# C6. Aplicatii C2x

Exemple

- Adresarea Bit-reversed
- instructiunea MAC
- Implementare FIR
- Generare Sinus (oscilator digital)
- Tabele

# Moduri de adresare C2x – Recapitulare

## 1. Modul de adresare direct:

Urmatoarea secventa permite adunarea la acumulator a continutului adresei 695h (AC)=(AC)+(0695h)

0695h = 0000 0110 1|001 0101b

DP=13 dma= 21

```
LDPK 13          ; DP=13=0dh
ADD 21  ; (AC)=(AC)+(21)
ADD 21,3         ; (AC)=(AC)+(21)*8
```

## 2. Modul de adresare indirect:

```
LARP 3           ;ARP=3
LRLK AR3,695h    ;AR3=0695h
ADD *, 3         ;(AC)=(AC)+(695h)*8
```

{ * | * + | * – | * 0 + | * 0 – | * BR0 + | * BR0 –} ;BRO – propagarea inversata a bitului de Carry

## 3. Modul de adresare imediata

```
ADLK 163,2       ; (AC)=(AC)+163*4
                 ; ADLK   - adaugă la acumulator valoarea imediată 16b cu deplasare
```

## Problema 1:

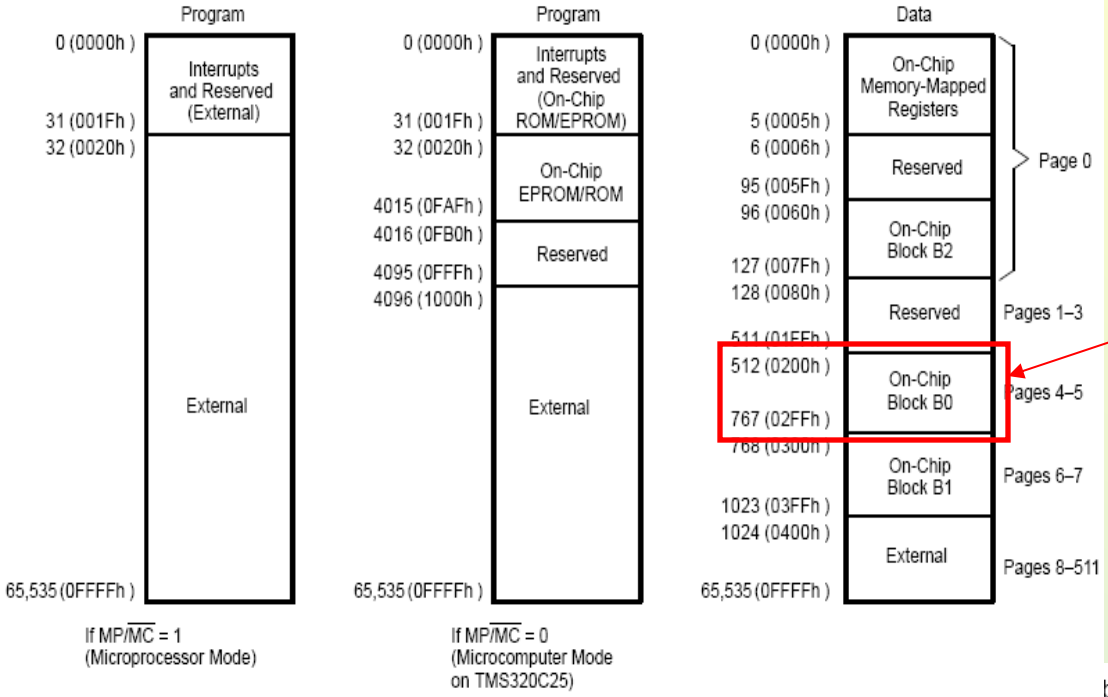
**Blocul B0 contine:**

| 200h | 201h | 202h | 203h | 204h | 205h | 206h | 207h |
|---|---|---|---|---|---|---|---|
| 10h | 20h | 40h | 80h | 100h | 200h | 400h | 800h |

**Care este continutul blocului B2 (60h-67h) dupa secventa urmatoare de program?**

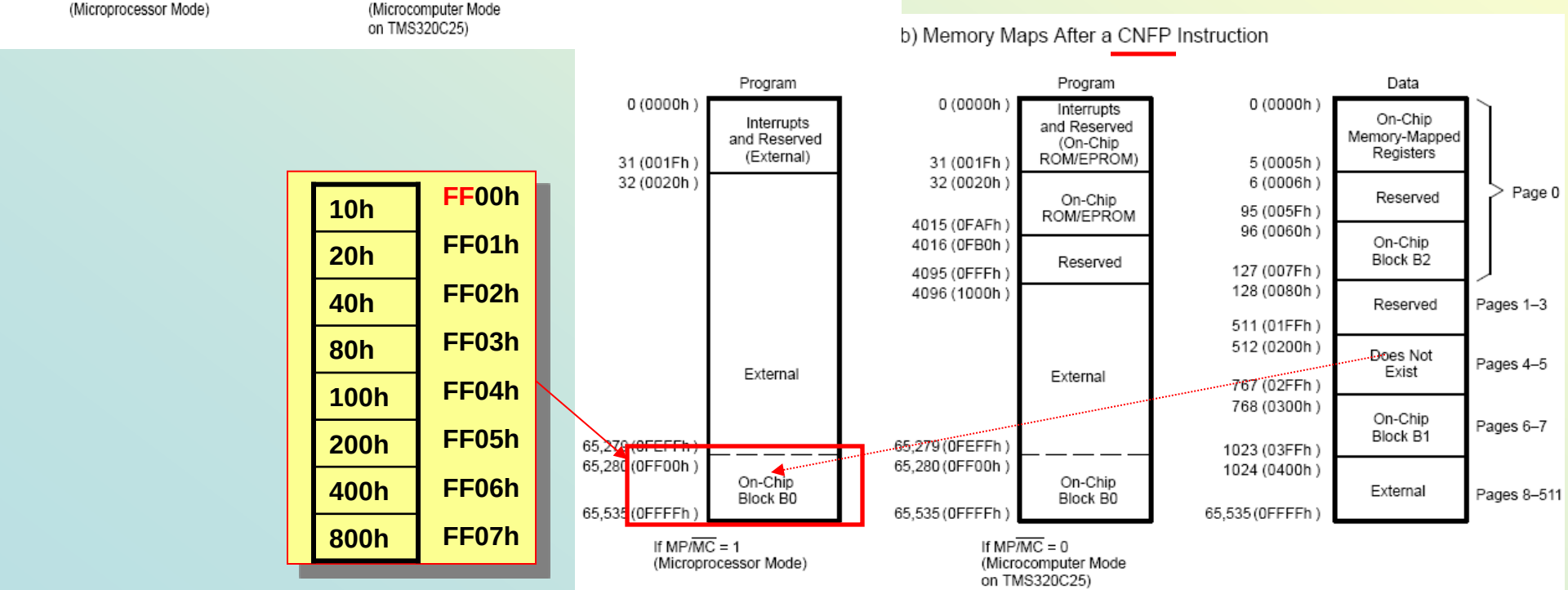
```
CNFP
LARP 1
LRLK  AR1,60h
LRLK  AR0,4
RPTK 7
BLKP  FF00h,*BR0+
```

(a) Memory Maps After a CNFD Instruction




| | |
|---|---|
| 10h | 200h |
| 20h | 201h |
| 40h | 202h |
| 80h | 203h |
| 100h | 204h |
| 200h | 205h |
| 400h | 206h |
| 800h | 207h |

(b) Memory Maps After a CNFP Instruction




| | |
|---|---|
| 10h | FF00h |
| 20h | FF01h |
| 40h | FF02h |
| 80h | FF03h |
| 100h | FF04h |
| 200h | FF05h |
| 400h | FF06h |
| 800h | FF07h |

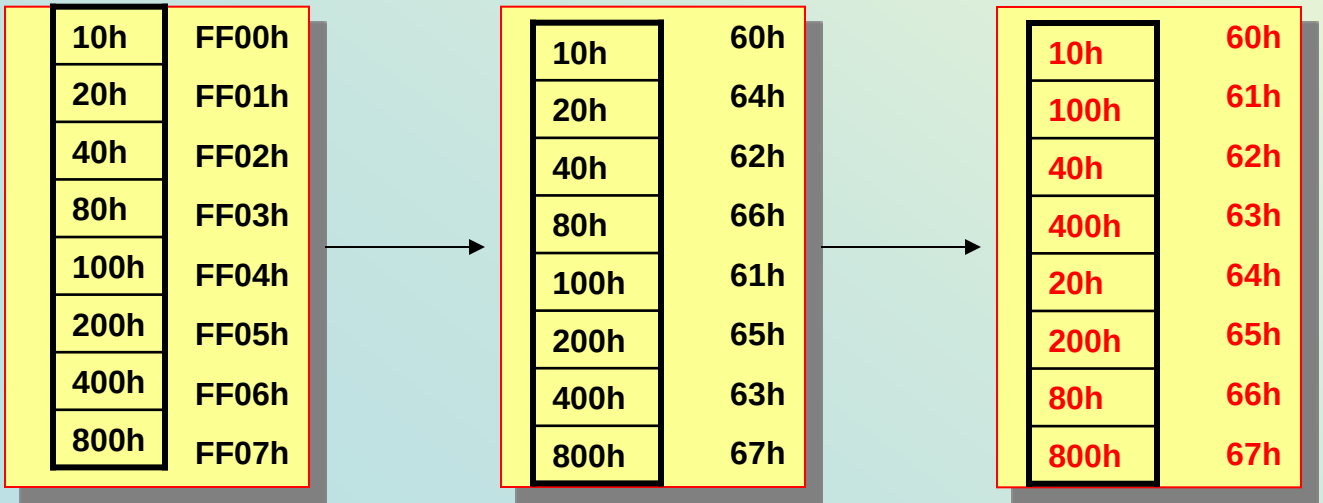
## Rezolvare:

**Blocul B0 (200h-2FFh) contine:**

| 200h | 201h | 202h | 203h | 204h | 205h | 206h | 207h |
|---|---|---|---|---|---|---|---|
| 10h | 20h | 40h | 80h | 100h | 200h | 400h | 800h |

**Care este continutul blocului B2 (60h-67h) dupa secventa urmatoare de program?**

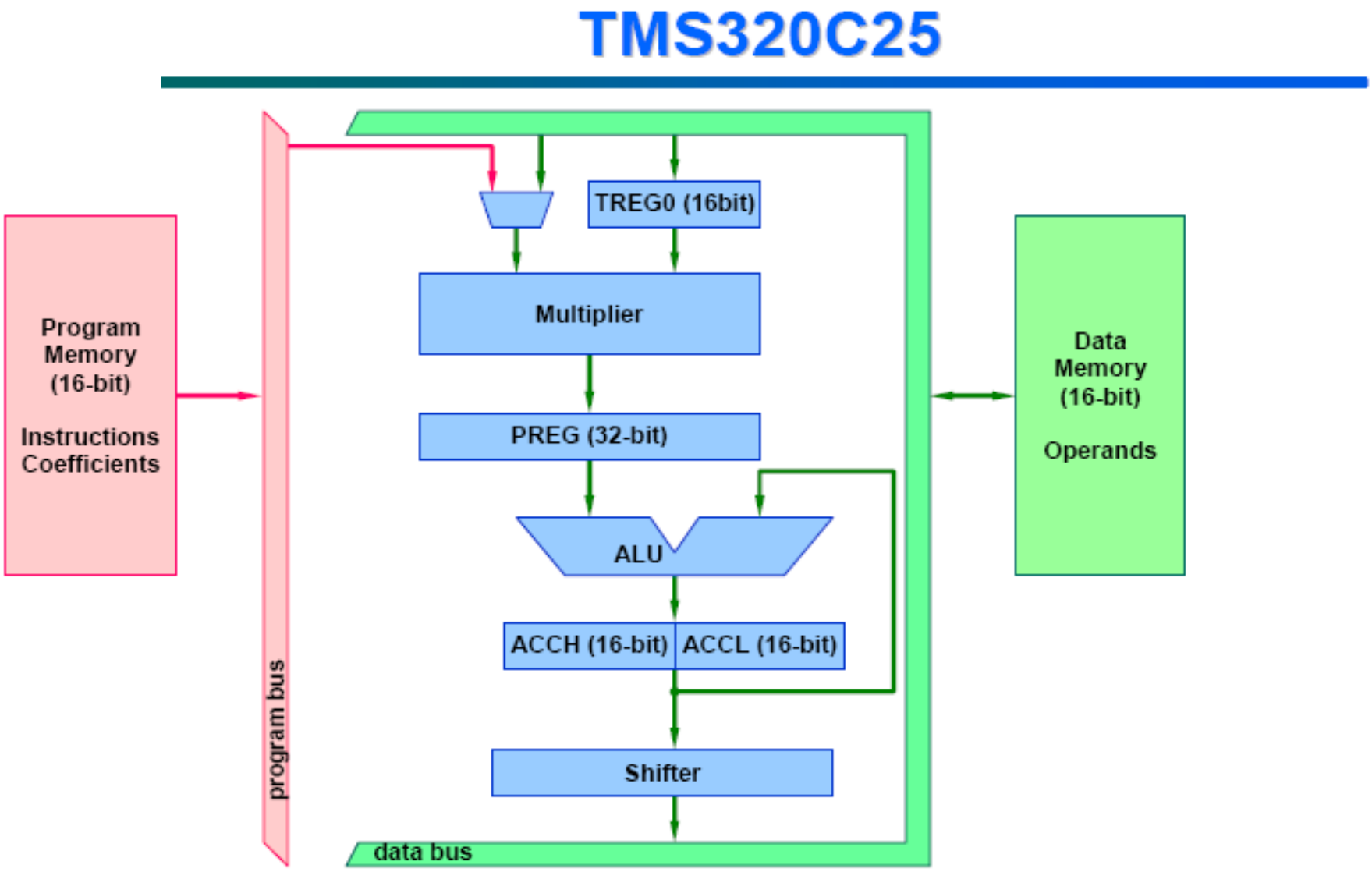
```
CNFP                ;conf Bloc B0 mem program (FF00h-FFFFh)
LARP 1              ; ARP=1
LRLK  AR1,60h       ; AR1=60h
LRLK  AR0,4         ;AR0=4
RPTK 7              ;repeta urm instr de 8 ori
BLKP  FF00h,*BR0+
```




**Adresarea bit-reversed**

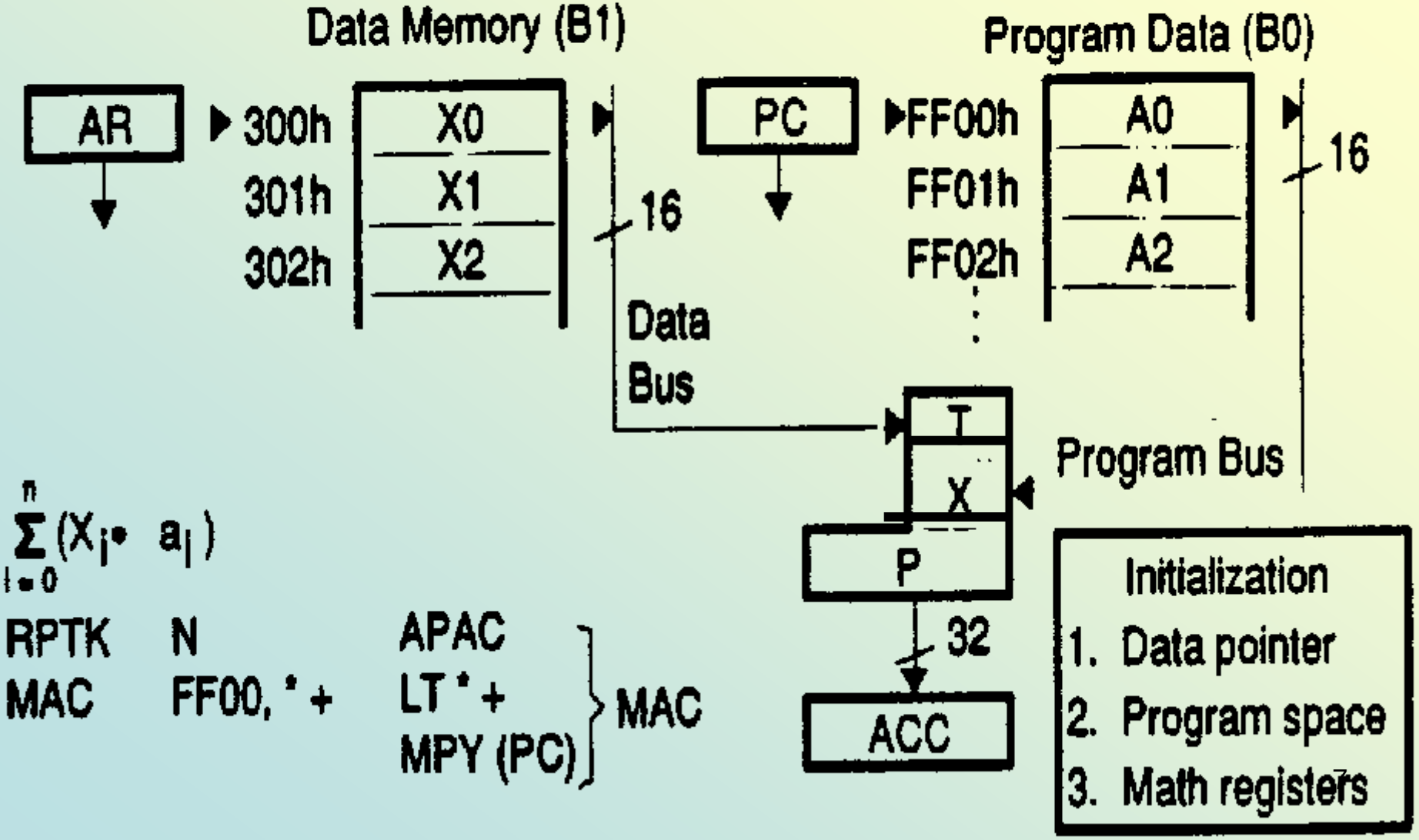
0 (000) => 0 (000)
1 (001) => 4 (100)
2 (010) => 2 (010)
3 (011) => 6 (110)
4 (100) => 1 (001)
5 (101) => 5 (101)
6 (110) => 3 (011)
7 (111) => 7 (111)

# Central ALU (CALU)



Problema 2:

# Multiply with Accumulate (MAC)



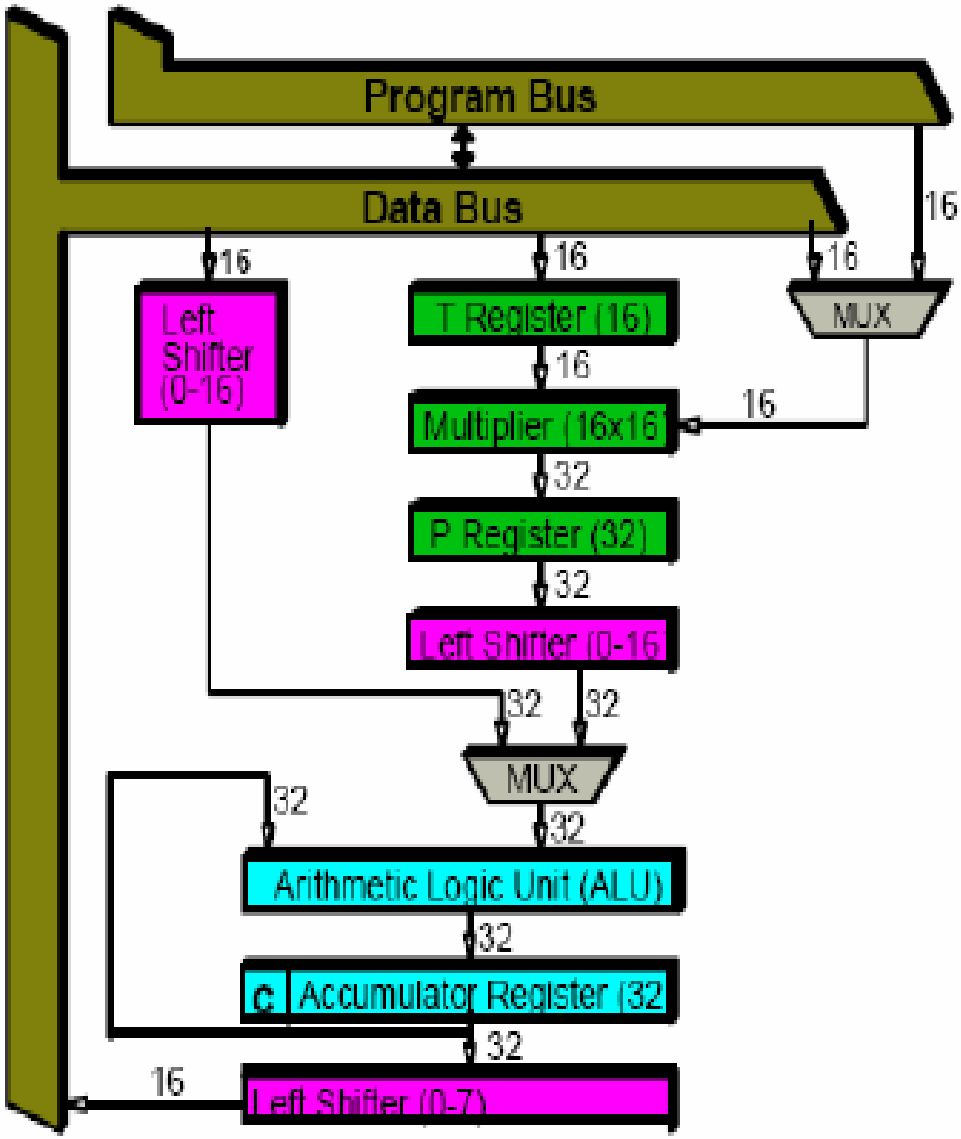
Problema 3:

# Implementare FIR pe DSP conventional

- TMS320C2x (1986)

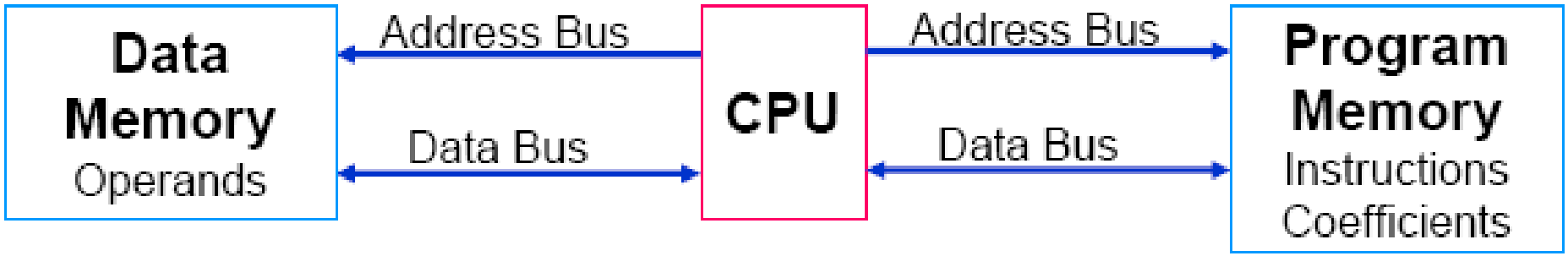
```
ZAC || LTD;
LTD || MPY;

RPTK N-1
MACD

APAC || MPY;
APAC;
```

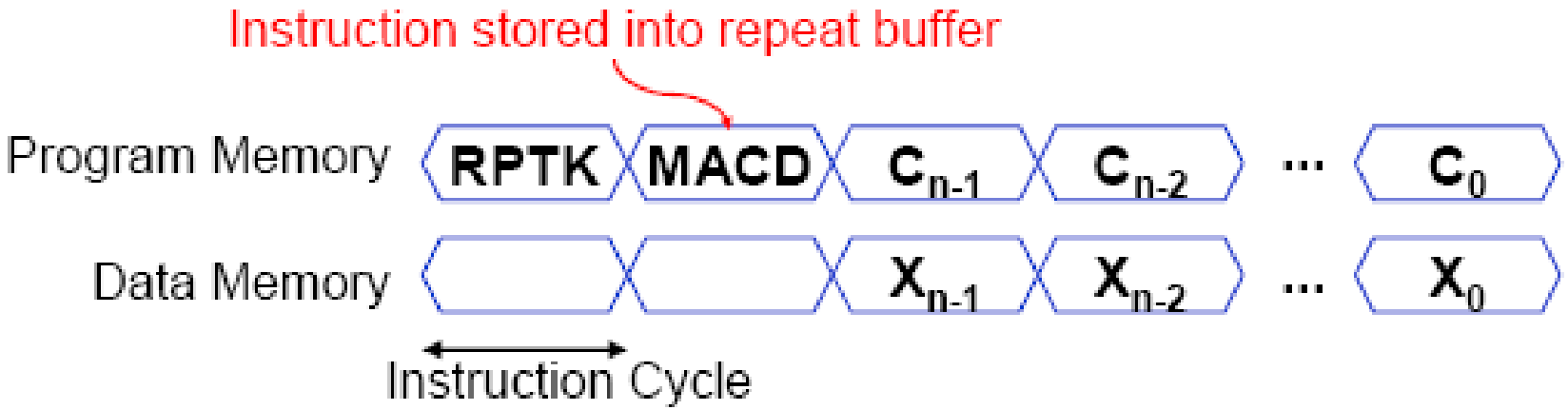
Exécution en N cycles



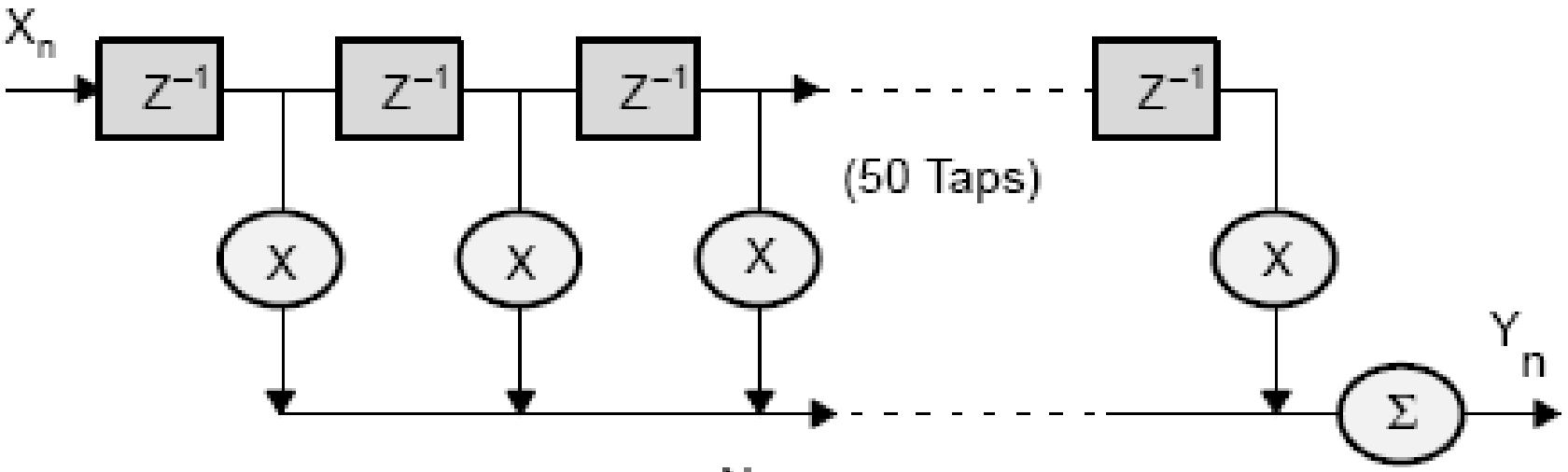
# Repeat Buffer in ‘C25

```
    RPTK    n-1          ; repeat the following
*                        ; instruction n times
    MACD    Xn-1,Cn-1    ; perform MAC with
*                        ; delay line update
```

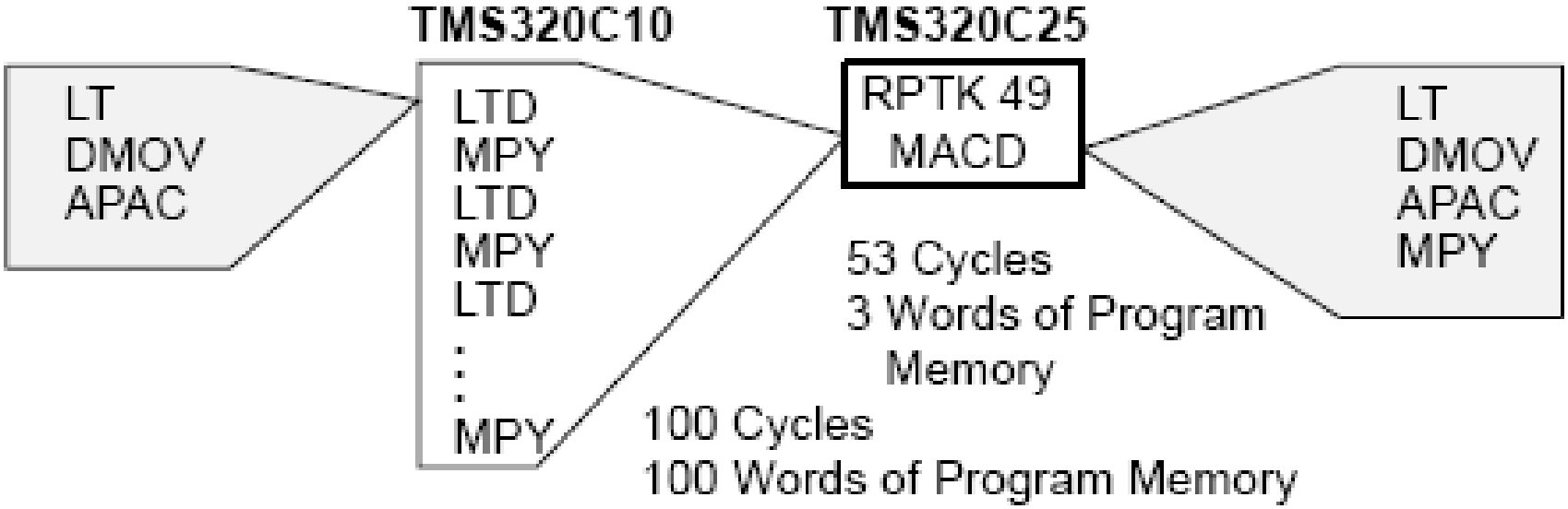


# TMS320C25 Sum of Products Example




$$Y_n = \sum_{k=0}^{N} b_k \; X_{(n-k)}$$



# FIR Filtering: A Motivating Problem

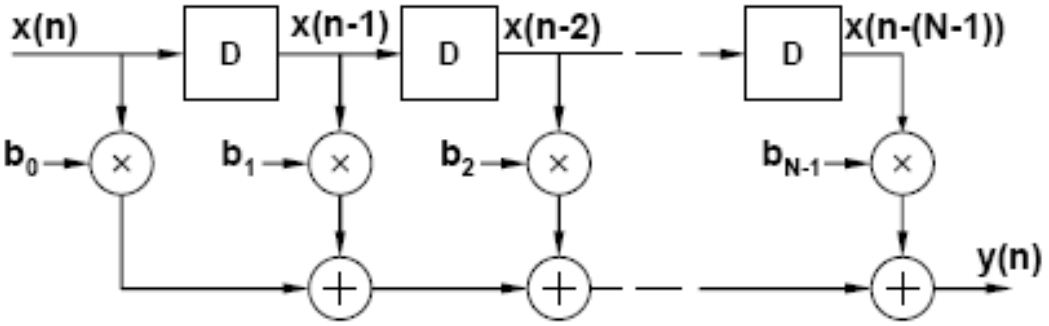
**Transfer Function**

$$H(z) = \sum_{k=0}^{N-1} b_k z^{-k}$$

**Difference Equation**

$$y[n] = \sum_{k=0}^{N-1} b_k x[n-k]$$

**Block Diagram**




**High-Level Language Description**

```
y = 0;x[0]=xin;
for(k=N-1; k>=0; k--) {
    y += x[k]*b[k];
    x[k]=x[k-1];
}
```

**Problema 4:**

# Metode/Algoritmi pentru generarea semnalelor

| Algorithm | Speed | Quality at low freq. | Accuracy | Stability | Orthogonality of sin/cos pair | Memory requirements | Modulation capability |
|---|---|---|---|---|---|---|---|
| Polynoms | Slow | Excellent | Good | Excellent | Excellent | Medium | Excellent |
| Lookup | Medium | Excellent | Fair | Excellent | Good | Huge | Good |
| Complex Oscillator | Fast | Limited | Good | Poor | Good | Low | Limited |
| 2nd order oscillator | Fastest | Poor | Good | Excellent | Good | Low | Difficult |

```
;Program generare sinus pentru placa SIDERAL
        fc      equ 20000
        fe      equ 40;
        PERD    equ fc/(4*fe)-1
        COS     EQU 60h         ; cos 9°
        yN      equ 70h         ;
        yN_1    EQU 71H         ;
        yN_2    EQU 72H         ;

begin   ldpk 0          ;DP=0
        lalk PRD        ;ACCL=PERD
        sacl 3          ;PRD=ACCL
        lack 8          ;ACCL=00…1000B
        or   4          ;ACCL=ACCL or IMR
        sacl 4          ;IMR=ACCL
        b start

          ORG 24        ;TINT
```

```
start   spm     1               ; PM=01, ACC=2*P
        rovm                    ; normal OVM
        ssxm                    ; sign extention
        lalk    7e6dh           ; ACCL= cos 9° in Q15
        sacl    COS             ; COS=ACCL
        lalk    1350h           ; ACCL= sin 9°
        sacl    yN_1            ; yN_1= ACCL
        zac                     ; ACC=0
        sacl    yN_2            ; yN_2= 0
        eint
loc     b loc                   ; wait interrupt
        end
```

```
aorg   24;
       lt     COS          ; T0 reg = cos 9° in Q15
       mpy    yN_1         ; P reg = cos 9° * y(n-1) in Q30
       pac                 ; ACC = 2 * cos 9° * y(n-1) in Q30
       sub    yN_2,15      ; ACC = 2*cos 9°* y(n-1) - y(n-2) in Q30
       dmov   yN_1         ; y(n-2) = y(n-1)
       sach   yN_1,1       ; y(n-1) = new y in Q15
       RPTK   7
       Sfr                 ;ACCH=ACCH/256
       adlk   127,15       ;scaling
      SACH    yN           ;ACCH= yN
       nop
       out    yN,PA0       ; output new value of y
       nop
       eint
       ret
```

5. Completati tabelul de mai jos si comentati executia unde este cazul.
Starea initiala la executia secventei este (flagul CNF=0, B0 –Memoria Date):

| Adresa | 300h | 301h | 302h | 303h | 304h | 305h | 306h | 307h |
|---|---|---|---|---|---|---|---|---|
| Continut | **100h** | **200h** | **400h** | **800h** | **700h** | **500h** | **300h** | **100h** |

| | **ACCL** | **DP** | **ARP** | **AR0** | **AR6** | **AR7** | **77h** | **72h** | **CNF** | **OBS.** |
|---|---|---|---|---|---|---|---|---|---|---|
| LARP 7 | | | | | | | | | | |
| LRLK AR6,70h | | | | | | | | | | |
| LRLK AR7,200h | | | | | | | | | | |
| RPTK 7 | | | | | | | | | | |
| BLKD 300h,*+ | | | | | | | | | | |
| CNFP | | | | | | | | | | |
| LRLK AR0,4 | | | | | | | | | | |
| LARP 6 | | | | | | | | | | |
| RPTK 7 | | | | | | | | | | |
| BLKP FF00h,*BR0+ | | | | | | | | | | |
| LDPK 0 | | | | | | | | | | |
| LAC 76h | | | | | | | | | | |
| SUB 72h,1 | | | | | | | | | | |

# - Comentarii, detalii

**BLKD** – muta bloc din mem de date in mem de date: **B1** → **B0(d)**

(300h-3FFh → 200h-2FFh)

| Adresa B1 | 300h | 301h | 302h | 303h | 304h | 305h | 306h | 307h |
|---|---|---|---|---|---|---|---|---|
| Adresa B0(d) | 200h | 201h | 202h | 203h | 204h | 205h | 206h | 207h |
| Continut | **100h** | **200h** | **400h** | **800h** | **700h** | **500h** | **300h** | **100h** |

**CNFP** – B0 mem program: **B0(p)**

(FF00h-FFFFH)

| Adresa B0(p) | FF00h | FF01h | FF02h | FF03h | FF04h | FF05h | FF06h | FF07h |
|---|---|---|---|---|---|---|---|---|
| Continut | **100h** | **200h** | **400h** | **800h** | **700h** | **500h** | **300h** | **100h** |

**BLKP** – muta bloc din mem program in mem date: **B0(p)** → **B2**

FF00h-FFFFH → 60h-7Fh

| Adresa B0(p) | FF00h | FF01h | FF02h | FF03h | FF04h | FF05h | FF06h | FF07h |
|---|---|---|---|---|---|---|---|---|
| Adresa B2 | 70h | 74h | 72h | 76h | 71h | 75h | 73h | 77h |
| Continut | **100h** | **200h** | **400h** | **800h** | **700h** | **500h** | **300h** | **100h** |

Raspuns

| | ACCL | DP | ARP | AR0 | AR6 | AR7 | 77h | 72h | CNF | OBS. |
|---|---|---|---|---|---|---|---|---|---|---|
| LARP 7 | | | 7 | | | | | | | |
| LRLK AR6,70h | | | | | 70h | | | | | |
| LRLK AR7,200h | | | | | | 200h | | | | |
| RPTK 7 | | | | | | | | | | |
| BLKD 300h,*+ | | | | | | | | | | # |
| CNFP | | | | | | | | | 1 | |
| LRLK AR0,4 | | | | 4 | | | | | | |
| LARP 6 | | | 6 | | | | | | | |
| RPTK 7 | | | | | | | | | | |
| BLKP FF00h,*BR0+ | | | | | | | | | | # |
| LDPK 0 | | 0 | | | | | | | | |
| LAC 76h | 800h | | | | | | | | | |
| SUB 72h,1 | 0 | | | | | | | | | |

# 6. Sa se completeze tabelul urmator:

| | | ACC | DP | ARP | T | P | AR1 | AR2 | PM | 100h | 200h | 201h | 284h |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Valori initiale | | | | | 10h | | | 1 | 10h | 2 | 4 | 100h |
| | | | | | | | | | | | | | |
| 1 | LDPK 5 | | | | | | | | | | | | |
| 2 | LRLK AR1, 200h | | | | | | | | | | | | |
| 3 | LRLK AR2, 100h | | | | | | | | | | | | |
| 4 | LARP 1 | | | | | | | | | | | | |
| 5 | ZAC | | | | | | | | | | | | |
| 6 | ADD 4, 2 | | | | | | | | | | | | |
| 7 | LT *+ | | | | | | | | | | | | |
| 8 | MPYA *, 2 | | | | | | | | | | | | |
| 9 | XORK 80h | | | | | | | | | | | | |
| 10 | SUB *, 1, 1 | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |

*8. MPYA* – inmulteste si acumuleaza produsul precedent deplasat conform bitilor PM

```
MPYA *,2        ; (P) = (T) x (AR1), deplasat stanga cu 1 bit (PM=1)
                ; ARP=2
                ; (ACC) = (ACC) + (P)anterior
```

*9.* 0000 0100 0010 0000 b **xor**
0000 0000 1000 0000 b
0000 0100 1010 0000 b → 4A0h

Raspuns

| | | ACC | DP | ARP | T | P | AR1 | AR2 | PM | 100h | 200h | 201h | 284h |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Valori initiale | | | | | 10h | | | 1 | 10h | 2 | 4 | 100h |
| | | | | | | | | | | | | | |
| 1 | LDPK 5 | | 5 | | | | | | | | | | |
| 2 | LRLK AR1, 200h | | | | | | 200h | | | | | | |
| 3 | LRLK AR2, 100h | | | | | | | 100h | | | | | |
| 4 | LARP 1 | | | 1 | | | | | | | | | |
| 5 | ZAC | 0 | | | | | | | | | | | |
| 6 | ADD 4, 2 | 400h | | | | | | | | | | | |
| 7 | LT *+ | | | | 2 | | 201h | | | | | | |
| 8 | MPYA *, 2 | 420h | | 2 | | 8 | | | | | | | |
| 9 | XORK 80h | 4A0h | | | | | | | | | | | |
| 10 | SUB *, 1, 1 | 480h | | 1 | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |