Explicatie pentru codul din Factor:

a = rez;

a = a>>(11-i); i-iteratia

Text

Description automatically generated

Adaugarea bitilor in rezultatul final se face de la MSB spre LSB, iar valoarea lui a este mereu rezultatul shiftat la dreapta cu (numarul maxim de biti al rezultatului – valoarea iteratiei).

Presupunem ca avem D = 1000 (numar pe 10 biti) si pentru a-l putea folosi in operatiile radicalului, il punem pe 12 biti => rezultatul va fi pe 12/3 = 4 biti.

1 0 1 0 = 10 = rez

= 1000

La iteratia 1 => f = 1 <= 001(primii 3 biti care ies din MUX1) 001-

* rez[4-1] = 1 (din f<=d) -------

000 111-

La ieratia 2 => a = 1000>>(4-2) = 0010 = 2 000 000

* f = 3\*2\*2+3\*2+1 = 19 > 7 => rez[4-2] = 0 ----------

000 111 101- = 61

La iteratia 3 => a = 1000>>(4-3) = 0100 = 4 000 111 101

* f = 3\*4\*4+3\*3+1 = 61 <= 61 => rez[4-3] = 1 ------------------

000 000 000 000-

La iteratia 4 => a = 1010>>(4-4) = 1010 = 10 000 000 000 000

* f = 3\*10\*10+3\*10+1 = 331 > 0 => rez[4-4] = 0 -----------------------

000 000 000 000 = rest