VECTOR CODE EXAMPLE

// ASSUME n = 4
for (i = 0; i < n; i++)
D[i] = A[i] * C[i]

SCALAR ASSEMBLY

LOOP:
LW r1, 0(r2)
MUL r3, r1, r4
SW r3, 0(r5)
ADDW r2, r2, 4
ADDW r5, r5, 4
ADDW r4, r4, -1
LJ+2 r7, loop

VECTOR ASSEMBLY

LW.V vrl, r2
MUL.VV vr2, vrl, vr4
SW.V vr3, r5

ASSUME n Vel = 4

VECTOR MICROARCHITECTURE

ASSUME 32 VECTOR REGISTERS; VECTOR REGISTER FILE HAS 32 x VIEW ELEMENTS
BASIC EXECUTION, VLEN = 4

```
LW.V  F D R [U][W]  
      R[0] U W  
      R[1] U W  
      R[2] U W  
      via vector register file

MUL.VV  F D D D D D D D D  
        [U][W]  
        via vreg file

SW.V  F F F F F F F F F F D D D D D D D D  
       [U][W]  
       R[0] to 31 W  
       R[0] to 31 W  
       R[0] to 31 W  
       R[0] to 31 W

Stall on raw vector previous
vector instructions not written with
result! * DEP checking once per vector word!
```

CHAINING, VLEN = 4

```
LW.V  F D R [U][W]  
      R[0] U W  
      R[1] U W  
      R[2] U W  
      via vreg file, analogous to scalar bypassing

MUL.VV  F D D D D D D D D  
        [U][W]  
        via vreg file

SW.V  F F F F F F F F F F D D D D D D D D  
       [U][W]  
       R[0] to 31 W  
       R[0] to 31 W  
       R[0] to 31 W  
       R[0] to 31 W

Stall due to R[0] to 31 W
with port conflict
```

CHAINING, VLEN = 2 (Assume n Also = 2)

```
LW.V  F D R [U][W]  
      D R[0] U W  
      via bypass network, stall in D

MUL.VV  F D D D D D D D  
        [U][W]  

SW.VV  F D D D D D D D D D  
        [U][W]  
        R[0] to 31 W  
        R[0] to 31 W

CANNOT HIDE ALU LATECY U DLP
CHAINING, VIEW = 8 (ASSUME n = 8)

DATA FOR FIRST ELEMENT IS READY EARLY!

Plenty of issue time!

STRUCTURAL HAZARD

Key Observation: With relatively little fetch issue bandwidth and limited fetch issue, we can still keep functional units busy for many cycles.

ADD MORE VREG FILE PORTS TO EXPLOIT VECTOR INSTA. LEVEL PARALLELISM

TO KEEP ALL FUNCTIONAL UNITS BUSY AT SAME TIME, NEED:

<table>
<thead>
<tr>
<th>TYPE</th>
<th>latency</th>
<th>bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-Pipe</td>
<td>2c</td>
<td>1w</td>
</tr>
<tr>
<td>X-Pipe</td>
<td>2c</td>
<td>1w</td>
</tr>
<tr>
<td>L-Pipe</td>
<td>1c</td>
<td>1w</td>
</tr>
<tr>
<td>S-Pipe</td>
<td>2c</td>
<td>0w</td>
</tr>
</tbody>
</table>

TOTAL: 3w
CUAITION, VLEN = 8 (ASSUME n = 8) of FROM VNEFEE

3 VECTOR UNIT EXEC AT SAME TIME, EQUIVALENT TO 32 SCALAR OPERATIONS IN FLIGHT AT ONCE IN SINGLE ISSUE!

USING 5-2W PARTS

WHAT IF VLEN != N?

NEW "INT VECTOR LENGTH REGISTER" INSTRUCTION

```
setvl ra, rb R[ra] ← min (vlen, R[rb]);
```

IF VLEN = 4, BUT N = 2

```l
li r8
setvl r9, r8
lw v r1, r2
mul.v v r3, v r1, v r4
sw v v r3, r5
```

EVEN THOUGH HARDWARE VECTOR LENGTH IS 4, WILL ONLY PROCESS TWO ELEMENTS PER VECTOR — USE HALF THE VNEFEE.

SAME PATTERN AS IF VLEN = 2
If \( \text{vlen} = 4 \), but \( n \) is unknown and \( n \) could be \( > \text{vlen} \)

**Scalar Assembly**

- **Loop:**
  - \( \text{lw} r1, 0(r2) \)
  - \( \text{mul} r3, r1, r4 \)
  - \( \text{sw} r5, 0(r\text{s}) \)
  - \( \text{addw} r6, r2, r4 \)
  - \( \text{addw} r7, r5, r4 \)
  - \( \text{byle} r7, r7, -1 \)
  - \( \text{byle} r7, r1, \text{loop} \)

**Vector Assembly**

- **Loop:**
  - \( \text{setvl} 8, r7 \)
  - \( \text{lw} v1, vr1, r2 \)
  - \( \text{mul} v3, vr1, v4 \)
  - \( \text{sw} v5, vr1, r5 \)
  - \( \text{sil} r9, r8, 2 \)
  - \( \text{addw} r2, r2, r9 \)
  - \( \text{addw} r3, r3, r9 \)
  - \( \text{subv} r7, r7, r8 \)
  - \( \text{byle} r7, r1, \text{loop} \)

**Strip Mining**

- **Vlen = 4**
- **N = 26 Elements**

For reference, what is execution time of Scalar code? \( n = 64 \)

- \( \text{iw} \)
- \( \text{mul} \)
- \( \text{sw} \)
- \( \text{addw} \)
- \( \text{byle} \)
- \( \text{opA} \)
- \( \text{opB} \)
- \( \text{opC} \)

\[
\text{# instr} = 7 \times 64 = 448
\]

\[
\text{cycles} \text{ lut} = \frac{(14 \times 64)}{448} = 2
\]

\[
\text{cycle time} = 1\text{ns}
\]

\[
\text{Total execution time} = 896\text{ns}
\]
\[
\text{# instr} = 9 \times 16 = 144 \\
\text{cycles/instr} = (16 \times 16) / 144 = 1.78 \\
\text{cycle time} = 1ns \\
\text{total execution time} = 256 \text{ cycles} \\
\text{speed up of 3.5 but needed to add 4x repeats} \\
\text{and 7+3w points}
\]

\[\text{What if } v_{\text{vec}} = 8?\]
MULTIPLE Lanes

Lane 0

Lane 1

Commit Pour

V Register is Partitioned by Lane.
Assume Lanes = 8 or 2 Lanes

Lane 0

Lane 1

Only needs to in Lane
16. Chewin', same as if I were

Scalar code: "gettym" in the way

- because scalar procession
Multiple lanes of decoupled control proc

Control processor is decoupled from vector unit

Word vector command queue