

What follows is a reworking of Example 11.5 to illustrate the use of BSR's memory access unit (MAU) shown in Fig. 11.14 and consisting of a SORT circuit and a PREFIX circuit. The SORT circuit may be, for example, the circuit of Fig 3.5 (rotated 90° clockwise). The PREFIX circuit is the circuit of Fig. 11.9 (also rotated 90° clockwise).

**Example 11.5 (revisited)**

Assume for the purpose of this example that  $N = M = 4$ , and let the **BROADCAST** instruction be

$$v_j \leftarrow \sum_{\substack{1 \leq i \leq N \\ g_i < l_j}} d_i.$$

Suppose that the four processor records  $(i, g_i, d_i)$  are

$$(1, 15, 9), \quad (2, -4, -5), \quad (3, 17, -2), \quad (4, 11, 10),$$

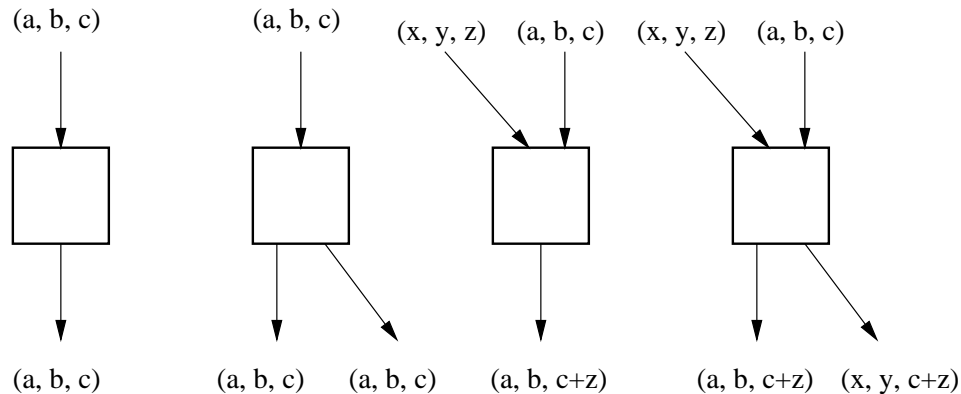
while the four memory records  $(j, l_j, v_j)$ , are

$$(1, 16, v_1), \quad (2, 12, v_2), \quad (3, 18, v_3), \quad (4, -6, v_4).$$

Initially,  $v_j = 0$ , for  $1 \leq j \leq 4$ . When the **BROADCAST** instruction is complete, we want  $v_1 = 9 - 5 + 10 = 14$  (since 15, -4, and 11 are less than 16),  $v_2 = -5 + 10 = 5$  (since -4 and 11 are less than 12),  $v_3 = 9 - 5 - 2 + 10 = 12$  (since all *tags* are less than 18), and  $v_4 = 0$  (since no *tag* is less than -6).

The processor and memory records are fed as input to the SORT circuit. They exit from it sorted (from left to right) on their second fields (i.e., the  $g_i$  and the  $l_j$ ) as  $\{(4, -6, v_4), (2, -4, -5), (4, 11, 10), (2, 12, v_2), (1, 15, 9), (1, 16, v_1), (3, 17, -2), (3, 18, v_3)\}$ . Note that in case of equality, a memory record precedes a processor record (since the selection rule is ' $<$ ' and we don't want any datum  $d_i$  to be included in the computation of  $v_j$  unless its tag  $g_i$  is strictly smaller than  $l_j$ ).

The sorted sequence of records now enters a PREFIX circuit whose width is  $N + M = 8$  and whose depth is  $1 + \log(N + M) = 4$ . This circuit computes the values of the  $v_j$ . It performs *selection* and *reduction* by computing the sum of the  $d_i$  entries of all processor records preceding  $v_j$ 's record in the sorted sequence. This computation uses only the links going 'down' and 'to the right' from one stage to the next in the PREFIX circuit (again because the selection rule is ' $<$ '). Depending on whether it has one or two inputs and one or two outputs, a component of the PREFIX circuit behaves in one of the following ways:



After the last stage of the PREFIX circuit has performed its computation, the records have become  $\{(4, -6, 0), (2, -4, -5), (4, 11, 5), (2, 12, 5), (1, 15, 14), (1, 16, 14), (3, 17, 12), (3, 18, 12)\}$ . Note in particular that  $v_4 = 0$ ,  $v_1 = 14$ ,  $v_2 = 5$ , and  $v_3 = 12$ , as required. The four memory records  $(4, -6, 0)$ ,  $(2, 12, 5)$ ,  $(1, 16, 14)$ , and  $(3, 18, 12)$  now retrace their paths through the MAU to return to their respective memory locations.  $\square$