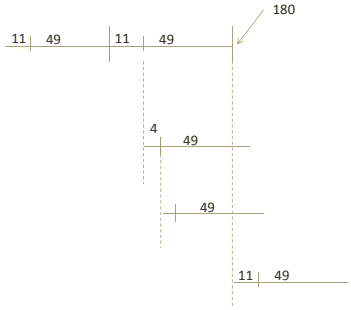
Vector Code – No Chaining

- The loop is vectorizable
- Initialize registers (including V_length and stride) 2+5 dynamic instruction

• LV	V1, R1	11+50-1
• LV	V2, R2	11+50-1
• ADDV	V3, V1, V2	4+50-1
• SLV	V3, V3, 1	1+50-1
• SV	V3, R4	11+50-1 =283+2

20

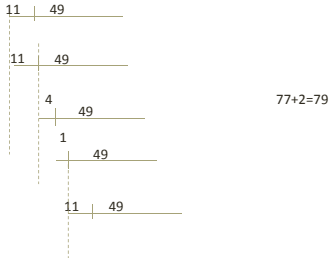
Vector Code – Chaining 1-port



21

Vector Code 2 load+1 store

- Chaining and 2 memory banks?



77+2=79

22

Chimes

- Convey:
  - *convoy,* which is the set of vector instructions that could potentially begin execution together in one clock period
  - Sequences with read-after-write dependency hazards can be in the same convey via *chaining*
- *Chaining*
  - Allows a vector operation to start as soon as the individual elements of its vector source operand become available
- *Chime*
  - Unit of time to execute one convey
  - *m* conveys executes in *m* chimes
  - For vector length of *n*, requires *m* x *n* clock cycles

23

Example

```
LV      V1,Rx      ;load vector X
MULVS.D V2,V1,F0   ;vector-scalar multiply
LV      V3,Ry      ;load vector Y
ADDVV.D V4,V2,V3   ;add two vectors
SV      Ry,V4      ;store the sum
```

Convoys:

```
LV      V1,Rx      ;load vector X
MULVS.D V2,V1,F0   ;vector-scalar multiply
LV      V3,Ry      ;load vector Y
ADDVV.D V4,V2,V3   ;add two vectors
SV      Ry,V4      ;store the sum
```

4 conveys => 4 x 64 » 256 clocks (or 4 clocks per result)

24

Vector length

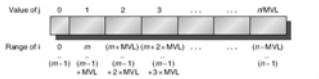
- In the previous example, the vector length is less than the VREG length.
- What if more (operation on a vector of 1000 elements)
- Loops each load perform on a 64 element vector (need to adjust vector length in the last iteration)

25

Vector Stripmining

- Vector length not known at compile time?
- Use Vector Length Register (VLR)
- Use strip mining for vectors over the maximum length:

```
low = 0;
VL = (n % MVL); /*find odd-size piece using modulo op % */
for (j = 0; j <= (n/MVL); j=j+1) { /*outer loop*/
    for (i = low; i < (low+VL); i=i+1) /*runs for length VL*/
      Y[i] = a * X[i] + Y[i] ; /*main operation*/
    low = low + VL; /*start of next vector*/
    VL = MVL; /*reset the length to maximum vector length*/
}
```



26

Effect of Memory

- Load/store unit is more complicated than FU's
- Start-up time, is the time for the first word into a register
- Memory system must be designed to support high bandwidth for vector loads and stores
- Spread accesses across multiple banks
  - Control bank addresses independently
  - Load or store non sequential words
  - Support multiple vector processors sharing the same memory
- Example:
  - 32 processors, each generating 4 loads and 2 stores/cycle
  - Processor cycle time is 2.167 ns, SRAM cycle time is 15 ns
  - How many memory banks needed?

27

Example

- Cray T932 has 32 processors. Each processor is capable of generating 4 loads and 2 stores per clock cycle.
- Clock cycle is 2.167 ns, SRAM cycle time 15 ns. How many bank do we need to allow the system to run at a full memory bandwidth?

28

Example

- 8 memory banks, bank busy time 6 cycles, total memory latency 12 cycles.
- What is the difference between a 64-element vector load with a stride of 1 and 32?

29

Stride

- Consider:

```
for (i = 0; i < 100; i=i+1)
        for (j = 0; j < 100; j=j+1) {
                A[i][j] = 0.0;
                for (k = 0; k < 100; k=k+1)
                A[i][j] = A[i][j] + B[i][k] * D[k][j];
        }
```

- Must vectorize multiplication of rows of B with columns of D
- Use *non-unit stride*
- Bank conflict (stall) occurs when the same bank is hit faster than bank busy time:
  - #banks / LCM(stride,#banks) < bank busy time

30

Strides

Add in a bank					SE	Q		M	O	D
	0	1	2	3	0	1	2	0	1	2
0	0	1	2	3	0	1	2	0	16	8
1	4	5	6	7	3	4	5	9	1	17
2	8	9	10	11	6	7	8	18	10	2
3	12	13	14	15	9	10	11	3	19	11
4	16	17	18	19	12	13	14	12	4	20
5	20	21	22	23	15	16	17	21	13	5
6	24	25	26	27	18	19	20	6	22	14
7	28	29	30	31	21	22	23	15	7	23

31

Strides

- MOD can be calculated very efficiently if the prime number is 1 less than a power of 2.
- Division still a problem
- But if we change the mapping such that
- Address in a bank = address MOD number of words in a bank.
- Since the number of words in a bank is usually a power of 2, that will lead to a very efficient implementation.
- Consider the following example, the first case is the usual 4 banks, then 3 banks with sequential interleaving and modulo interleaving and notice the conflict free access to rows and columns of a 4 by 4 matrix

32

Vector Mask Register

- What if we have a conditional IF statement inside the loop?
- Using scalar architecture, that introduces control dependence.
- The *vector-mask control*: A mask register is used to conditionally execute using a Boolean condition.
- When the *vector-mask register* is enabled, any vector instruction executed operate only on vector elements whose corresponding entries in the VMR are ones.
- The rest of the elements are unaffected.
- Clearing the vector mask register, sets to all 1's and operations are performed on all the elements.
- Does not save execution time for masked elements

33

Vector Mask Register

- Consider:

```
for (i = 0; i < 64; i=i+1)
    if (X[i] != 0)
        X[i] = X[i] – Y[i];
```

- Use vector mask register to "disable" elements:

```
LV        V1,Rx       ;load vector X into V1
LV        V2,Ry       ;load vector Y
L.D       F0,#0       ;load FP zero into F0
SNEVS.D   V1,F0       ;sets VM(i) to 1 if V1(i)!=F0
SUBVV.D   V1,V1,V2    ;subtract under vector mask
SV        Rx,V1       ;store the result in X
```

34

Scatter-Gather

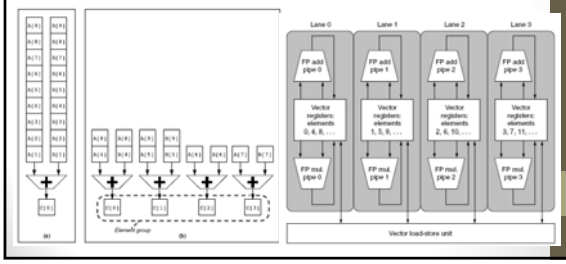
- Consider:
  for (i = 0; i < n; i=i+1)
  A[K[i]] = A[K[i]] + C[M[i]];
- Use index vector:

```
LV      Vk, Rk          ;load K
LVI     Va, (Ra+Vk)     ;load A[K[]]
LV      Vm, Rm          ;load M
LVI     Vc, (Rc+Vm)     ;load C[M[]]
ADDVV.D Va, Va, Vc      ;add them
SVI     (Ra+Vk), Va     ;store A[K[]]
```

35

Multiple lanes

- Operations are interleaved across multiple lanes.
  - Allows for multiple hardware lanes and no changes to machine code



36

Not Quite SIMD

- Intel extension MMx, SSE, AVX, PowerPC AltiVec, ARM Advanced SIMD
- No vector length, just depends on the instruction, the register can be considered 16 8-bit numbers, 8 16-bit numbers, …

37