Vector Multiply High and Add Signed Half Word Saturate

vmhaddshs vD,vA,vB,vC

04	vD	vA	vB	vC	32
0 5	6 10	11 15	16 20	21 25	26 31

```
do i=0 to 127 by 16
  prod0-31← (vA)i:i+15 Xsi (vB)i:i+15
  temp0-16← prod0:16 +int SignExtend((vC)i:i+15,17)
  vDi:i+15← SItoSIsat(temp0-16,16)
```

Each signed-integer half word element in vA is multiplied by the corresponding signed-integer half word element in vB, producing a 32-bit signed-integer product. Bits 0-16 of the product are added to the corresponding signed-integer half-word element in vC.

If the intermediate result is greater than $2^{15}-1$ it saturates to $2^{15}-1$ and if it is less than -2^{15} it saturates to -2^{15}.

The signed-integer result is placed into the corresponding half-word element of vD.

Other registers altered:

- Vector status and control register (VSCR):

  Affected: SAT

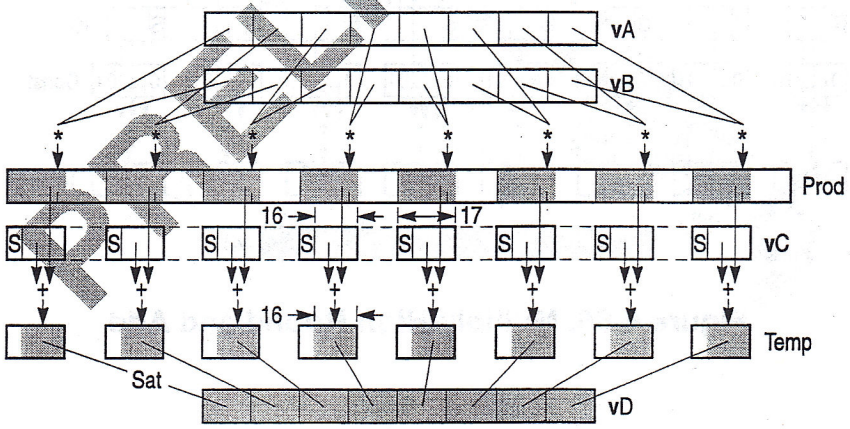


Figure 6-55. Multiply-High and Add

Vector Multiply High and Add Signed Half Word Saturate

vmhraddshs vD,vA,vB,vC

04	vD	vA	vB	vC	33
0 5	6 10	11 15	16 20	21 25	26 31

```
do i=0,127,16
 prod0-31 ← (vA)i:i+15 *si (vB)i:i+15
 temp0-16 ← (prod0-17 +int (SignExtend((vC)i:i+15,17) || 0b1))0-16
 (vD)i:i+15 ← SItoSIsat(temp0-16,16)
 (VSCRsat) ← (VSCRsat) ∨ ((vD)i:15 saturated)
end
```

The eight 16-bit signed integers in vA are multiplied by the eight 16-bit signed integers in vB. The intermediate product is rounded to 17 bits of significance and each intermediate rounded product added to the 16-bit signed integers in vC after they have been sign extended to 17-bits. The 16-bit saturated result from each of the eight 17-bit sums is placed in vD.

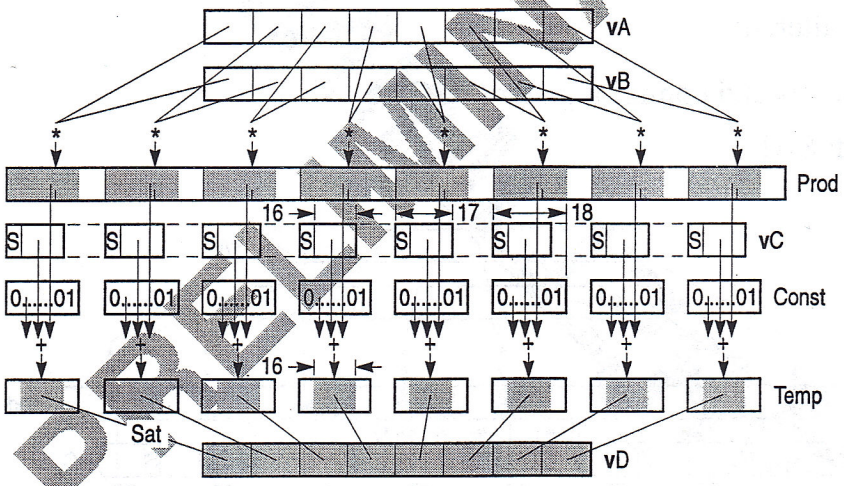
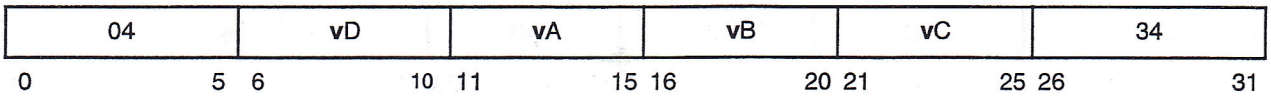


Figure 6-56. Multiply-High Round and Add

Vector Multiply Low and Add Unsigned Half Word Modulo

vmladduhm vD,vA,vB,vC




```
do i=0 to 127 by 16
  prod0-31← (vA)i:i+15 Xui (vB)i:i+15
  vDi:i+15← prod0:31 +int (vC)i:i+15
```

Each unsigned-integer half-word element in vA is multiplied by the corresponding unsigned-integer half-word element in vB, producing a 32-bit unsigned-integer product. The product is added to the corresponding unsigned-integer half-word element in vC. The unsigned-integer result is placed into the corresponding half-word element of vD.

Note that **vmladduhm** can be used for unsigned or signed integers.

Other registers altered:

- None

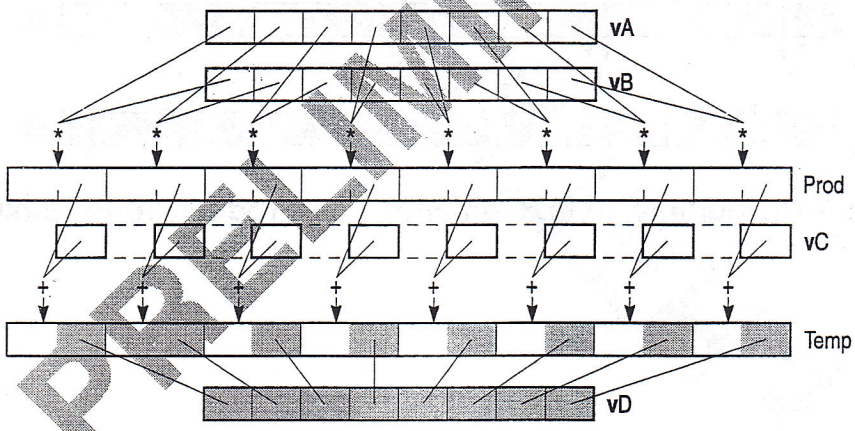
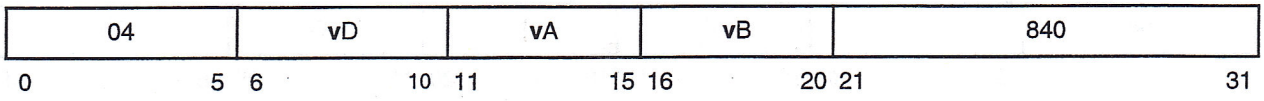


Figure 6-64. Multiply-Low and Add

vmulesh

Vector Multiply Even | Odd | Signed | Half Word

vmulesh vD,vA,vB




```
n ← LENGTH(element)
do i=0 to 127 by n*2
  prod0:n*2-1← (vA)i:i+n-1 ×si (vB)i:i+n-1
  vDi:i+n*2-1← prod0:n*2-1
```

Each even-numbered signed-integer half-word element in vA is multiplied by the corresponding signed-integer half-word element in vB. The four 32-bit signed-integer products are placed, in the same order, into the four words of vD.

Other registers altered:

- None

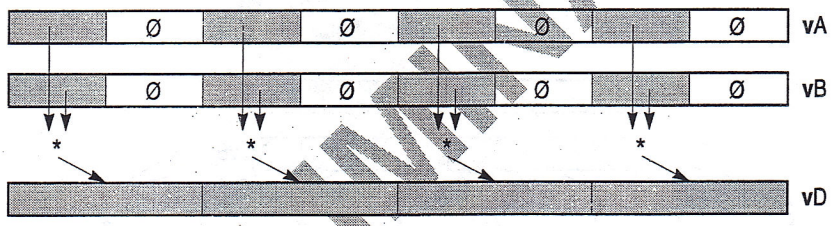


Figure 6-78. Multiply, Even Elements, Full-Product, 16-Bit Elements

(1) BYTE is only modulo
(2) BYTE add 4 to 1 at the end

vmsumshs

Vector Multiply Sum Signed (UNSIGNED) Half Word Saturate (BYTE, MODULO)

vmsumshs

vmsumshs vD,vA,vB,vC

04	vD	vA	vB	vC	41
0 5	6 10	11 15	16 20	21 25	26 31

```
do i=0 to 127 by 32
  temp0-33 ← SignExtend((vC)i:i+31,34)
  do j=0 to 31 by 16
    prod0-31 ← (vA)i+j:i+j+15 ×si (vB)i+j:i+j+15
    temp0-33 ← temp0-33 +int SignExtend(prod0-31,34)
  vDi:i+31 ← SItoSIsat(temp0-33,32)
```

For each word element in vC the following operations are performed in the order shown.

- Each of the two signed-integer half-word elements in the corresponding word element of vA is multiplied by the corresponding signed-integer half-word element in vB, producing a signed-integer product.
- The signed-integer sum of these two products is added to the signed-integer word element in vC.
- If the intermediate result is greater than $2^{31}-1$ it saturates to $2^{31}-1$ and if it is less than -2^{31} it saturates to -2^{31}.
- The signed-integer result is placed into the corresponding word element of vD.

Other registers altered:

- SAT

This is a MADD22

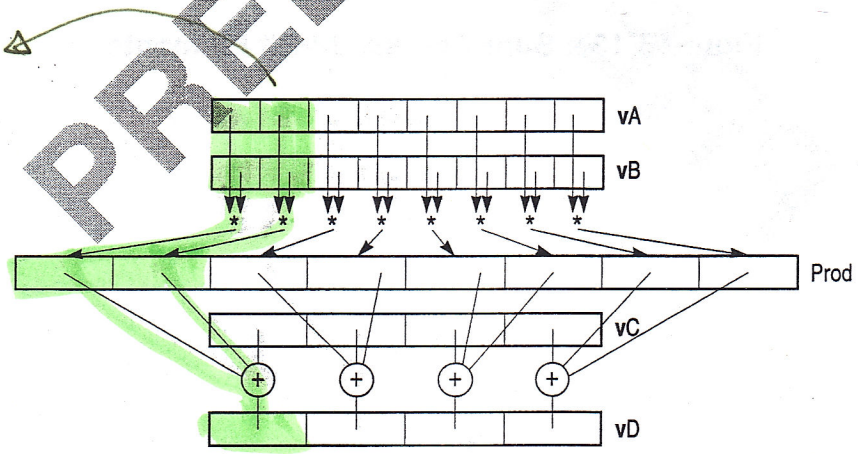
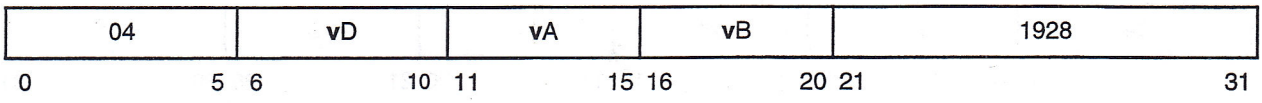


Figure 6-73. Multiply-Sum—16-Bit Elements

Vector Sum Across Signed Word Saturate

vsumsws vD,vA,vB




```
temp0-34 ← SignExtend((vB)96-127,35)
do i=0 to 127 by 32
  temp0-34 ← temp0-34 +int SignExtend((vA)i:i+31,35)
vD ← 960 || SItoSIsat(temp0-34,32)
```

The signed-integer sum of the four signed-integer word elements in vA is added to the signed-integer word element in bits of vB[96-127]. If the intermediate result is greater than $2^{31}-1$ it saturates to $2^{31}-1$ and if it is less than -2^{31} it saturates to -2^{31}. The signed-integer result is placed into bits of vD[96–127]. Bits of vD[0–95] are cleared.

Other registers altered:

- SAT

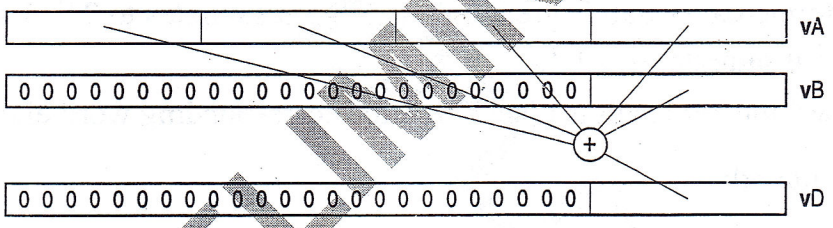


Figure 6-137. Sum-Across, 32-Bit Elements

Vector Sum Across Partial (1/2) Signed Word Saturate

vsum2sws vD,vA,vB

04	vD	vA	vB	1672
0 5	6 10	11 15	16 20	21 31

```
do i=0 to 127 by 64
  temp0:33 ← SignExtend((vB)i+32:i+63,34)
  do j=0 to 63 by 32
    temp0:33 ← temp0:33 +int SignExtend((vA)i+j:i+j+31,34)
vDi:i+63 ← 320 || SItoSIsat(temp0:33,32)
```

The signed-integer sum of the first two signed-integer word elements in register vA is added to the signed-integer word element in vB[32–63]. If the intermediate result is greater than $2^{31}-1$ it saturates to $2^{31}-1$ and if it is less than -2^{31} it saturates to -2^{31}. The signed-integer result is placed into vD[32–63]. The signed-integer sum of the last two signed-integer word elements in register vA is added to the signed-integer word element in vB[96-127]. If the intermediate result is greater than $2^{31}-1$ it saturates to $2^{31}-1$ and if it is less than -2^{31} it saturates to -2^{31}. The signed-integer result is placed into vD[96–127]. The register vD[0–31,64–95] are cleared to 0.

Other registers altered:

- SAT

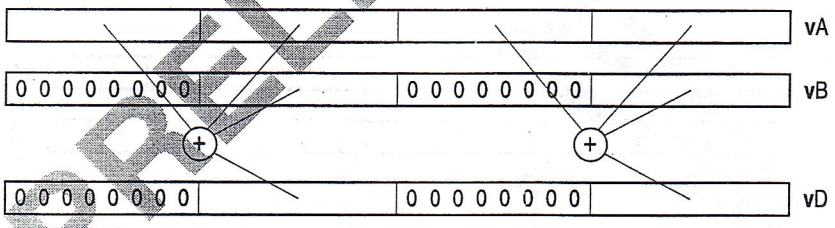


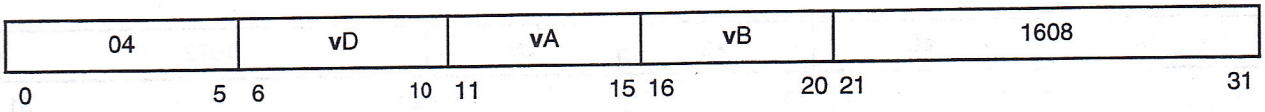
Figure 6-138. Partial (1/2) Sum Across, 32-Bit Elements

vsum4shs

Vector Sum Across Partial (1/4) Signed Half Word Saturate

vsum4shs

vsum4shs vD,vA,vB




```
do i=0 to 127 by 32
  temp0:32 ← SignExtend((vB)i:i+31,33)
  do j=0 to 31 by 16
    temp0:32 ← temp0:32 +int SignExtend((vA)i+j:i+j+15,33)
  vDi:i+31 ← SItoSIsat(temp0:32,32)
```

For each word element in register vB the following operations are performed, in the order shown.

- The signed-integer sum of the two signed-integer halfword elements contained in the corresponding word element of register vA is added to the signed-integer word element in vB.
- If the intermediate result is greater than $2^{31}-1$ it saturates to $2^{31}-1$ and if it is less than -2^{31} it saturates to -2^{31}.
- The signed-integer result is placed into the corresponding word element of vD.

Other registers altered:

- SAT

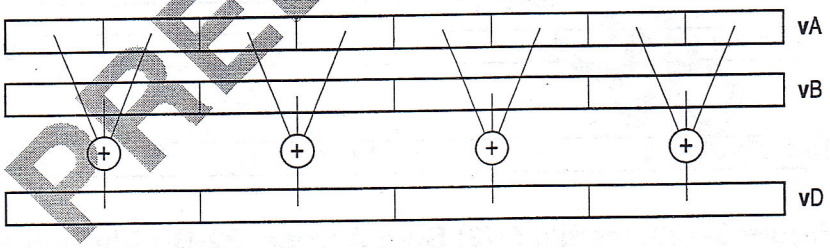


Figure 6-140. Partial (1/4) Sum-Across, 8-Bit Elements