Programare in limbaj de asamblare

Instructiuni MMX

Tehnologia MMX

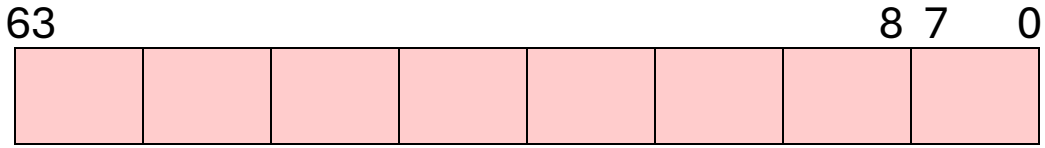
- MMX – Multi-Media eXtension
- Obiectiv:
  - cresterea vitezei de prelucrare a informatiilor audio, video, grafica, etc.
  - Achizitia, prelucrarea si generarea **in timp-real** a informatiilor multimedia
  - Dezvoltarea de aplicatii multimedia la care timpul de executie (reactie) este critic
- Componente MMX:
  - Registre MMX (8), MM0 .. MM7; 64biti/registru
  - Instructiuni MMX (57)
  - Tipuri de date impachetate
  - Unitate de executie MMX

Tehnologia MMX – caracteristici generale

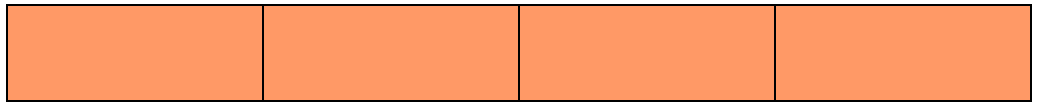
- Ideea de baza:
  - Executia in paralel a unei instructiuni pe un set de date
  - Paralelism de tip SIMD – Single Instruction Multiple Data (simulare)
- Tehnologia MMX este eficienta daca:
  - Datele au o structura de tip vector, matrice, sau o alta structura cu grad ridicat de regularitate
  - Aceeasi secventa de operatii se aplica in mod identic pe fiecare element al structurii
  - Aceeasi secventa de prelucrare se executa de un numar foarte mare de ori (ex: pe un milion de pixeli ai unei imagini)
  - Timpul de executie este critic

Tipuri de date MMX

Octeti impachetati (Packed byte): 8 octeti, 64 biti

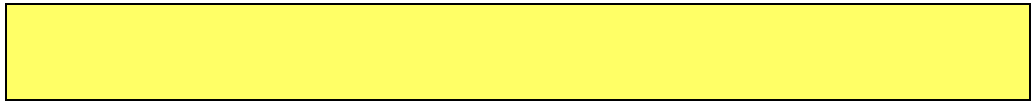


Cuvinte impachetate (Packed word): 4 cuvinte, 64 biti



Dublu-cuvinte impachetate (Packed dword): 2 dcuvinte, 64 biti

Cuadruplu-cuvant (Quadword): 1 qcuvant, 64 biti



Sintaxa instructiunilor MMX

- <instructiune>:= <mnemonica> [<destinatie>,<sursa>]
- <mnemonica>:=P<operatie><sufixe>
- <sufixe>:= [US|S|SS][B|W|D|Q][B|W|D|Q]
- Sufixele indica tipul datelor si modul de efectuare a a operatiilor aritmetice:
  - US – unsigned, saturation
  - S, SS – signed, saturation (in lipsa – wrapearound)
  - B, W, D, sau Q – byte, word, dword, qword; daca apar 2 litere atunci sursa este de tipul primei litere iar destinatia de tipul celei de a doua litere
- Exemplu:

```
paddusw MM4, mem1
psubusb MM4, mem1
```

Operatii aritmetice cu Saturare si cu Intoarcere (wraparound)

- Tehnici de solutionare a depasirilor de capacitate (superioara sau inferioara):
  - Prin “intoarcere”
  - Prin saturare

Exemplul 1: operatii cu numere fara semn (octeti):

Intoarcere:	Saturare:
F0h+20h = 10h	F0h+20h = FFh
80h -A0h = 20h	80h -A0h=00h

Exemplul 2: operatii cu numere cu semn (octeti):

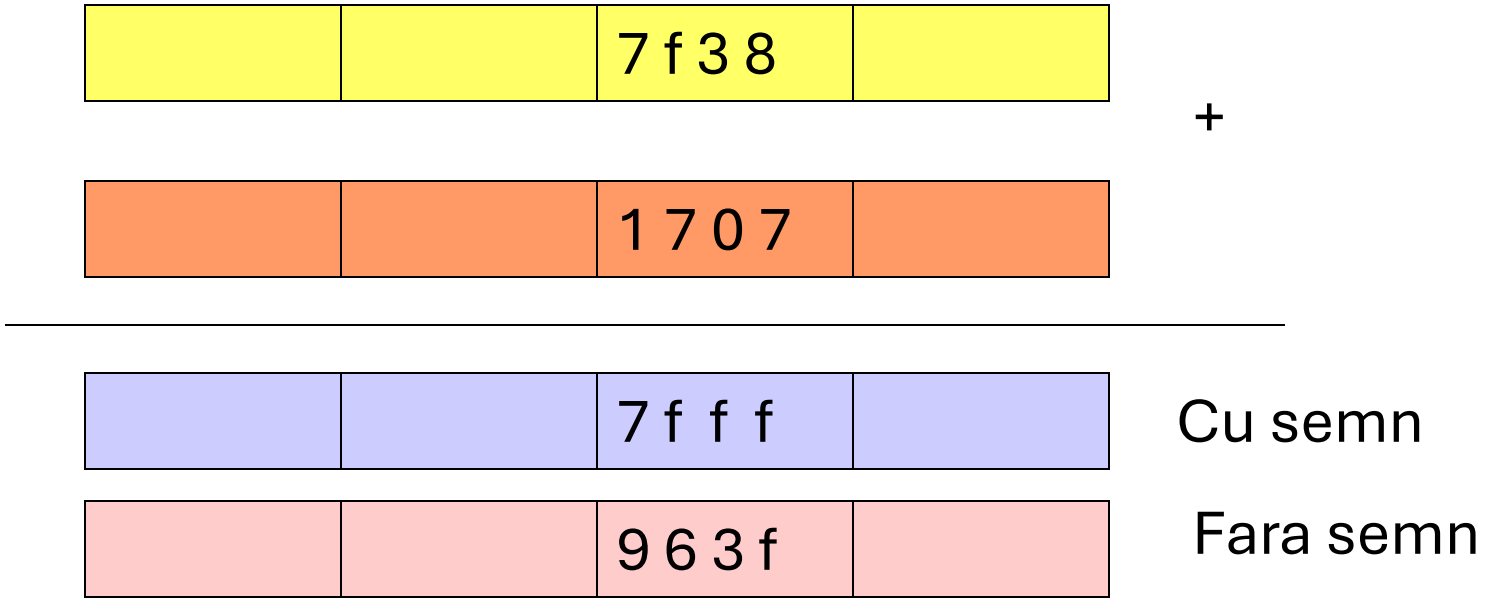
127+1= - 127	127+1= 127
- 128 – 1 = 127	- 128 – 1 = -128

Operatii aritmetice cu saturare

- Exemplul 3: `paddsw`

  7f38+1707 = 7fff - saturare
- Exemplul 4: `paddusw`

  7f38+1707 = 963f - nu este saturare



Instructiunile MMX

- EMMS
- Adunare si scadere
- Instructiuni de deplasare (shift)
- Instructiuni logice
- Instructiuni de multiplicare
- Instructiuni de comparare
- Instructiuni de impachetare/despachetare
- Instructiuni de transfer

Instructiuni MMX

- Instructiunea EMMS
  - Se foloseste dupa o secventa de instructiuni MMX si inainte de o instructiune in virgula flotanta pentru a evita generarea unei exceptii sau a unui rezultat incorect pt. operatii in virgula flotanta
  - Problema:
    - starea modulului MMX este o copie a starii unitatii flotante
    - dupa instructiuni MMX registrul tag al coprocesorului este alterat; poate genera o exceptie de tip stiva invalida
  - EMMS sterge starea MMX si astfel valideaza registrul tag (toate intrarile in stiva flotanta sunt validate)

Instructiuni de adunare si scadere

Instructiune	b	w	d	q
padd	X	X	X	
padds	X	X		
paddus	X	X		
psub	X	X	X	
psubs	X	X		
psubus	X	X		

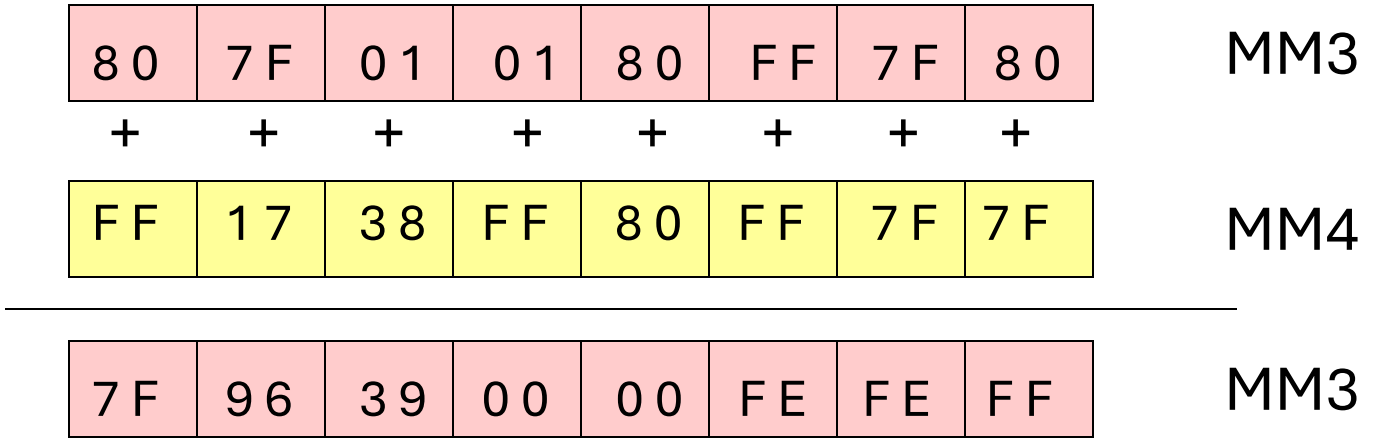
paddxx <destinatie>, <sursa>

<destinatie>:= MMi

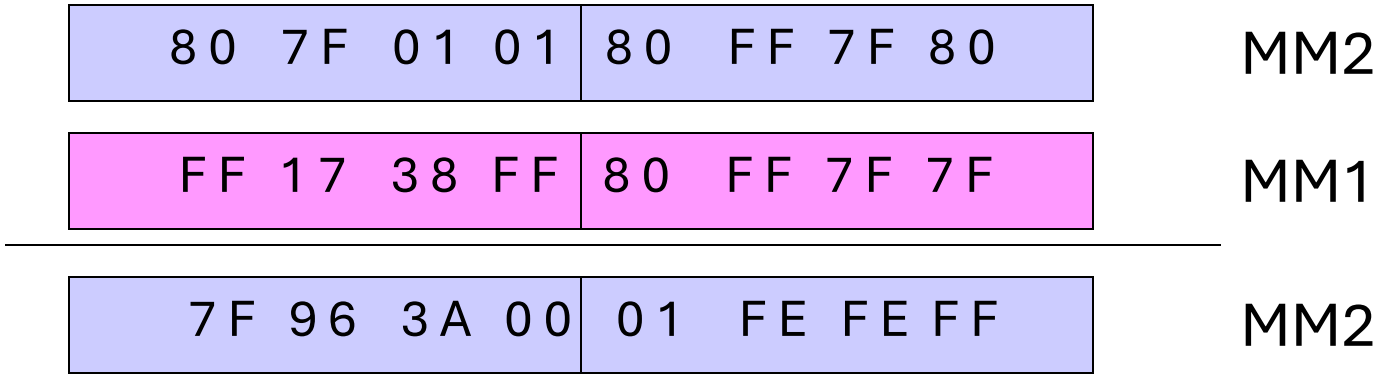
<sursa>:=MMi | <variabila>

Adunare si scadere - exemple

paddb MM3,MM4



paddd MM2, MM1



Instructiuni de deplasare (shift)

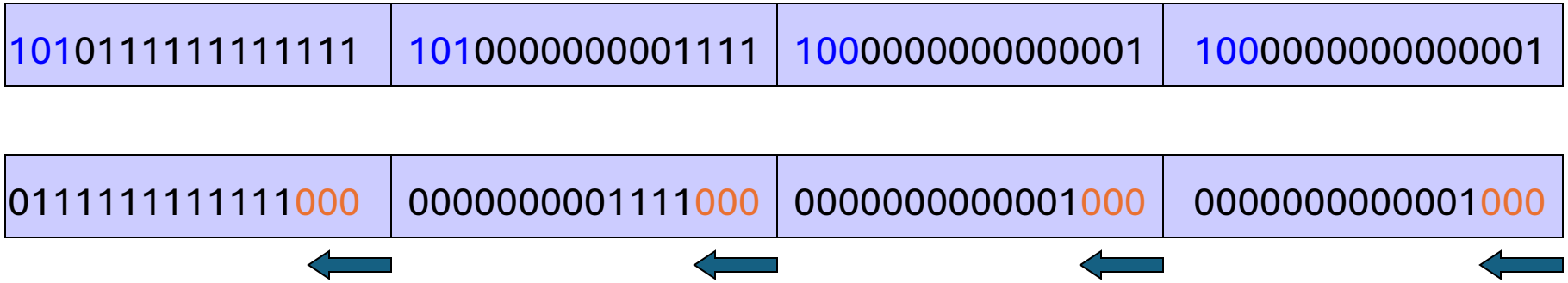
Instructiune	b	w	d	q	
psll		X	X	X	Shift left logic
psra		X	X		Shift right arithmetic
psrl		X	X	X	Shift right logic

psllx | psrax | psrl <destinatie>, <numar>

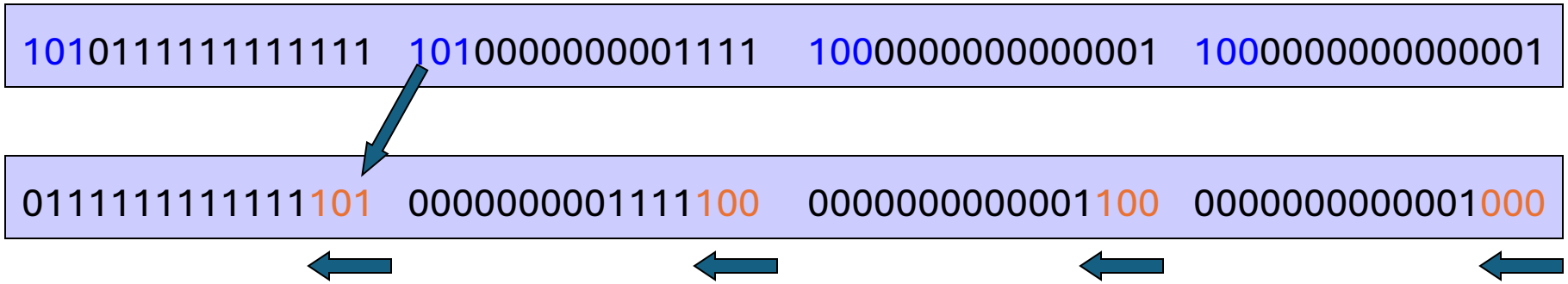
Elementele destinatiei se deplaseaza la stanga sau la dreapta cu un <numar> de pozitii binare

Instructiuni de deplasare - exemple

psllw MM4, 3



psllq MM4, 3



Instructiuni logice

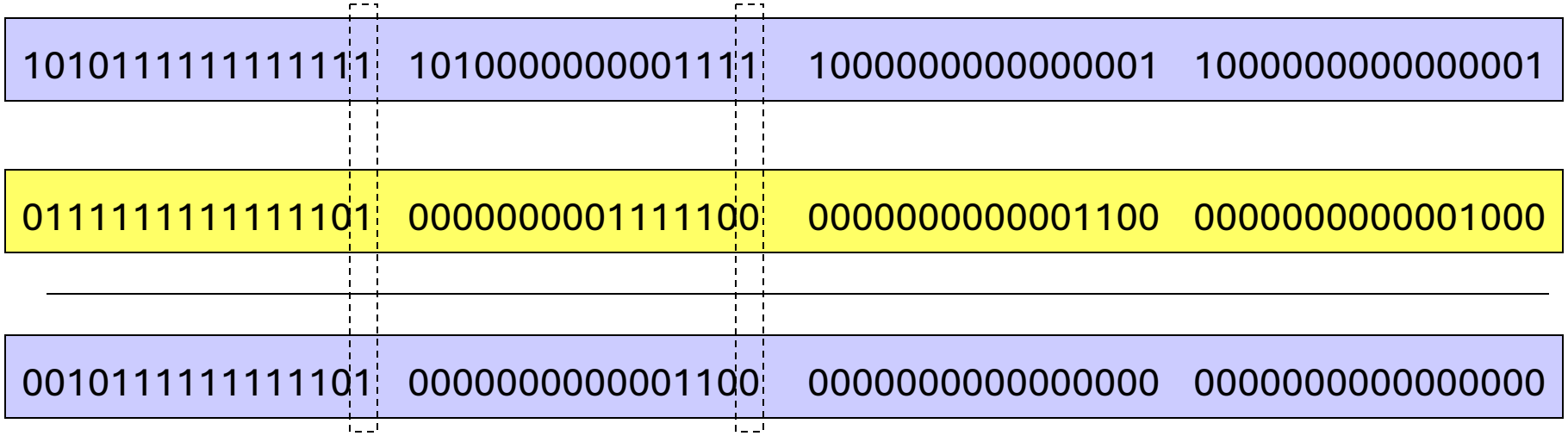
Instructiune	64 biti	
pand	X	X=X AND Y
pandn	X	X=(NOT X) AND Y
por	X	X=X OR Y
pxor	X	X=X XOR Y

pand | pandn | por | pxor <destinatie>, <sursa>

Operatia logica se executa bit cu bit

Instructiuni logice - exemplu

pand MM4, MM3



Instructiuni de multiplicare

Instructiuni	b	w	d	q
pmadd		→	→	
pmulh		X		
pmull		X		

multiply and add

multiply and keep high

multiply and keep low

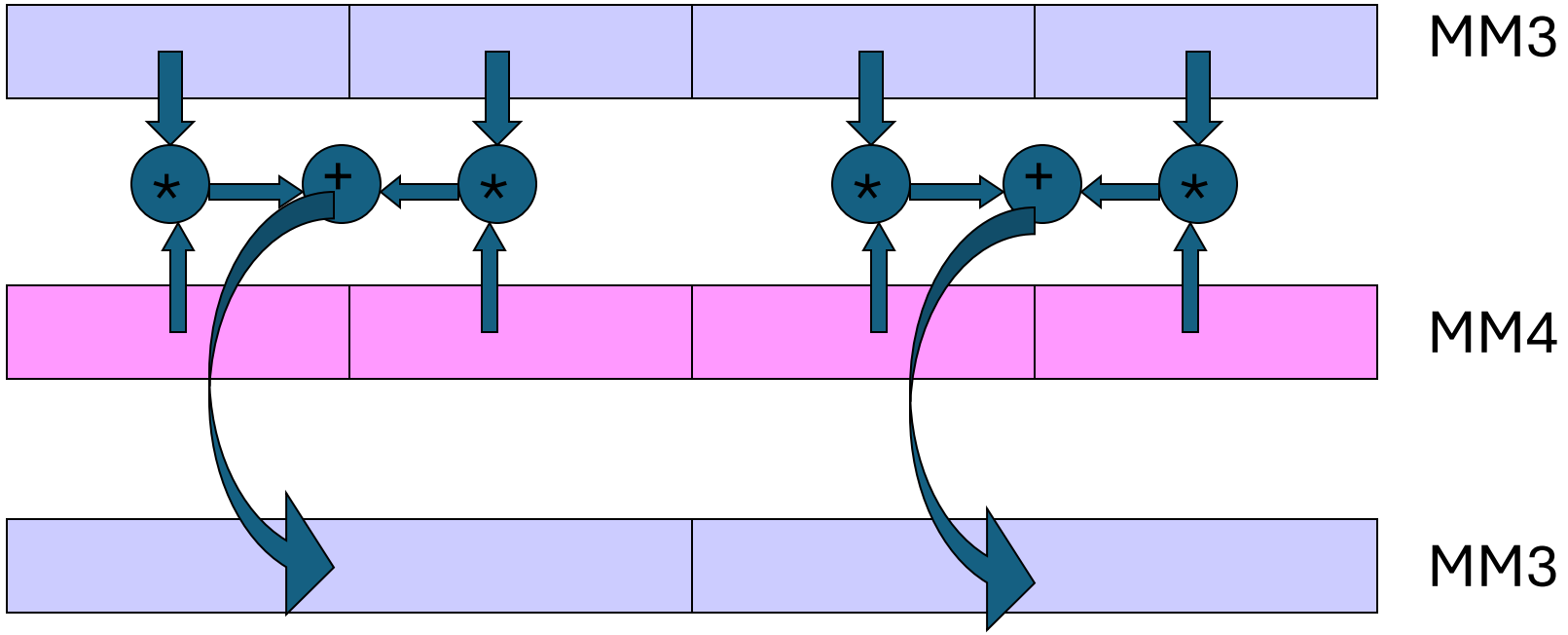
pmadd – multiply and add

$X_{31-0} = X_{15-0} * Y_{15-0} + X_{31-16} * Y_{31-16}$

$X_{63-32} = X_{47-32} * Y_{47-32} + X_{63-48} * Y_{63-48}$

Instructiuni de multiplicare

pmadd MM3,MM4



Instructiuni de multiplicare

pmulh MM3,MM4

7 1 C 7	8 D 9 4	8 0 0 0	7 1 C 7	MM3
8 0 0 0	C F C B	F F F F	0 4 0 0	MM4

C71C8000 158BF05C 00008000 01C71C00

C 7 1 C	1 5 8 B	0 0 0 0	0 1 C 7	MM3

pmull MM3,MM4

8 0 0 0	F 0 5 C	8 0 0 0	1 C 0 0	MM3

Instructiuni de comparare

Instructiuni	b	w	d	q	
pcmpeq	X	X	X		compare equal
pcmpgt	X	X	X		compare greater (intregi cu semn)

Exemplu: pcmpeqw MM3,MM4

0 0 0 0	1 2 3 4	F F F F	0 0 C B	MM3
0 5 6 0	1 2 3 4	1 3 F F	0 0 C B	MM4
0 0 0 0	F F F F	0 0 0 0	F F F F	MM3

Instructiuni de impachetare/despachetare

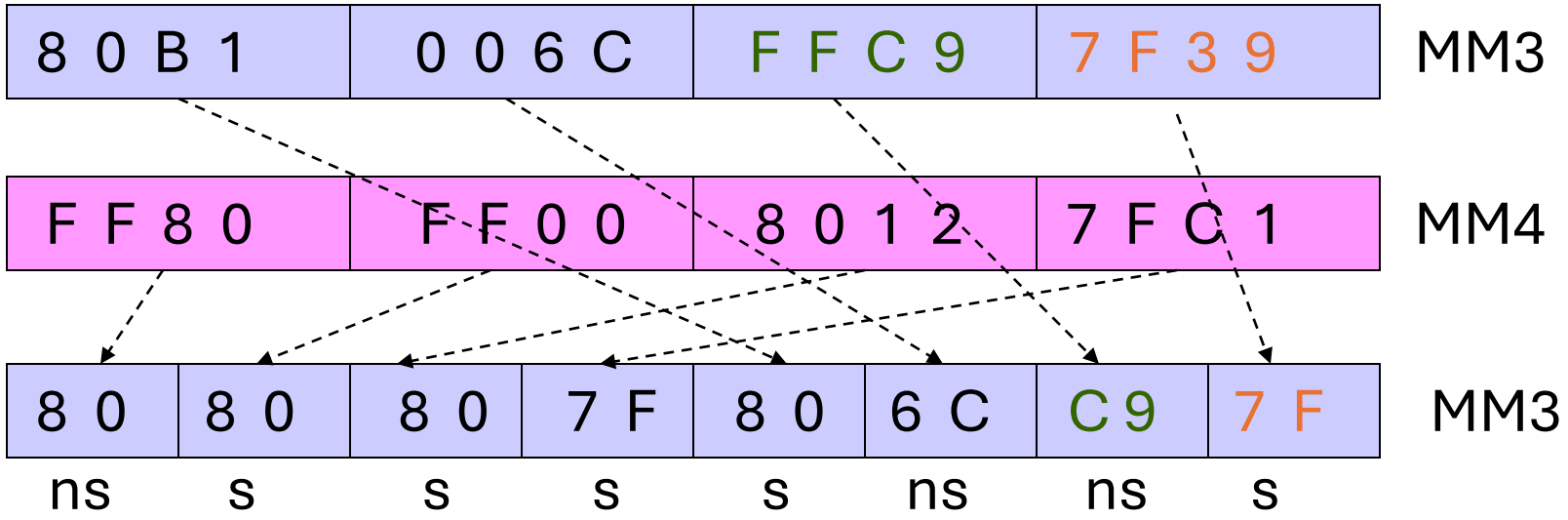
Instructiuni	b	w	d	q	
packss	←	←	←		pack signed, saturation
packus	←	←			pack signed to unsigned
punpckh	→	→	→	→	unpack high
punpckl	→	→	→	→	unpack low

Impachetare: copierea unei structuri mai mari (ex: word) intr-o structura mai mica (ex: byte) cu saturare in caz de nevoie

Despachetare: interclasarea elementelor celor doi operanzi – se foloseste pentru extinderea reprezentarii

Instructiuni de impachetare/despachetare
Exemplu de impachetare

packss MM3,MM4



ns – nesaturat

s - saturat

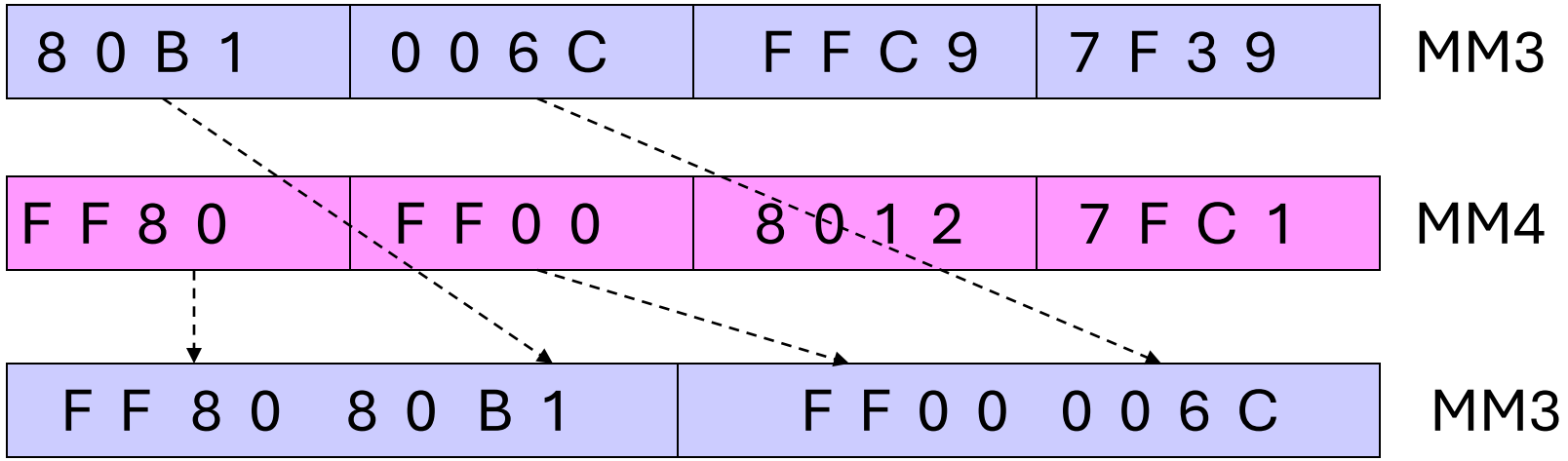
7F39 = 32569 > 7F00 = 32512 => sat.

FFC9 = - 37 > 8000 = -32512 => nsat.

1111.1111.1100.1001 => 1100.1001

Instructiuni de impachetare/despachetare
Exemplu de despachetare

punpckhwd MM3,MM4 ; cuvant ->dublu-cuvant



Instructiuni de transfer

Instructiune	32biti	64 biti
movd	X	
movq		X

move dword (32 biti)

move qwordword (64 biti)

movd | movq <destinatie>, <sursa>

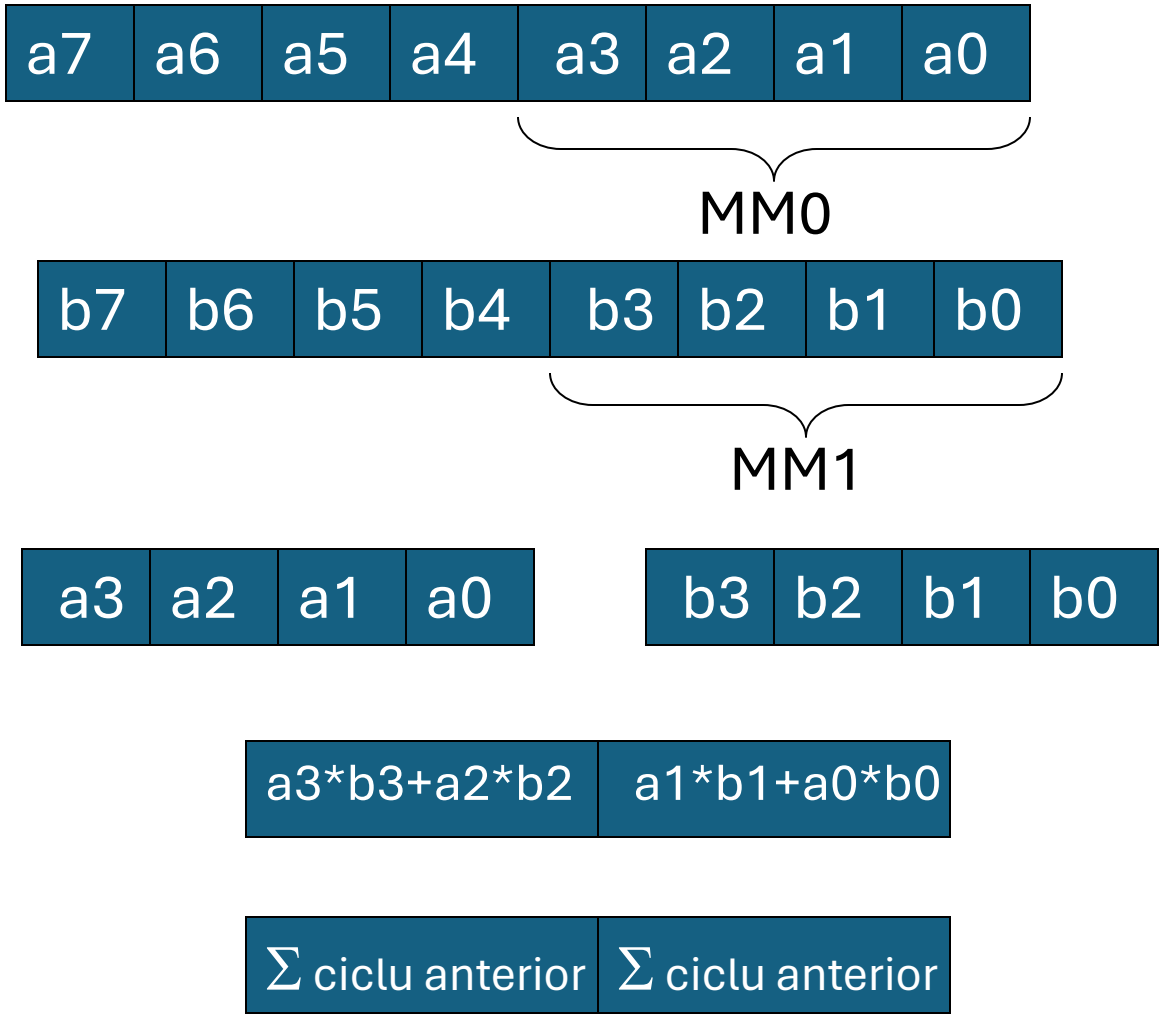
<destinatie> := MMi | <variabila_mem> |<registru_x86>

<sursa> := MMi | <variabila_mem> |<registru_x86>

exemple:

```
movd MM2, var_32     ; se extinde cu 0
movd var_32, MM3     ; se copiaza partea low
movq var_64, MM5
```

Exemple de proceduri MMX
Produsul scalar a doi vectori: Σ a(i)*b(i)




```
et:   movq  MM0, [adr_v_a]
      movq  MM1, [adr_v_b]
      pmaddwd  MM0, MM1
      paddd   MM2, MM0
      add  adr_v_a, 8
      add adr_v_b, 8
      sub contor
      jnz et
      movq  MM0, MM2
      psrlq   MM2, 32
      paddd   MM2, MM0
      movd    rez, MM2
```

Exemple de programe MMX
Suprapunerea selectiva a imaginilor

1. pcmpeqb MM1, MM3
2. pand MM4, MM1
3. pandn MM1, MM3
4. por MM4, MM1