TP FIR

en assembleur SHARC

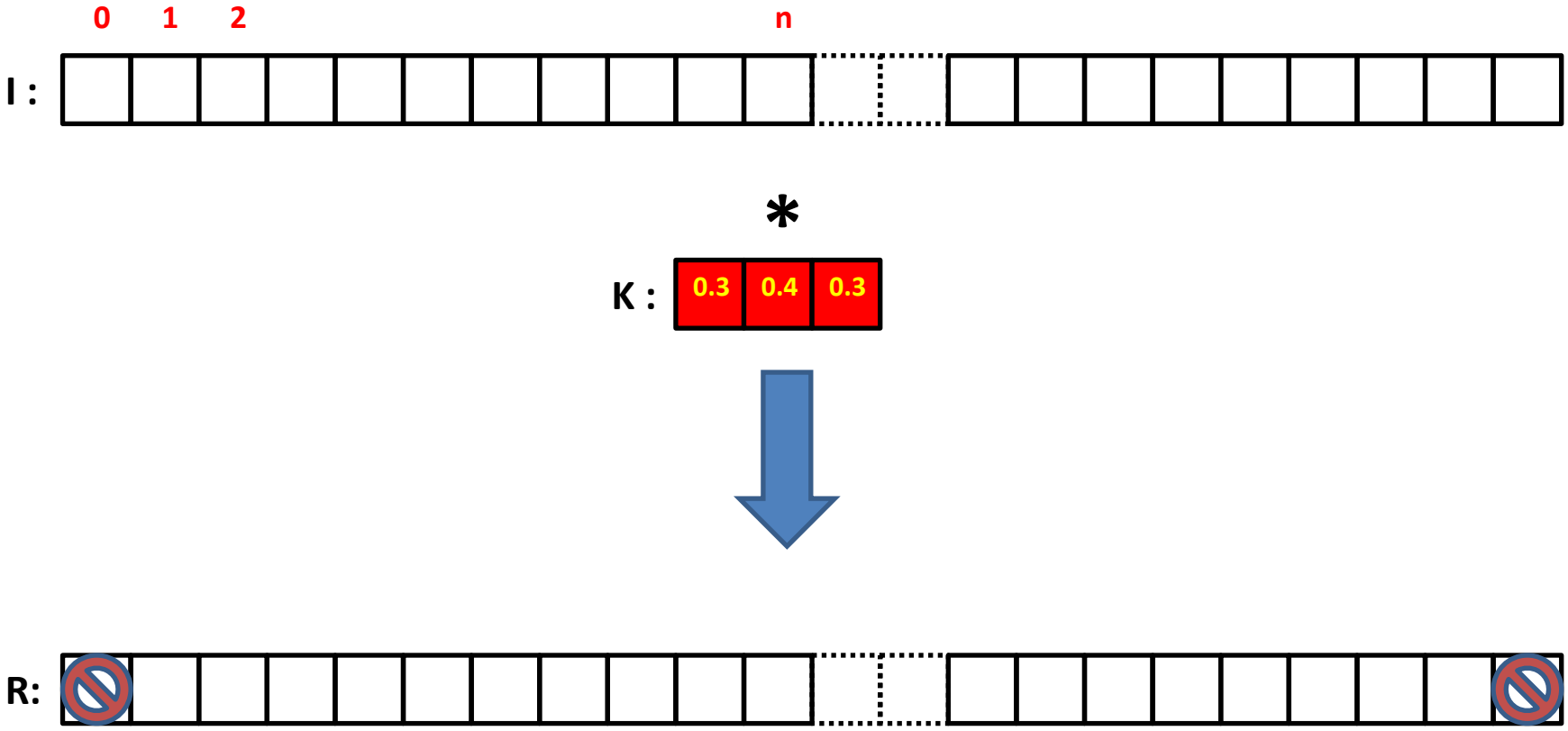
Filtre FIR

- Équation de récurrence :

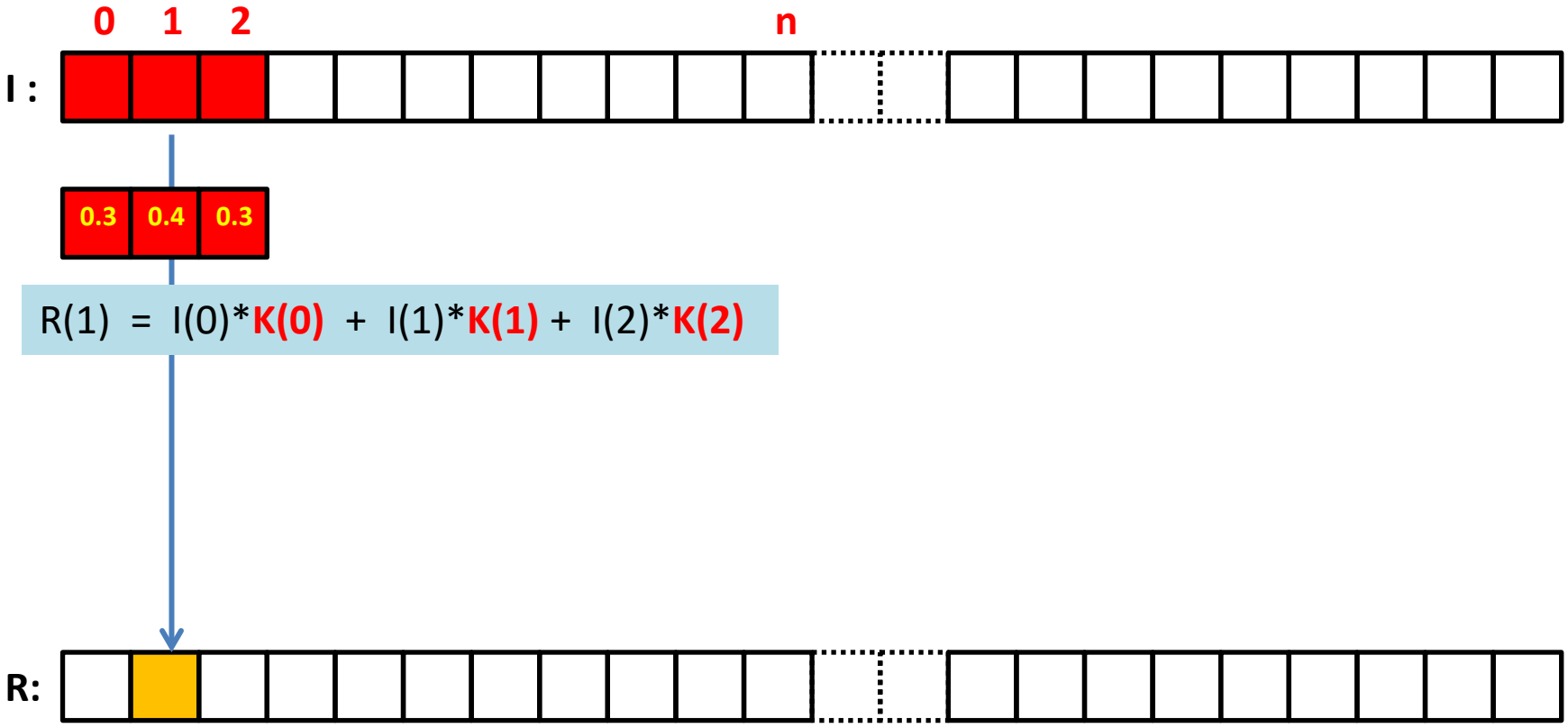
$$S_n = \sum_{i=0}^{p-1} C_i * X_{n-i}$$

- P => Nombre de coefficients
- C_i => Les coefficients
- X_n => échantillon à traiter à l ’instant n
- S_n => résultat du traitement en sortie

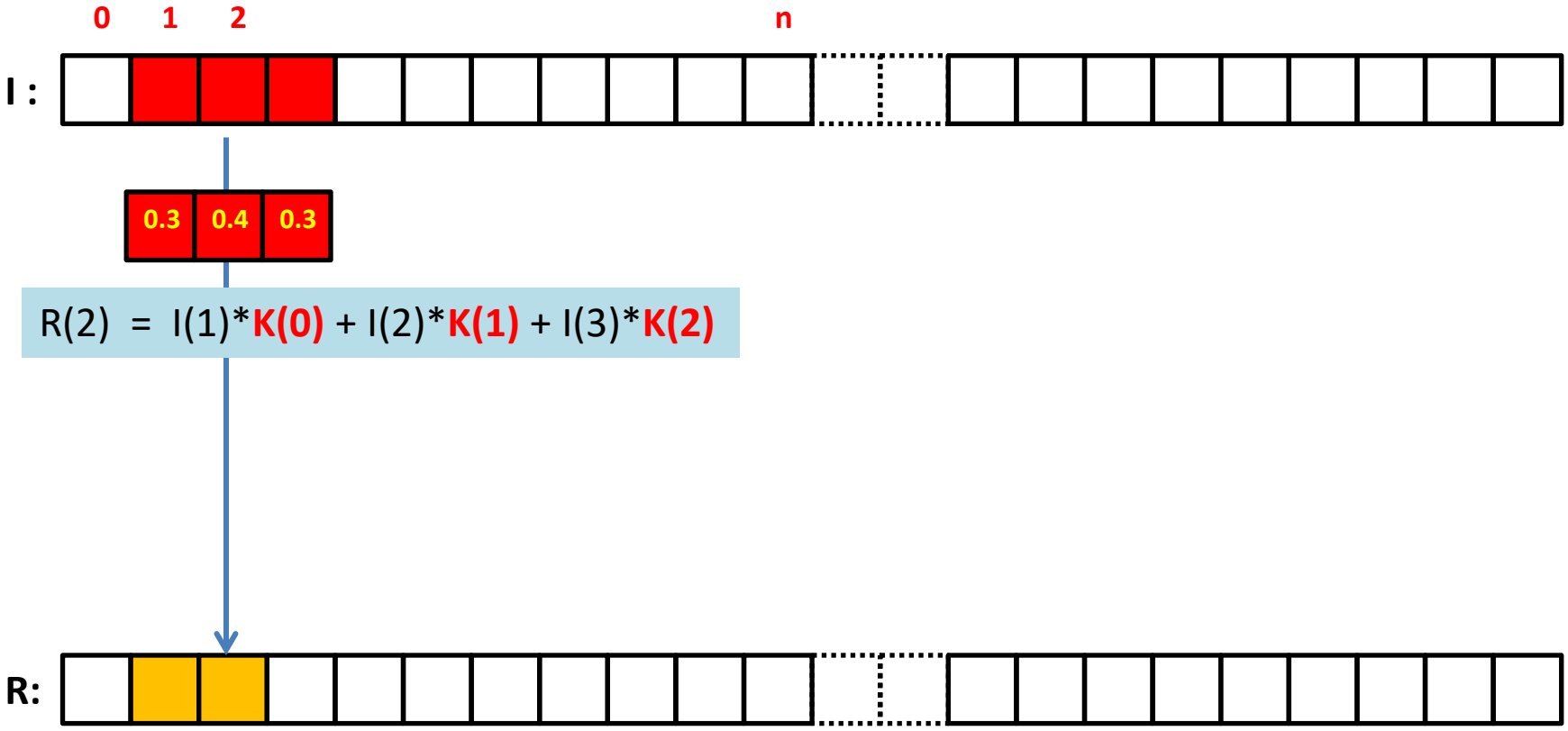
Traitement par convolution



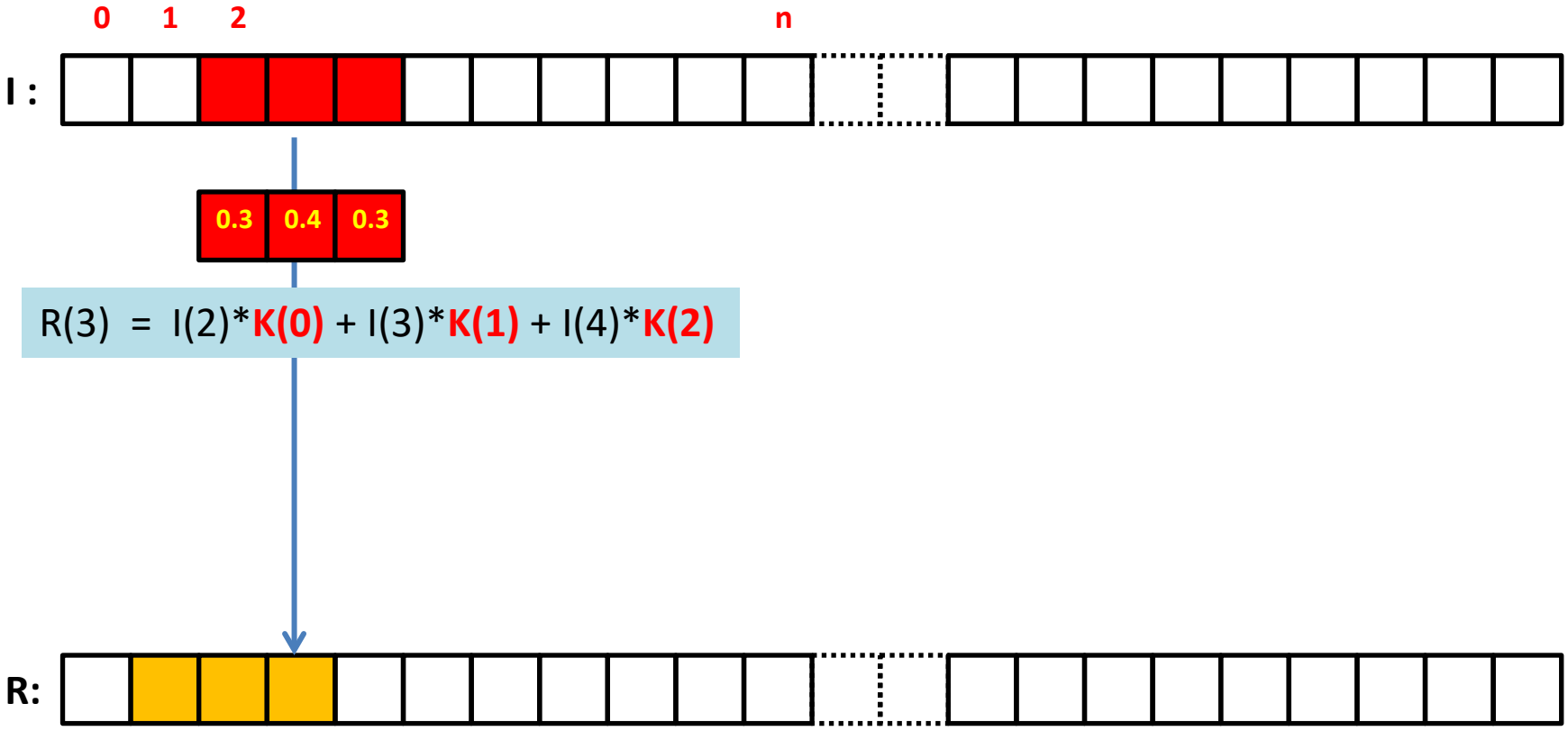
Traitement par convolution



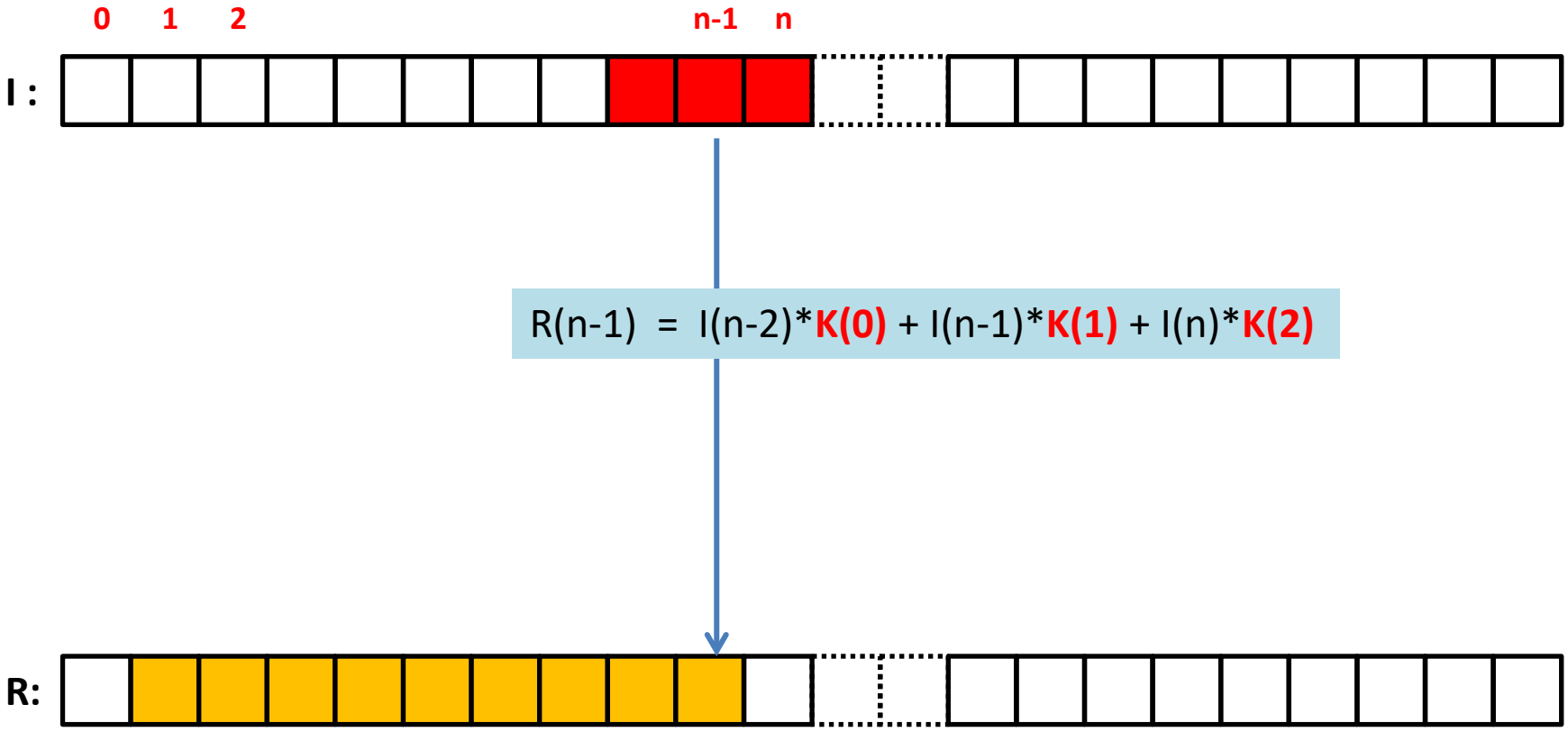
Traitement par convolution



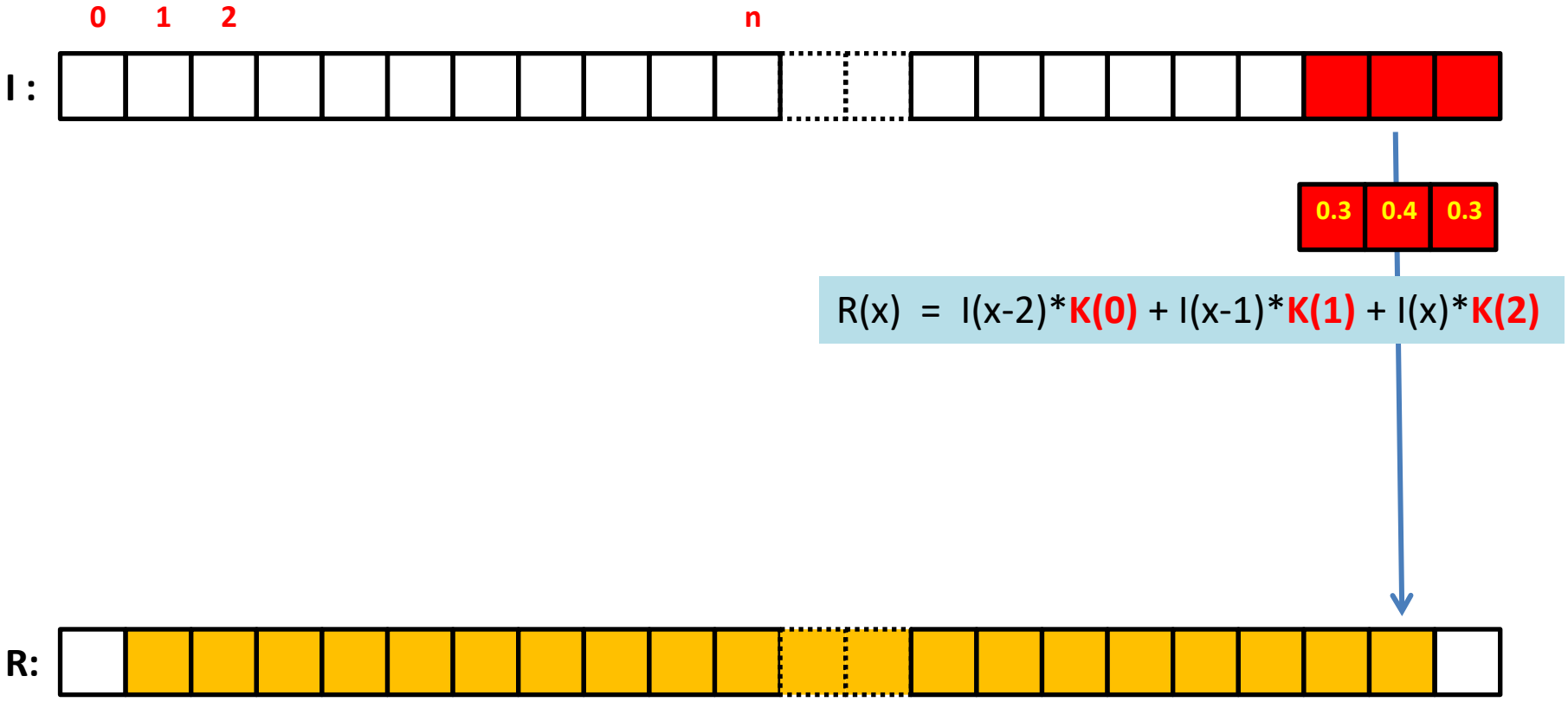
Traitement par convolution



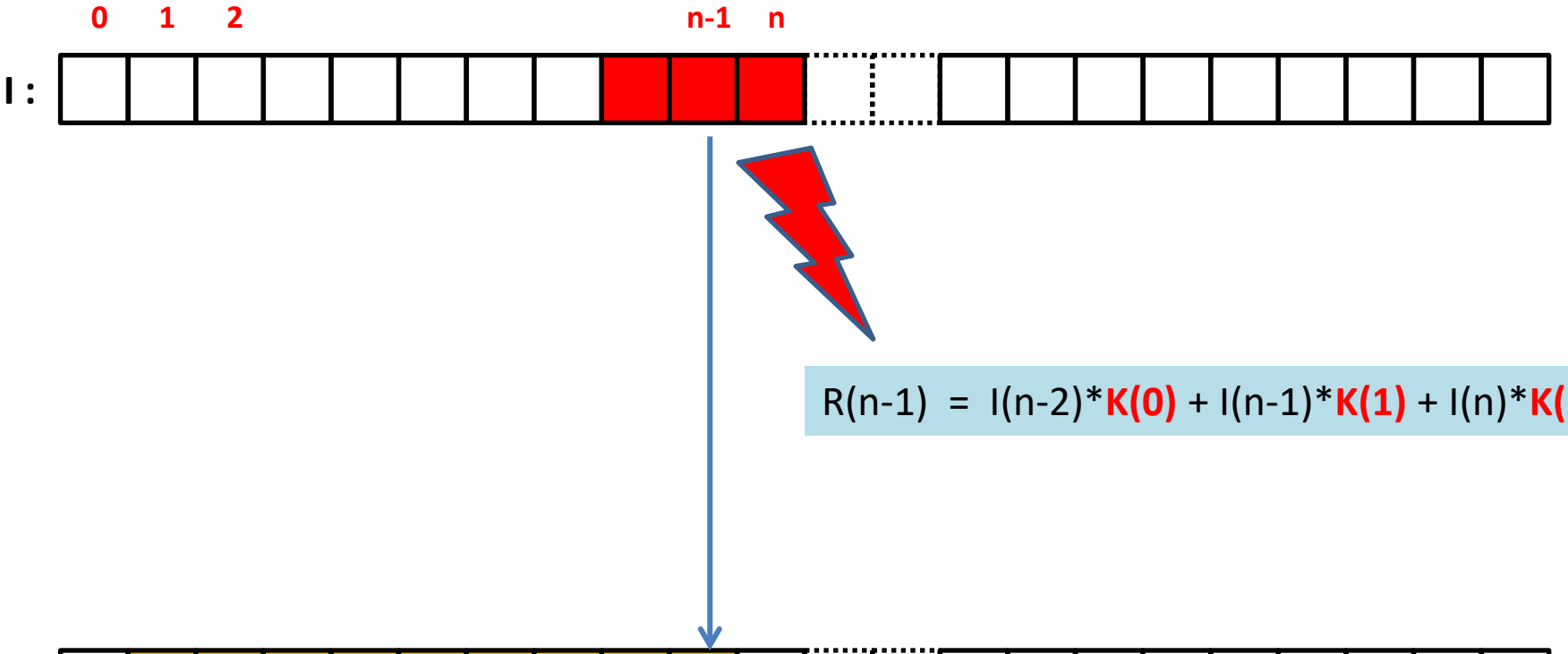
Traitement par convolution



Traitement par convolution



Traitement par convolution

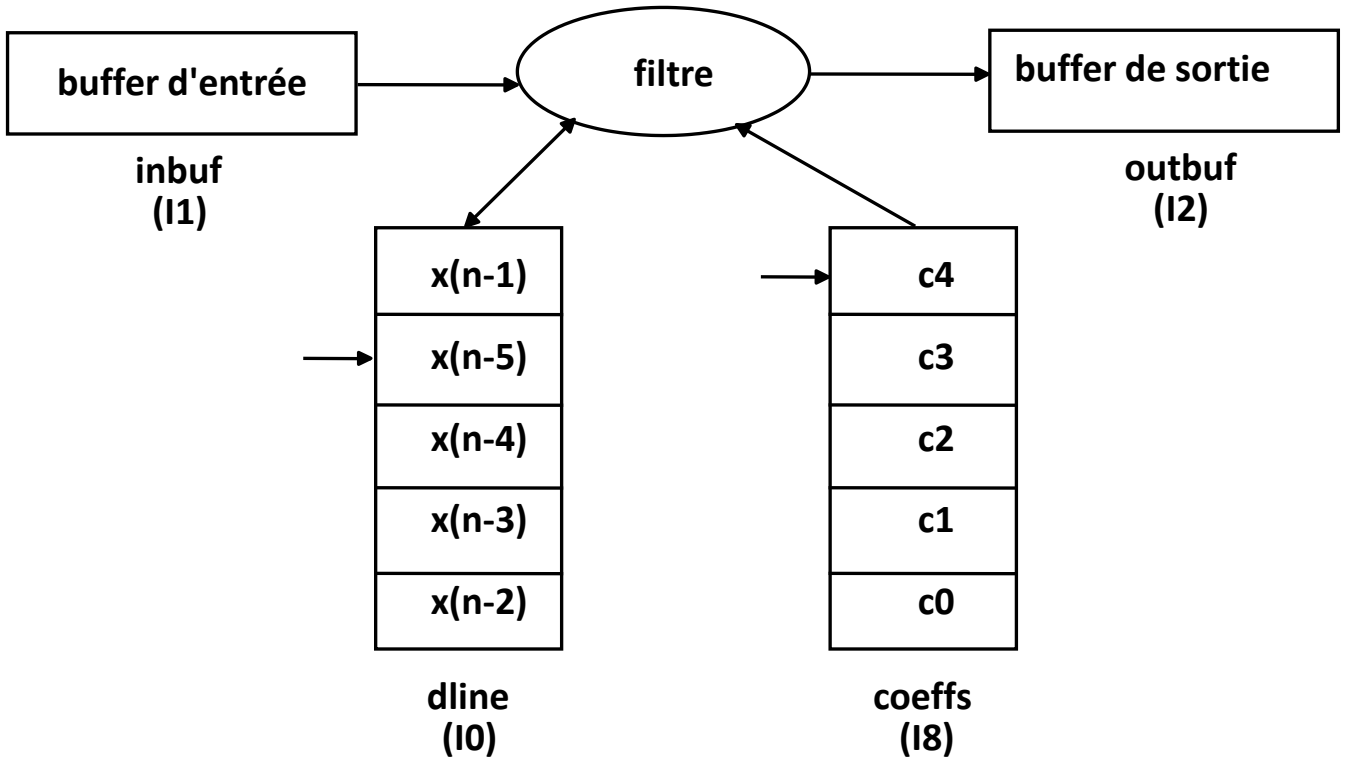


$$R(n-1) = I(n-2)*K(0) + I(n-1)*K(1) + I(n)*K(2)$$

Réalisation ???

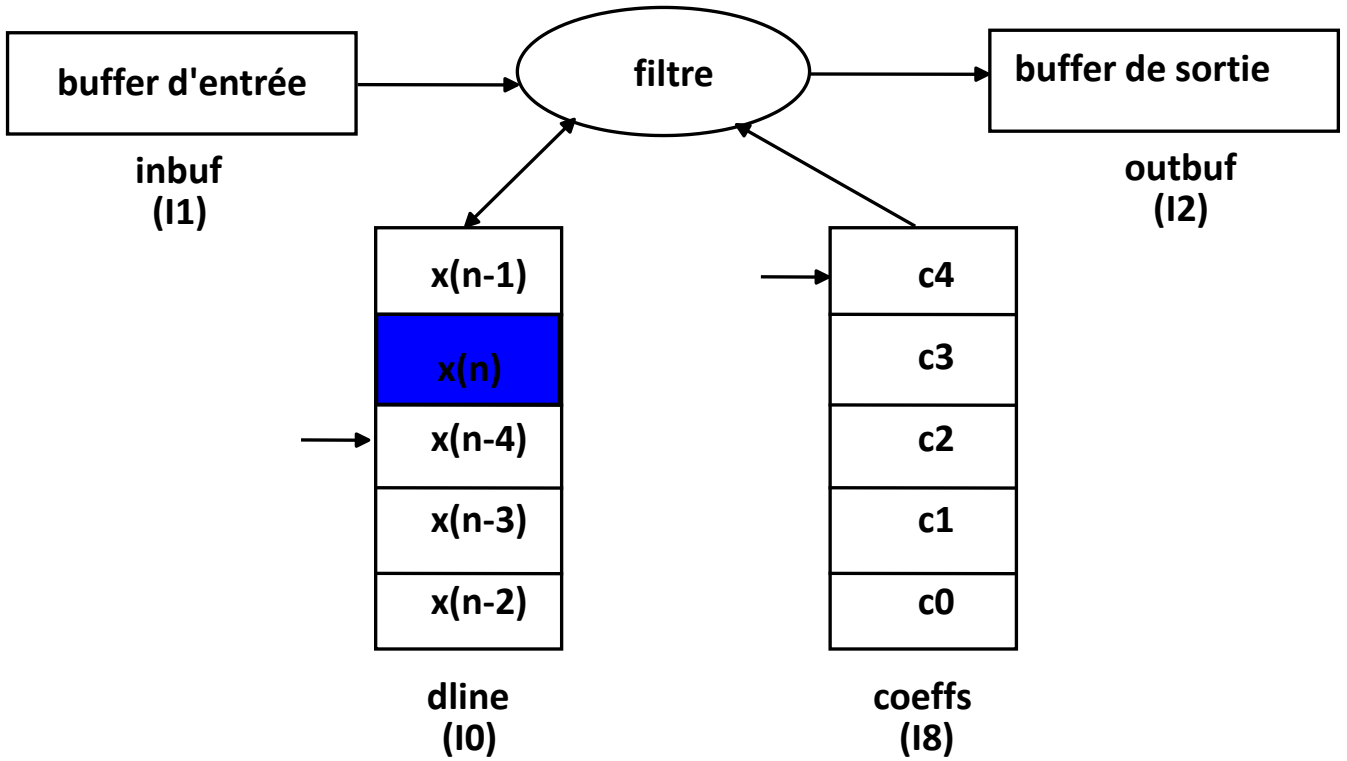
Filtre à réponse Impulsonnelle Finie

$$y(n)= c0\ x(n) + c1\ x(n-1) + c2\ x(n-2) + c3\ x(n-3) + c4\ x(n-4)$$



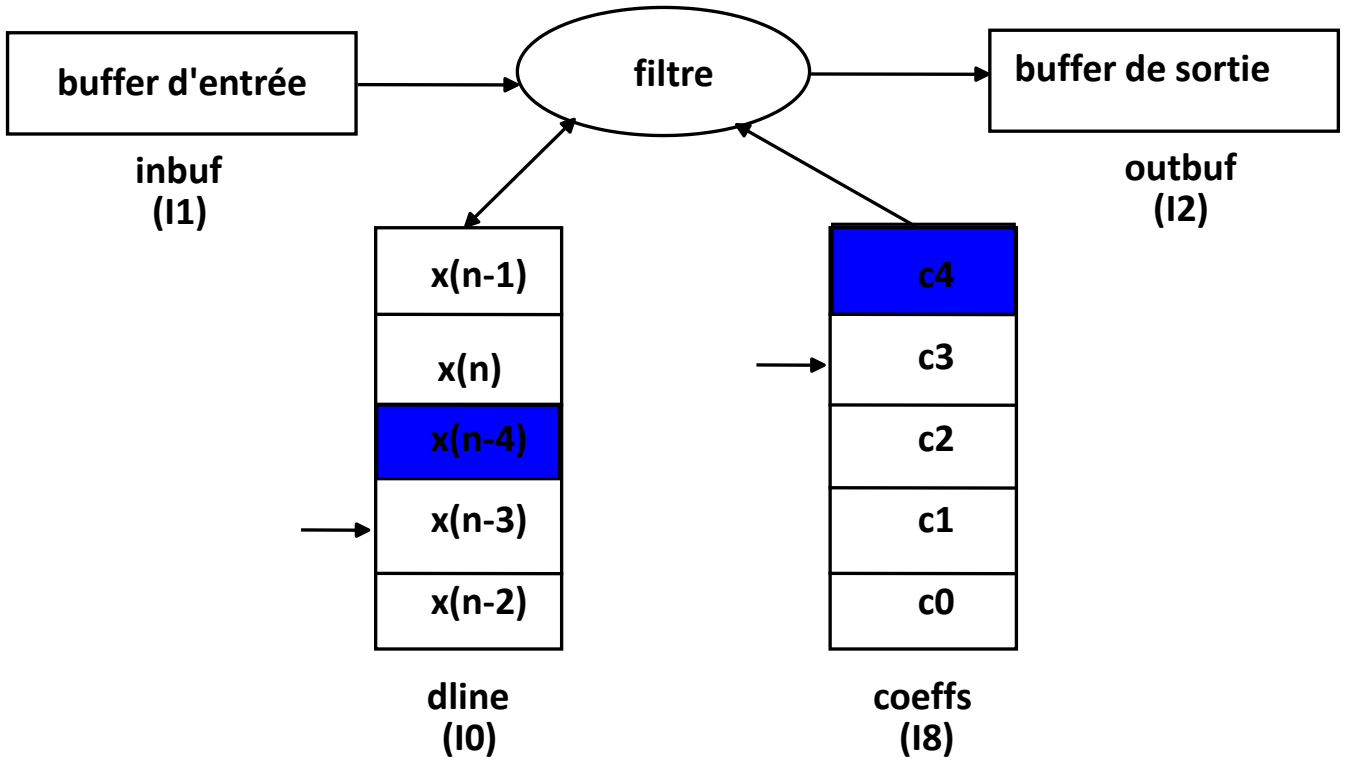
Filtre à réponse Impulsonnelle Finie

$$y(n)= c0\ x(n) + c1\ x(n-1) + c2\ x(n-2) + c3\ x(n-3) + c4\ x(n-4)$$



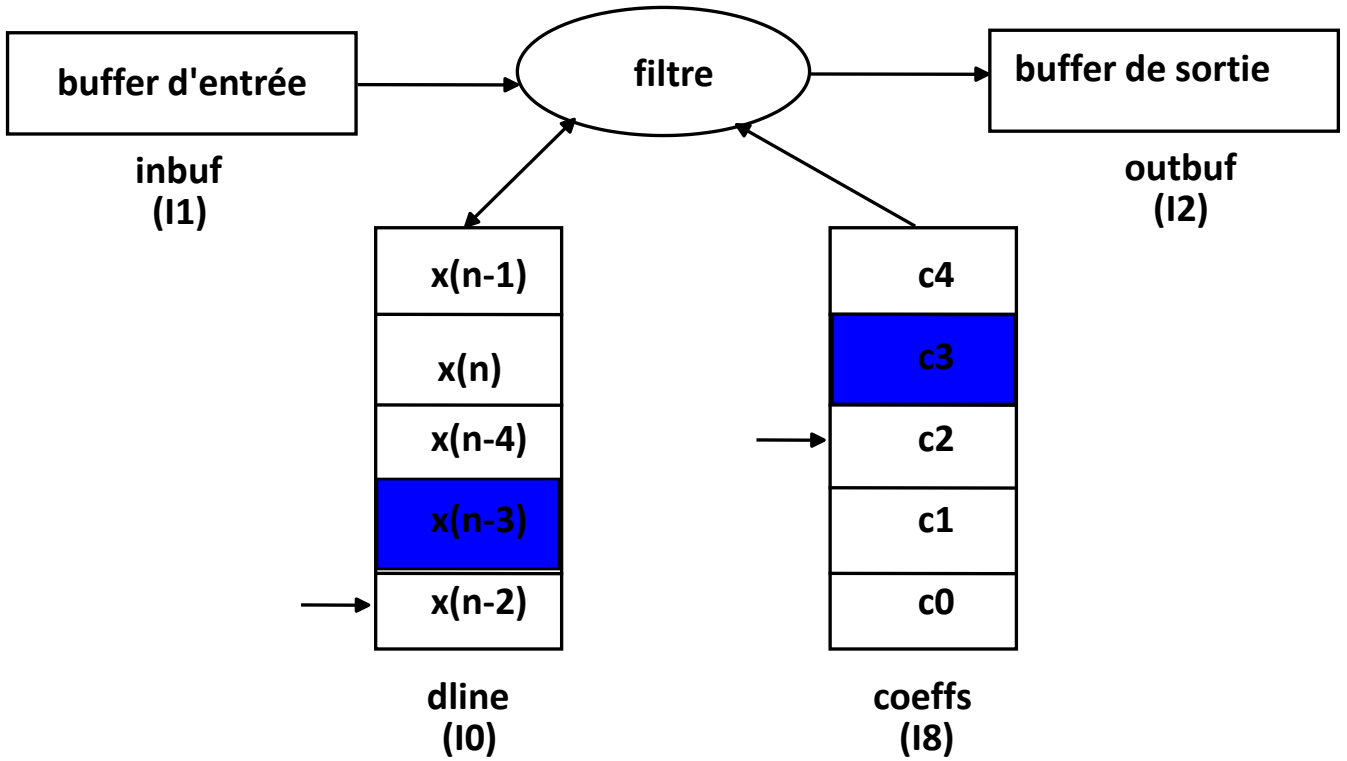
Filtre à réponse Impulsonnelle Finie

$$y(n)= c0\ x(n) + c1\ x(n-1) + c2\ x(n-2) + c3\ x(n-3) + c4\ x(n-4)$$



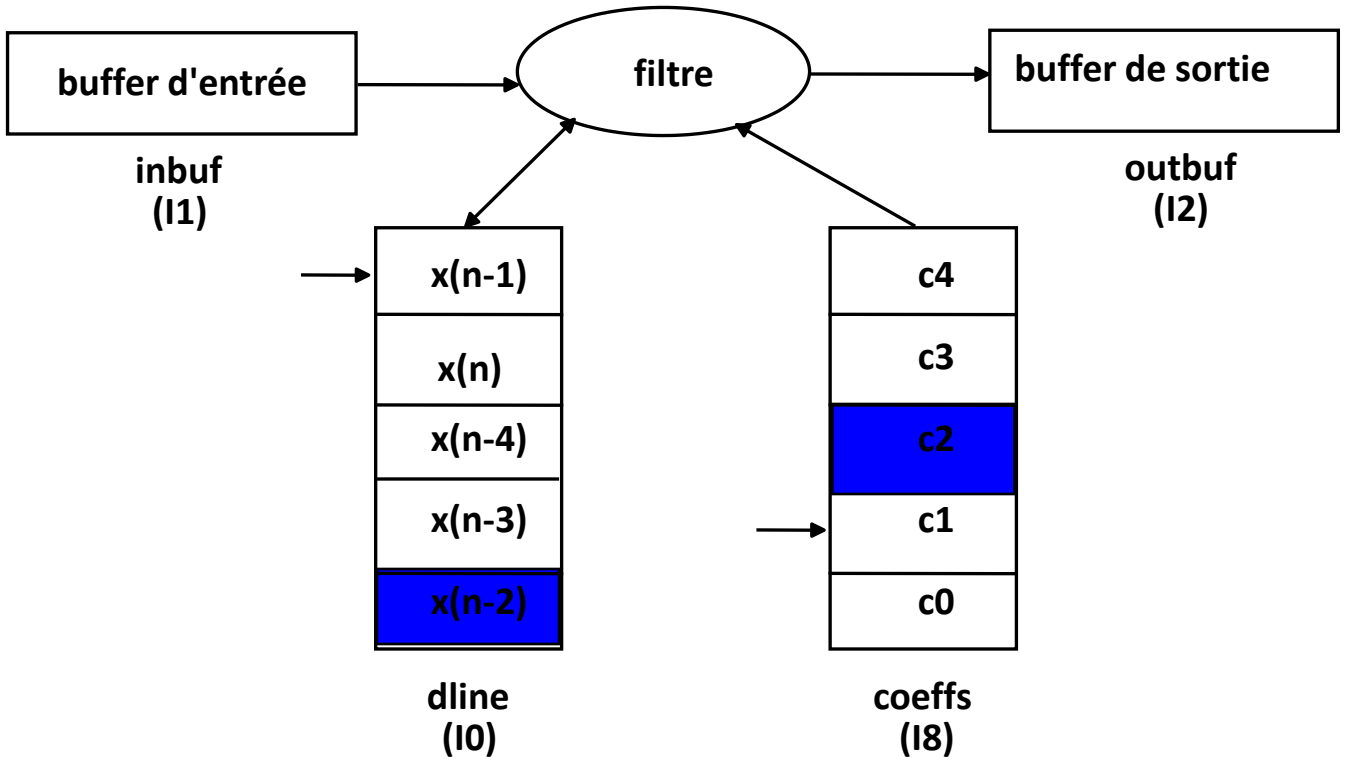
Filtre à réponse Impulsonnelle Finie

$$y(n)= c0\ x(n) + c1\ x(n-1) + c2\ x(n-2) + c3\ x(n-3) + c4\ x(n-4)$$



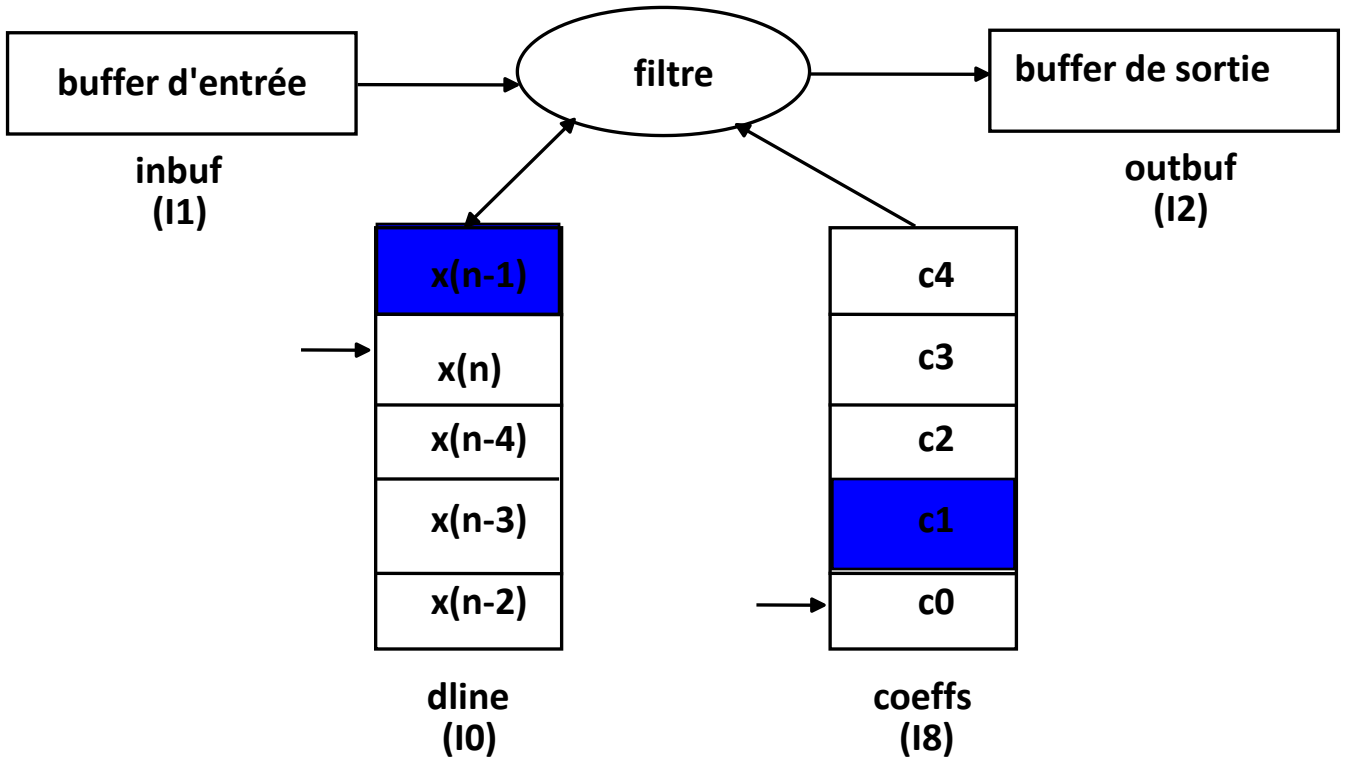
Filtre à réponse Impulsonnelle Finie

$$y(n)= c0\ x(n) + c1\ x(n-1) + c2\ x(n-2) + c3\ x(n-3) + c4\ x(n-4)$$



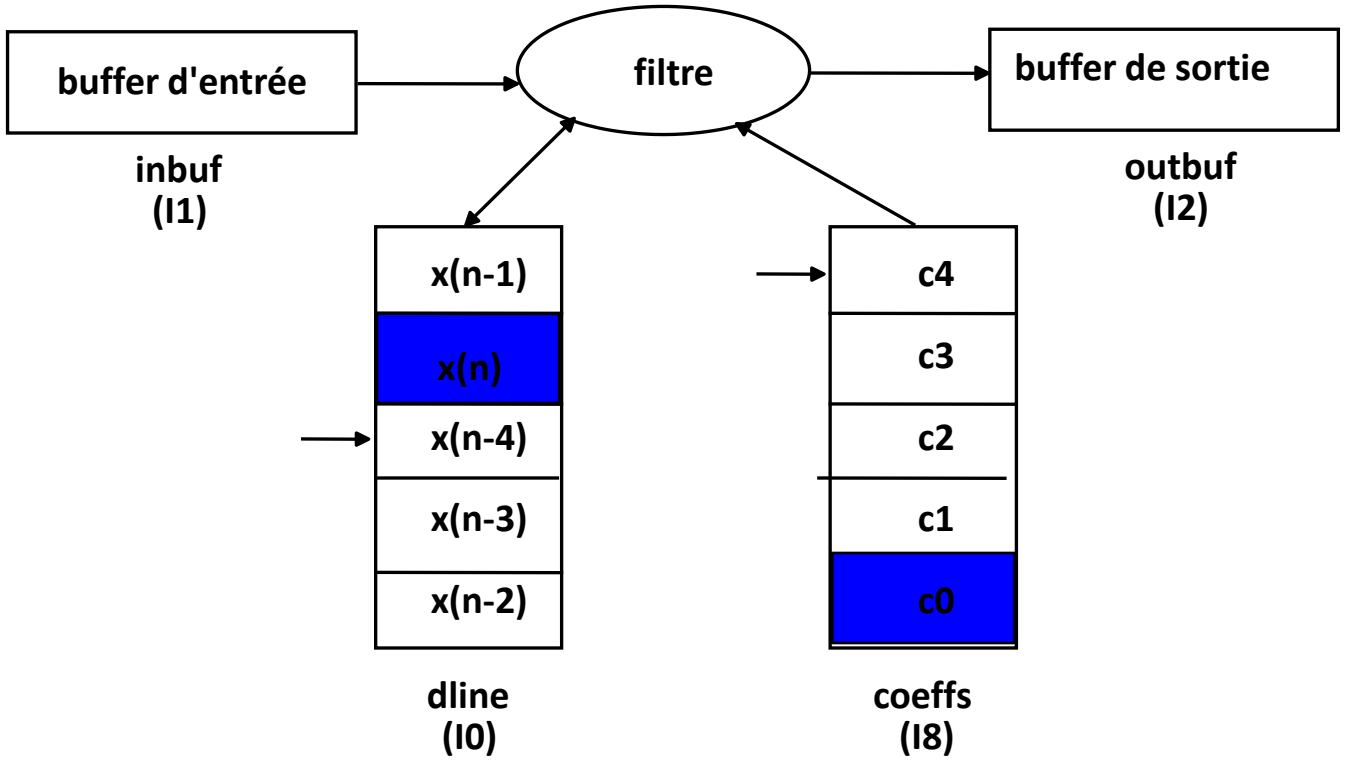
Filtre à réponse Impulsonnelle Finie

$$y(n)= c0\ x(n) + c1\ x(n-1) + c2\ x(n-2) + c3\ x(n-3) + c4\ x(n-4)$$



Filtre à réponse Impulsonnelle Finie

$$y(n) = c0\,x(n) + c1\,x(n-1) + c2\,x(n-2) + c3\,x(n-3) + c4\,x(n-4)$$



Code du sous-programme FIR

```
/*************************************************************************
f12 : résultat multiplication
f8 : accumulation résultat
f0 : échantillons entrant (xn) et précédents en cours de calcul
     puis résultat du calcul
f4 : coefficients
r1 : nombre de coefficients moins 1
*************************************************************************/

.GLOBAL rif;
.EXTERN coefs, dline;
.SEGMENT /PM seg_pmco;

rif:
  r12=r12 xor r12;
  dm(i0,m0)=f0; /* r12 =0 et stockage échantillon entrant dans dline*/
  r8=r8 xor r8;
  lcntr=r1, do macs until lce; /* boucle itérant nb_coeff fois*/
     f0=dm(i0,m0);  /* r8=0 et chargement donnée et coeff issus de dline et coef*/
     f4=pm(i8,m8);
     f12=f0*f4;
     macs: f8=f8+f12; /* multiplie, accumule et charge donnée + coeff itération suivante*/
  f0=f8;      /* stockage résultat dans f0*/
  rts;

rif.END:
```

Mémoires et Fichier Architecture

```
!-------------------------------------------------------------------
.SYSTEM  SHARC_EZKIT_Lite;

.PROCESSOR = ADSP21061;
!   ---------------------------------------------------------
!   Internal memory Block 0
!   ---------------------------------------------------------
.SEGMENT/RAM/BEGIN=0x00020000 /END=0x00020084 /PM/WIDTH=48        seg_rth;
.SEGMENT/RAM/BEGIN=0x00020085 /END=0x00020094 /PM/WIDTH=48        seg_init;
.SEGMENT/RAM/BEGIN=0x00020095 /END=0x000202ff /PM/WIDTH=48        seg_knlc;
.SEGMENT/RAM/BEGIN=0x00020300 /END=0x00021fff /PM/WIDTH=48        seg_pmco;
.SEGMENT/RAM/BEGIN=0x00023000 /END=0x00023fff /PM/WIDTH=32        seg_pmda;
!   ---------------------------------------------------------
!   Internal memory Block 1
!   ---------------------------------------------------------
.SEGMENT/RAM/BEGIN=0x00024000 /END=0x00025fff /DM/WIDTH=32        seg_dmda;
.SEGMENT/RAM/BEGIN=0x00026000 /END=0x00026fff /DM/WIDTH=32 /cheap seg_heap;
.SEGMENT/RAM/BEGIN=0x00027000 /END=0x00027e7f /DM/WIDTH=32        seg_stak;
.SEGMENT/RAM/BEGIN=0x00027e80 /END=0x00027fff /DM/WIDTH=32        seg_knld;
.ENDSYS;
```