Data-Level Parallelism

- Regular parallelism occurring in matrix-vector computations
- Ex: Daxpy $Y[i] = a \cdot X[i] + Y[i]$

Scalar Code

```
      L.D    F0, a
Loop: L.D    F2, 0(Rx)
      MUL.D  F2, F2, F0
      L.D    F4, 0(Ry)
      ADD.D  F4, F4, F2
      S.D    F4, 0(Ry)
      DADDIU Rx, Rx, #8
      DADDIU Ry, Ry, #8
      DSUBU  R20, R4, Rx
      BNEZ   R20, Loop
```

⇒

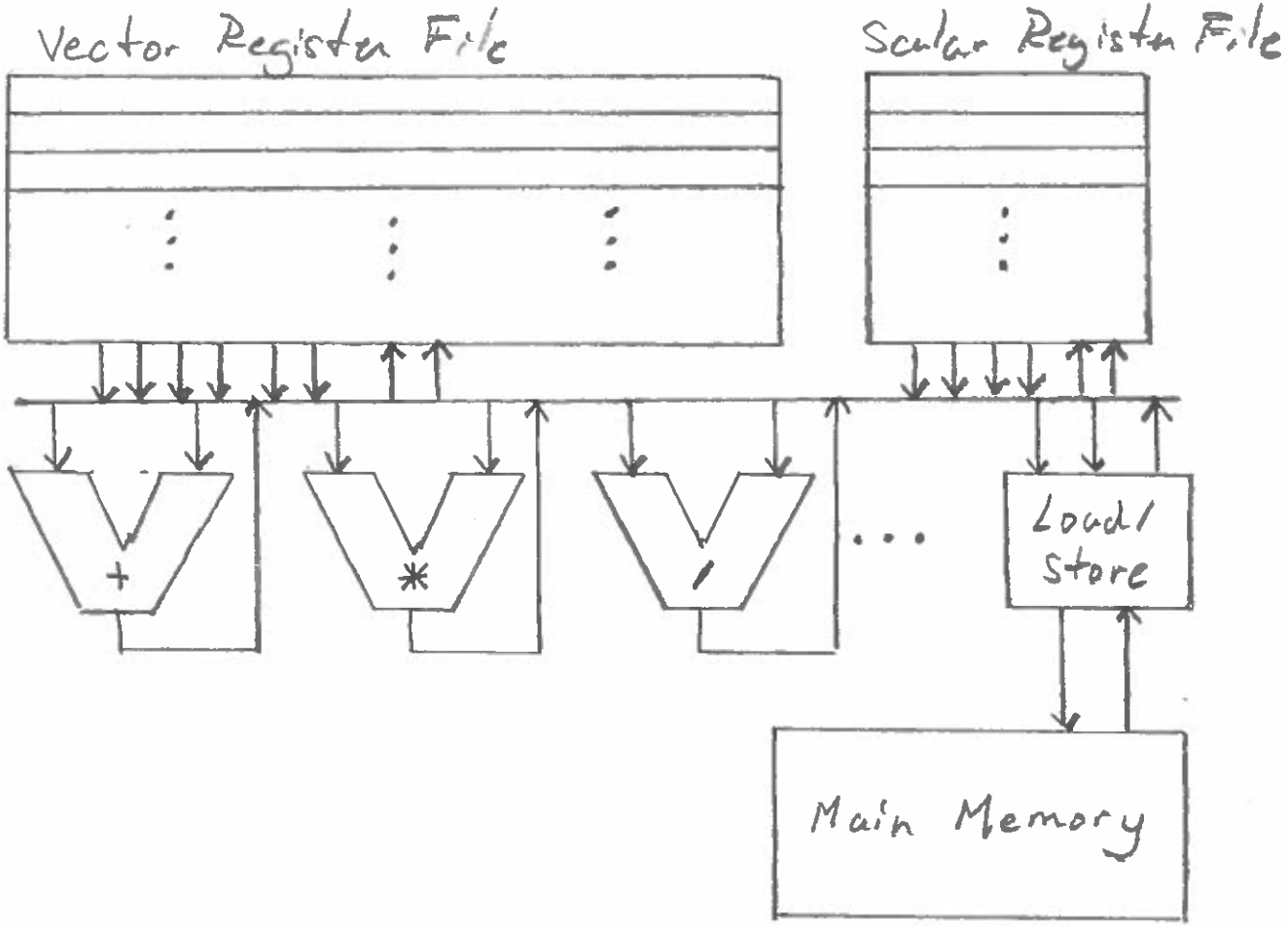
```
l l l l
* * * *
l l l l ...
+ + + +
S S S S
```

⇒

Vector Code

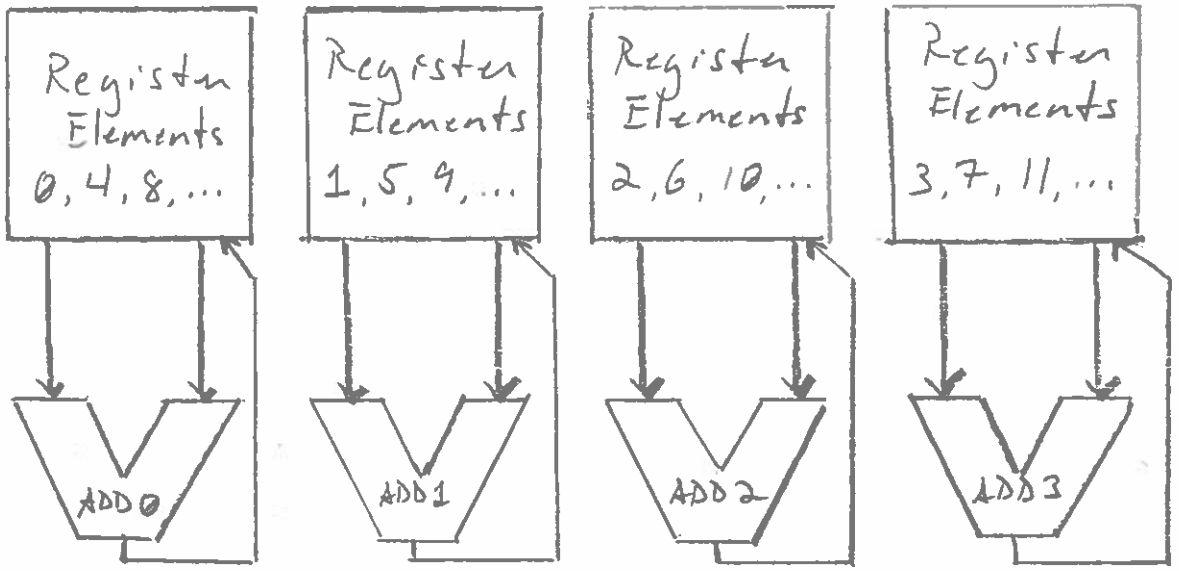
```
L.D    F0, a
LV     V1, Rx
MULVS.D V2, V1, F0
LV     V3, Ry
ADDVV.D V4, V2, V3
SV     V4, Ry
```

Vector Datapath



Multiple Lanes

Use multiple functional units to exploit parallelism within a single vector spatially.



Loop Unrolling for SIMD Parallelism

```
for (i=0; i<n; i++)
  d[i] = a[i] * b[i] + c;
```

```
LOOP: L.D F0, (R1)
      L.D F2, (R2)
      DADDI R1, #8
      DADDI R2, #8
      MUL.D F4, F0, F2
      ADD.D F6, F4, F16
      S.D F6, (R3)
      DADDI R3, R3, #8
      DADDI R4, R4, #1
      SLT R5, R4, R6
      BNEZ R5, LOOP
```

⇒

```
LOOP: L.D F0, (R1)
      L.D F1, 8(R1)
      L.D F2, 16(R1)
      L.D F3, 24(R1)
      L.D F4, (R2)
      L.D F5, 8(R2)
      L.D F6, 16(R2)
      L.D F7, 24(R2)
      MUL.D F8, F0, F4
      MUL.D F9, F1, F5
      MUL.D F10, F2, F6
      MUL.D F11, F3, F7
      ADD.D F12, F8, F16
      ADD.D F13, F9, F16
      ADD.D F14, F10, F16
      ADD.D F15, F11, F16
      S.D F12, (R3)
      S.D F13, 8(R3)
      S.D F14, 16(R3)
      S.D F15, 24(R3)
      DADDI R1, #32
      DADDI R2, #32
      DADDI R3, #32
      DADDI R4, #4
      SLT R5, R4, R6
      BNEZ R5, LOOP
```

⇒

```
LOOP: L.4D F0, (R1)
      L.4D F4, (R2)
      MUL.4D F8, F0, F4
      ADD.4D F12, F8, F16
      S.4D F12, (R3)
      DADDI R1, #32
      DADDI R2, #32
      DADDI R3, #32
      DADDI R4, #4
      SLT R5, R4, R6
      BNEZ R5, LOOP
```