

Image Down-Sample

- 1) $I \leftarrow$ 1D array of the image
- 2) $H \leftarrow$ Height of the image
- 3) $W \leftarrow$ Width of the image
- 4) $D \leftarrow$ Empty image array
- 5) $K \leftarrow$ Factor
- 6) $\text{Count} \leftarrow 1$
- 7) For $X = K : K : H$
 - a. $L \leftarrow (X-1) \times W$
 - b. For $j = K : K : W$
 - i. $\text{center_pixel} = L + j$
 - ii. $\text{Total} \leftarrow 0$
 - iii. $\text{Total} += I(\text{center pixel}) \times 1/4$
 - iv. $\text{Total} += I(\text{center pixel} + 1) \times 1/8$
 - v. $\text{Total} += I(\text{center pixel} - 1) \times 1/8$
 - vi. $\text{Total} += I(\text{center pixel} - 1 - W) \times 1/16$
 - vii. $\text{Total} += I(\text{center pixel} + 1 - W) \times 1/16$
 - viii. $\text{Total} += I(\text{center pixel} - W) \times 1/8$
 - ix. $\text{Total} += I(\text{center pixel} - 1 + W) \times 1/16$
 - x. $\text{Total} += I(\text{center pixel} + 1 + W) \times 1/16$
 - xi. $\text{Total} += I(\text{center pixel} + W) \times 1/8$
 - xii. $D(\text{count}) = \text{Total}$
 - xiii. $\text{count} += 1$
 - c. end
- 8) end
- 9) $\text{Downsampled Image} \leftarrow \text{Reshaped } D$

```
CONST H #VALUE          // H ← #VALUE
CONST W #VALUE          // W ← #VALUE
CONST K #VALUE          // K ← #VALUE
CONST COUNT #VALUE      // COUNT ← #VALUE = H * W
```

```
CONST X #VALUE          // X ← #VALUE
CONST J #VALUE          // J ← #VALUE
```

```
MV AC X                 // AC ← X                               ← L1
DEAC                    // AC ← AC - 1
MUL W                   // AC ← AC * W
MVL AC                  // L ← AC
```

```
MV AC J                 // AC ← J                               ← L2
ADD L                   // AC ← AC + L
MV CENTERP AC          // CENTERP ← AC
```

CLAC	// $AC \leftarrow 0$
CONST T 0	// $T \leftarrow 0$
MV MAR CENTERP	// $MAR \leftarrow CENTERP$
LDAC	// $AC \leftarrow DRAM[MAR]$
MUL4	// $AC \leftarrow AC \ll 2$
ADD T	// $AC \leftarrow AC + T$
MV T AC	// $T \leftarrow AC$

MV AC CENTERP	// $AC \leftarrow CENTERP$
INAC	// $AC \leftarrow AC + 1$
MV MAR AC	// $MAR \leftarrow AC$
LDAC	// $AC \leftarrow DRAM[MAR]$
MUL2	// $AC \leftarrow AC \ll 1$
ADD T	// $AC \leftarrow AC + T$
MV T AC	// $T \leftarrow AC$

MV AC CENTERP	// $AC \leftarrow CENTERP$
DEAC	// $AC \leftarrow AC - 1$
MV MAR AC	// $MAR \leftarrow AC$
LDAC	// $AC \leftarrow DRAM[MAR]$
MUL2	// $AC \leftarrow AC \ll 1$
ADD T	// $AC \leftarrow AC + T$
MV T AC	// $T \leftarrow AC$

MV AC CENTERP	// $AC \leftarrow CENTERP$
ADD W	// $AC \leftarrow AC + W$
MV MAR AC	// $MAR \leftarrow AC$
LDAC	// $AC \leftarrow DRAM[MAR]$
MUL2	// $AC \leftarrow AC \ll 1$
ADD T	// $AC \leftarrow AC + T$
MV T AC	// $T \leftarrow AC$

MV AC CENTERP	// $AC \leftarrow CENTERP$
INAC	// $AC \leftarrow AC + 1$
ADD W	// $AC \leftarrow AC + W$
MV MAR AC	// $MAR \leftarrow AC$
LDAC	// $AC \leftarrow DRAM[MAR]$
ADD T	// $AC \leftarrow AC + T$
MV T AC	// $T \leftarrow AC$

MV AC CENTERP	// $AC \leftarrow CENTERP$
DEAC	// $AC \leftarrow AC - 1$
ADD W	// $AC \leftarrow AC + W$
MV MAR AC	// $MAR \leftarrow AC$
LDAC	// $AC \leftarrow DRAM[MAR]$
ADD T	// $AC \leftarrow AC + T$
MV T AC	// $T \leftarrow AC$

<i>MV AC CENTERP</i>	<i>// AC ← CENTERP</i>
<i>SUB W</i>	<i>// AC ← AC - W</i>
<i>MV MAR AC</i>	<i>// MAR ← AC</i>
<i>LDAC</i>	<i>// AC ← DRAM[MAR]</i>
<i>MUL2</i>	<i>// AC ← AC << 1</i>
<i>ADD T</i>	<i>// AC ← AC + T</i>
<i>MV T AC</i>	<i>// T ← AC</i>

<i>MV AC CENTERP</i>	<i>// AC ← CENTERP</i>
<i>INAC</i>	<i>// AC ← AC + 1</i>
<i>SUB W</i>	<i>// AC ← AC - W</i>
<i>MV MAR AC</i>	<i>// MAR ← AC</i>
<i>LDAC</i>	<i>// AC ← DRAM[MAR]</i>
<i>ADD T</i>	<i>// AC ← AC + T</i>
<i>MV T AC</i>	<i>// T ← AC</i>

<i>MV AC CENTERP</i>	<i>// AC ← CENTERP</i>
<i>DEAC</i>	<i>// AC ← AC - 1</i>
<i>SUB W</i>	<i>// AC ← AC - W</i>
<i>MV MAR AC</i>	<i>// MAR ← AC</i>
<i>LDAC</i>	<i>// AC ← DRAM[MAR]</i>
<i>ADD T</i>	<i>// AC ← AC + T</i>

<i>DIV16</i>	<i>// AC ← AC >> 4</i>
<i>MV T AC</i>	<i>// T ← AC</i>

<i>MV MAR COUNT</i>	<i>// MAR ← COUNT</i>
<i>STAC</i>	<i>// DRAM[MAR] ← AC</i>

<i>MV AC COUNT</i>	<i>// AC ← COUNT</i>
<i>INAC</i>	<i>// AC ← AC + 1</i>
<i>MV COUNT AC</i>	<i>// COUNT ← AC</i>

<i>MV AC J</i>	<i>// AC ← J</i>
<i>ADD K</i>	<i>// AC ← AC + K</i>
<i>MV J AC</i>	<i>// J ← AC</i>
<i>MV AC W</i>	<i>// AC ← W</i>

<i>JUMPN J L2</i>	<i>// IF W < J, GO TO L2</i>
-------------------	---------------------------------

<i>CONST J #VALUE</i>	<i>// J ← #VALUE</i>
<i>MV AC X</i>	<i>// AC ← X</i>
<i>ADD K</i>	<i>// AC ← AC + K</i>
<i>MV X AC</i>	<i>// X ← AC</i>
<i>MV AC H</i>	<i>// AC ← H</i>

JUMPN X LI // IF $H < X$, GO TO LI

NOP