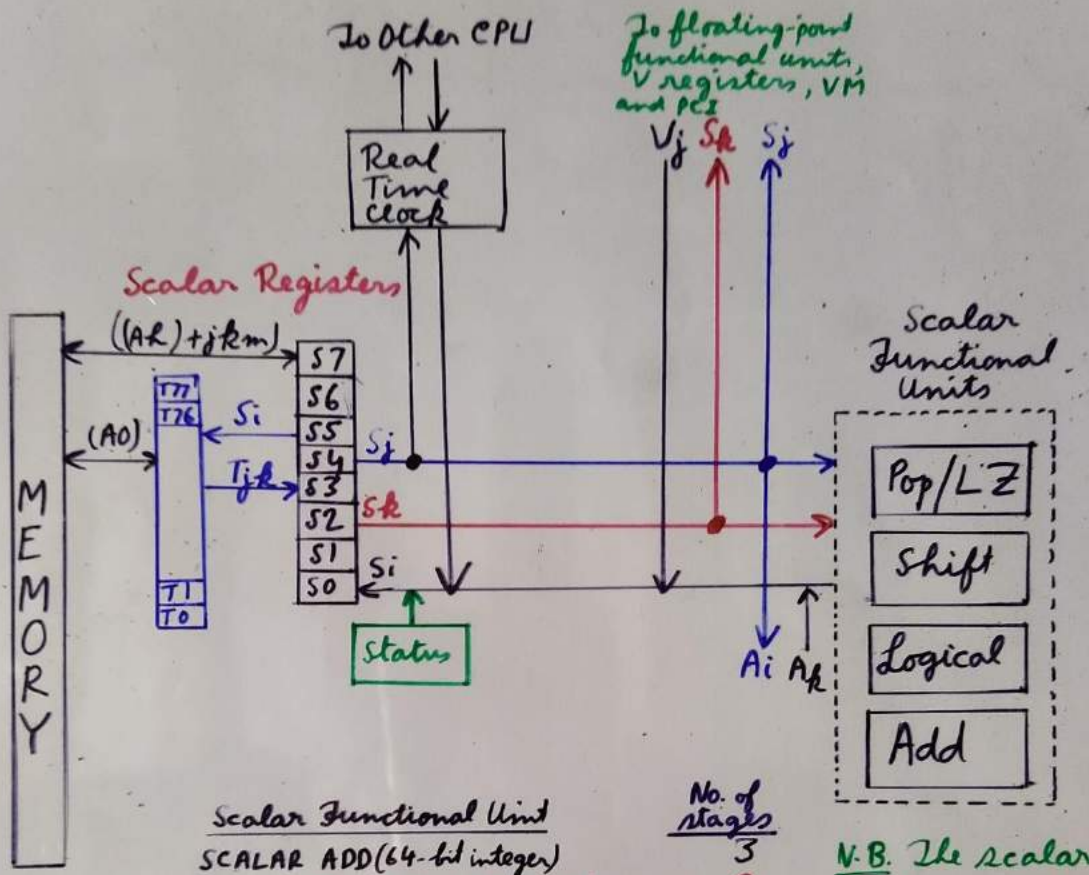
Cray X-MP: Instruction Issue

Branches: When a branch instruction reaches the CIP, it stays there until the instruction completes. No succeeding instructions can be issued until the branch is determined. Thus branch instructions spend their entire instruction cycle in the issue phase and no time in the execution phase.

2-parcel instructions

Two-parcel instructions spend a minimum of two clock cycles in the CIP. In the cycle in which the first parcel moves from the NIP to the CIP, the second parcel is transferred from the instruction buffer to the LIP. The NIP is then filled with a zero (no operation) so that the processor does not issue an instruction on the next cycle.

SCALAR SECTION of the Cray X-MP



Scalar Functional Unit	No. of stages
SCALAR ADD (64-bit integer)	3
SCALAR SHIFT (64-bit logical)	2
SCALAR SHIFT (128-bit logical)	3
SCALAR LOGICAL (64-bit logical)	1
SCALAR POP/PARITY (population or parity)	4
SCALAR POP/PARITY (leading zero count)	3

N.B. The scalar floating point operations are performed in the vector functional unit.

Gray X-MP: Scalar Section [cont'd]

Scalar output operands are reserved

Line	Instruction	
1	S1	S2 + S3
2	S1	S2 + S3
3	S2	S1 + S7
4	S1	S4 + S5

instruction	1	2	3	4	5	6	7	8	9	0	1
1	I	E	E	E							
2				I	E	E	E				
3							I	E	E	E	
4								I	E	E	E

Instruction 2 cannot issue until instruction 1 completes because S1 has already been reserved by instruction 1.

Instruction 3 cannot issue until instruction 2 completes because S1 has been reserved.

However, instruction 4 can issue on the clock cycle after instruction 3 issues because S1 is not reserved by instruction 3.

Cray X-MP: Scalar Section [contd.]

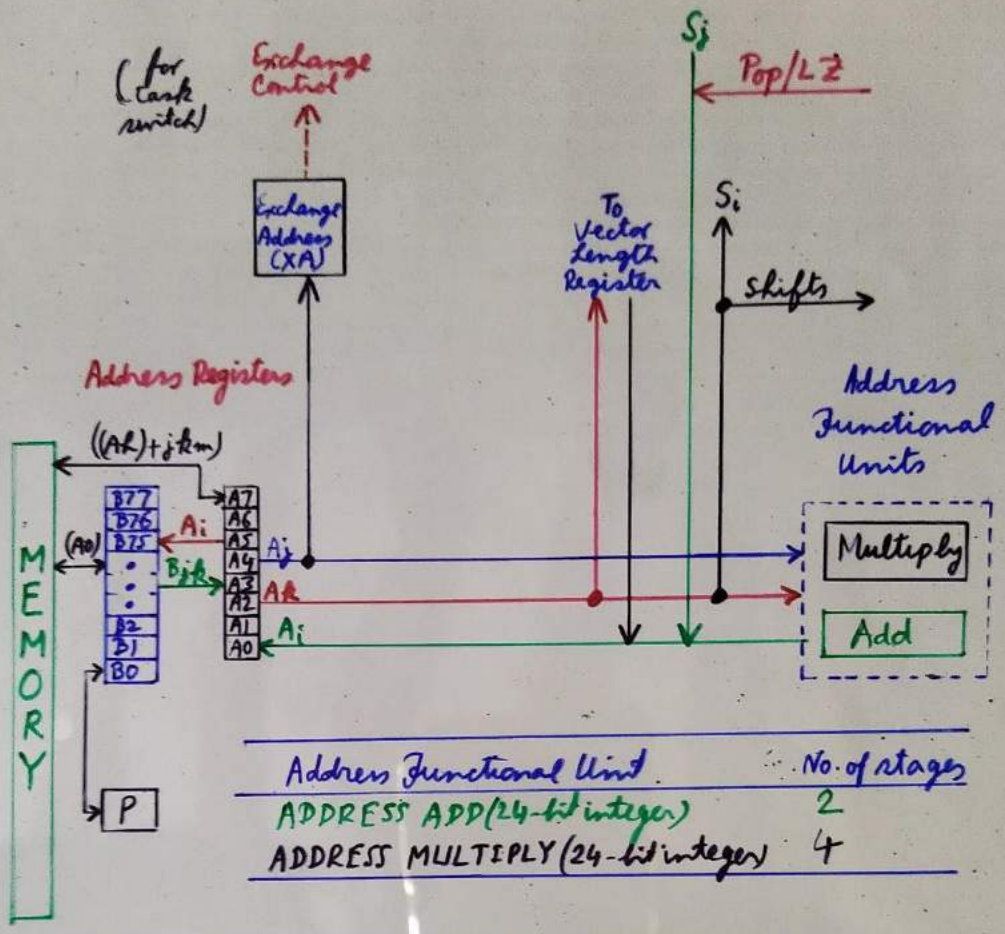
The pipeline may delay instructions because of different execution times

Line	Instruction	Description
1	S1 S1,S2<A2	Shift (S1,S2) left into S1 by A2 places.
2	S3 S3<8	Shift S3 left 8 places.

instruction	1	2	3	4	5
1	I	E	E	E	
2			I	E	E

Instruction 1 takes 3 clock cycles to execute while instruction 2 takes 2 clock cycles to execute. Both instructions are performed by the same functional unit. If instruction 2 issued at clock cycle 2, the results of both instructions would be available at clock cycle 4. Additional hardware would be required to handle multiple results produced by a single functional unit in the same clock cycle. For this reason, a one word shift is held for one clock cycle when it immediately follows a double word shift.

A similar situation occurs when a leading zero count immediately follows a population or parity count in the scalar pop/parity unit.



Address Functional Unit	No. of stages
ADDRESS ADD (24-bit integer)	2
ADDRESS MULTIPLY (24-bit integer)	4

The address section of the Gray X-MP

- Address registers (A0..A7) → 8 24-bit registers
  They hold addresses referenced in memory operations.
  Also used as index registers and for short integer computations.

- B registers (B0..B77) → 64 24-bit registers
  Not connected directly to the functional units.
  Used for saving registers during subroutine linkage or for temporary storage of addresses.

Addresses on the Cray X-MP:

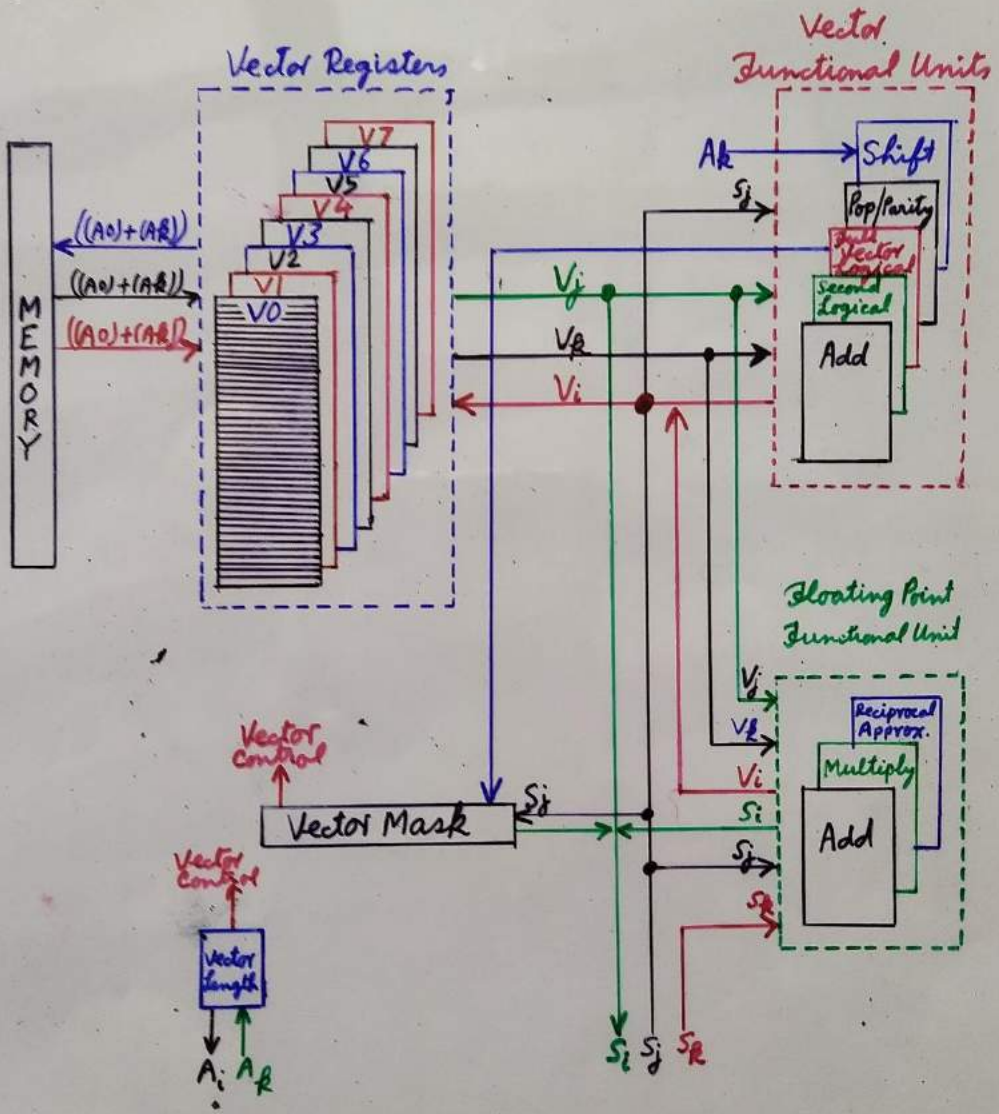
- All memory data is stored in words of 64 bits, and word addresses are specified by 22 bits.
- Instructions take up either 16 bits (one parcel) or 32 bits (two parcels), and parcel addresses are 24-bit wide.
- A and B register addresses are either interpreted as 22-bit word addresses or as 24-bit parcel addresses depending on the instruction in which they occur.

Gray X-MP Address Section

Instructions that use A and B registers
(A few examples)

Syntax	Description	Octal code
Ai exp,Ak	Read A_i from location Ak+exp	10hijkm
exp is an immediate value (jkm)		16-bit
exp,Ak Ai	Store A_i at location Ak+exp	11hijkm
J Bjk	Branch to address in Bjk	0050jk
R exp	Branch to address exp=ijkm, after setting B00 to P+2	007ijkm
Ai Bjk	Set A_i to Bjk	024ijk
Bjk Ai	Set Bjk to A_i	025ijk
Ai Sj	Copy low 24 bits of Sj into A_i	023ij0
Ai VL	Set Ai to VL.	023i01
Ai ZSj	Set A_i to leading zero count of Sj. A_i=64 if j=0.	027ij0
Ai Aj+Ak	Set A_i to $A_j + A_k$	030ijk
Ai Aj*Ak	Set A_i to $A_j * A_k$ (low 24 bits only)	032ijk
Ai PSj	Set Ai to number of ones in Sj (population count)	026ij0

The vector section of the Gray X-MP



Gray X-MP : Vector Section

- A vector is defined as a collection of values which are stored in memory locations spaced a fixed distance apart.
- The spacing of the elements is called the stride of the vector.
- Parallelism and computational efficiency are achieved in a vector processor when a significant portion of a program's data can be mapped into vectors.

The same operations can be performed on all of the elements of a vector with results becoming available on consecutive clock cycles.

Cray X-MP: Basic operation of the Vector Section

- Each processor of the Cray X-MP has 8 vector registers, V0...V7. Each register is 64 words long.
- Since there are no direct connections from main memory to the vector functional units, all vector arithmetic operations are performed through the vector registers.
- A vector is processed by reading its components into the consecutive elements of the vector register starting with element 0 and going through element VL-1. VL is the value of the vector length register when the instruction is issued.
- The source and destination vector registers and the functional units are reserved during vector operations.
- All of the vector functional units are pipelined. The units are designed to perform the same operation on each element of a vector.
- Once the pipeline of a functional unit is filled the unit will produce a result on every clock cycle.

Cray X-MP: Basic Operation of the Vector Section (Contd.)

- There are three phases of the pipelined operation.
- The first phase is called the setup phase. During this phase the functional unit is set to perform the appropriate operation, and the source and destination routes to the vector registers are established. The setup time for all of the vector functional units is 3 clock cycles.
- The second phase is called the execution phase. During each clock cycle in this phase one pair of source elements enters the first stage of the pipeline and the result computed for the previous pair is sent to the next stage.

Vector Functional Unit	No. of Stages
VECTOR ADD (64-bit integer)	3
VECTOR SHIFT (64-bit logical)	3
VECTOR SHIFT (128-bit logical)	4
FULL VECTOR LOGICAL (64-bit logical)	2
SECOND VECTOR LOGICAL (64-bit logical)	4
VECTOR POP/PARITY	5
FLOATING ADD	6
FLOATING MULTIPLY	7
RECIPROCAL APPROXIMATION	14

Cray X-MP: Basic Operation of the Vector Section [cont'd.]

- One cycle after the last pair of input elements has entered the pipeline, the functional unit can be used for another operation. Thus the functional unit will be available in
  $3 + VL + 1 = VL + 4$ clock cycles.
  Here VL is the value of the vector length register when the instruction issues.

- The source operands become available immediately after the last pair of input elements has entered the pipeline. Vector source registers are therefore reserved for $VL + 3$ clock cycles.

- The third phase of a pipelined operation is the shutdown phase. The shutdown time is the difference between the time when the last individual result emerges from the pipeline and the time when the destination vector register becomes available for another operation. The shutdown time is 3 clock cycles, so the destination register becomes available in
  $3 + n + (VL - 1) + 3 = n + VL + 5$ clock cycles
  where n is the number of stages in the pipeline and VL is the number of component operations to be performed. (Chaining is an exception to this rule.)

Cray X-MP: Vector Operations (contd.)

Example 1: Timing for a simple vector operation.

Line	Instruction		Description
1	A1	53	Set register A1 to 53
2	VL	A1	Set the length of the vector to 53
3	V4	V2+V3	Perform the addition

Setup time ---------- 3 cycles

Vector integer adder is a 3-stage pipeline

Time required to compute first result ------ 3 cycles

First result emerges after 3+3 = 6 cycles

Time required to compute remaining 52 results 52 cycles

Shutdown time ---------- 3 cycles

Destination register V4 will be available after 6+52+3 = ---------- 61 cycles

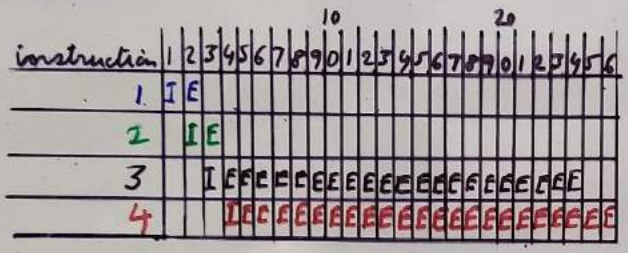
The functional unit, i.e, adder can be reused after VL+4 = 57 cycles

The vector source registers V2 and V3 are reserved for VL+3 = 56 cycles

Gray X-MP : Vector Operations [contd.]

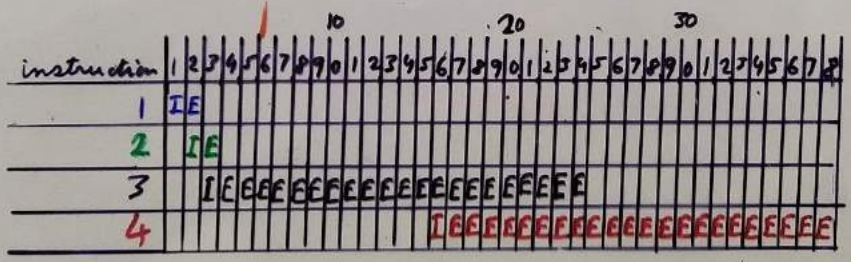
Example 2(a) : No source reservation conflict

Line	Instruction		Description
1	A1	10	Set A1 to 10
2	VL	A1	Set the vector length to 10
3	V4	V3+FV2	Set V4 to floating sum of V3 and V2
4	V6	V5*FV7	Set V6 to floating product of V5 and V7



Example 2(b) : Source reservation results in a conflict

Line	Instruction		Description
1	A1	10	Set A1 to 10
2	VL	A1	Set the vector length to 10
3	V4	V3+FV2	Set V4 to floating sum of V3 and V2
4	V6	V3*FV7	Set V6 to floating product of V3 and V7



V3 is reserved by instruction 3 for VL+3 cycles, i.e, 13 cycles [4..16]

Cray X-MP: Vector Chaining.

Line	Instruction		Description
1	A1	10	Set A1 to 10
2	VL	A1	Set the vector length to 10
3	V4	V3+FV2	Set V4 to floating sum of V3 and V2
4	V5	V4*FV7	Set V5 to floating product of V4 and V7

In chaining, the results produced by one operation can be used as input to a succeeding operation before the first instruction has completed. That is, the component results from the first instruction are used by the second instruction as they are produced.

With chaining, a value which emerges from the pipeline may be used by a waiting operation 2 cycles after it is produced. This is in contrast to unchained operations where the destination register cannot be accessed until 3 cycles after the last component computation is complete.