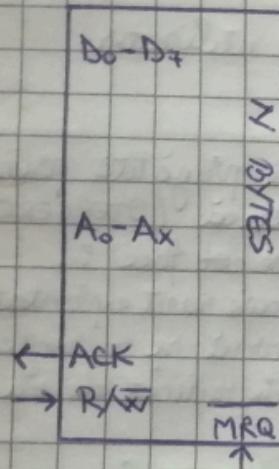


Reprezentarea bloc a unei memorii



Memorie : 1 octet
($N = 1$)

- $D_0 \dots D_7$ = borne de date
- R/W = borne de citire - scriere
- MRQ = borne de selectie
- $A_0 \dots A_x$ = borne de adresa ($x = \log_2 N - 1$)
- ACK = borne de confirmare

CAPSULA 0: 0000
3 FFF

CAPSULA 1: 4000
4 FFF

CAPSULA 2: 8000
B FFF

CAPSULA 3: C000
FFF

RAM

- volatilă?
- Random Access Memory
- citire + scriere
- păstrează informațile doar câtă vreme este alimentată

ROM

- nesivolatilă?
- Read Only Memory
- citire
- păstrează informațile și când nu este alimentată

ROM propriu-zisă

- înscrivă prin însăși procesul de fabricație
- înscrivă de utilizator

PROM

- odată înscrivă, nu mai poate fi modificată, ci doar citită

PROGRAMMABLE

EPROM

- înscrivă de utilizator

EEPROM

- înscrivă de utilizator

- înainte, este pregătită prin iradiere cu raze ultraviolete (efect iradiere: stergerea memoriei, adică aducerea tuturor celulelor la „0”)
- poate fi steară și reprogramată de un nr. foarte ridicat de ori;

ERASABLE

ELECTRICAL

- înainte, este pregătită prin stergere electrică (STERGEREA + ÎNSCRIREA se pot face în circuitul în care sunt exploatate).
- poate fi steară și reprogramată de un nr. foarte ridicat de ori;

- Sumator elementar serial (= Sumator serial cu bistabile D)

$$z_i = x_i \oplus y_i \oplus c_{i-1}$$

$$c_i = x_i \cdot y_i + x_i \cdot c_{i-1} + y_i \cdot c_{i-1}$$

- Sumator serial cu bistabile J-K

$$Q_{m+1} = J \cdot \overline{Q_m} + \overline{K} \cdot Q_m \rightarrow \text{ecuația bistabilului J-K}$$

$$J = x \cdot y$$

$$K = \overline{x+y} = \overline{x} \cdot \overline{y}$$

$$z = x \oplus y \oplus Q$$

- Avantaje: - simplitate

- implementare printr-un nr. redus de circuite

} SERIAL

- Dezavantaje: - viteză de însumare redusă

- Sumator paralel „carry lookahead”

$$c_i = \underbrace{x_i y_i}_{g_i} + \underbrace{(x_i + y_i) \cdot p_{i+1}}_{p_i} \cdot c_{i+1}$$

$$\nearrow c_{i+1} = g_{i+1} + p_{i+1} \cdot g_i + p_{i+1} \cdot p_i \cdot c_{i+1}$$

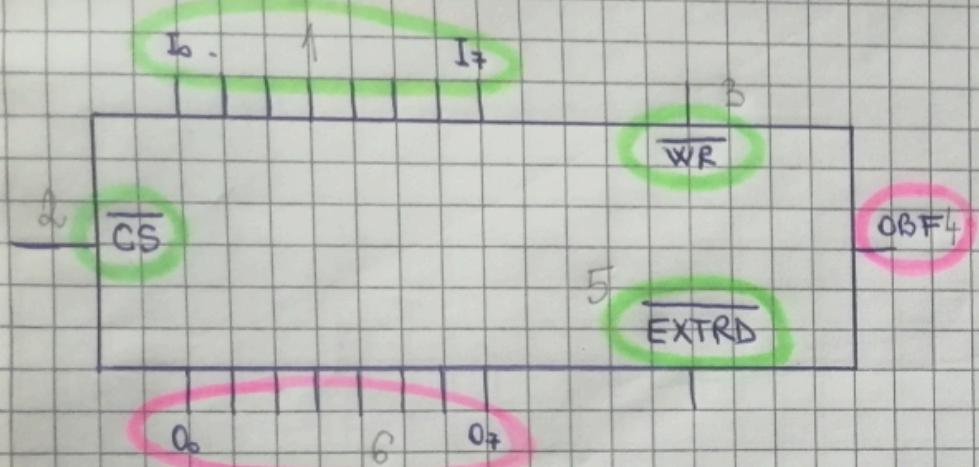
$$T_{ad} = m \cdot (2\tilde{\tau}) + \tilde{\tau} = (2m+1) \cdot \tilde{\tau}$$

$$\rightarrow c_i = g_i + p_i \cdot g_{i-1} + p_i \cdot p_{i-1} \cdot c_{i-2}$$

$$\bullet l_{c-mm-\Theta} = \log_2 \left(\frac{2M}{n} \right) = \log_2 2M - \log_2 n$$

$$l_{c-mm-p} = \log_2 \left(\frac{2M^2}{n} \right) = 2 \log_2 2M - \log_2 n = 2 \log_2 M + 1 - \log_2 n$$

Port de ieșire

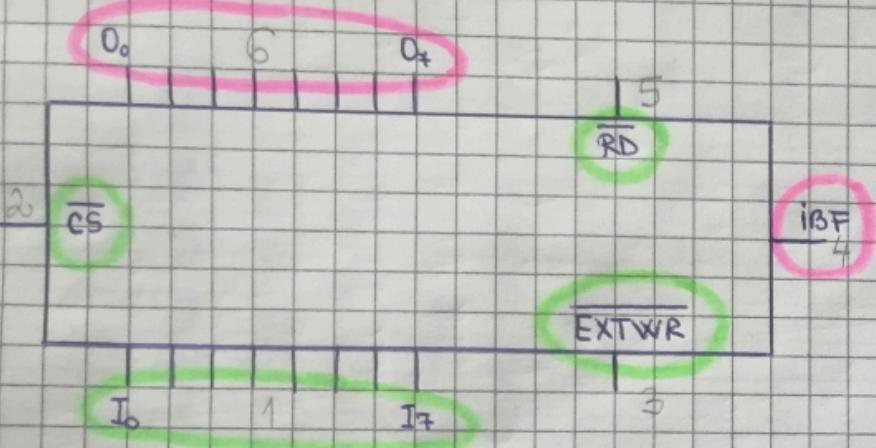


$\overline{WR} \rightarrow$ activ când $OBF = 1$

$\overline{EXTRD} \rightarrow$ activ când $OBF = 0$

- $I_0 \dots I_7$ = borne de intrare date (preiau datele de la precessor)
- \overline{WR} = borne de ieșire (borne de intrare de comandă a scrierii în port a datelor de pe limitele $I_0 \dots I_7$)
- $D_0 \dots D_7$ = borne de ieșire date (transmit datele spre echipamentul periferic).
- \overline{EXTRD} = borne de citire (borne de intrare de informare a portului asupra faptului că echipamentul periferic a preluat datele inscrise în el).
- \overline{OBF} = borne de remnalare (borne de ieșire de remnalare a portului că în port există o dată înscrise și încă necitită)
- $\overline{CE} / \overline{CS}$ = borne de selecție (borne de intrare de validare/invalidare).

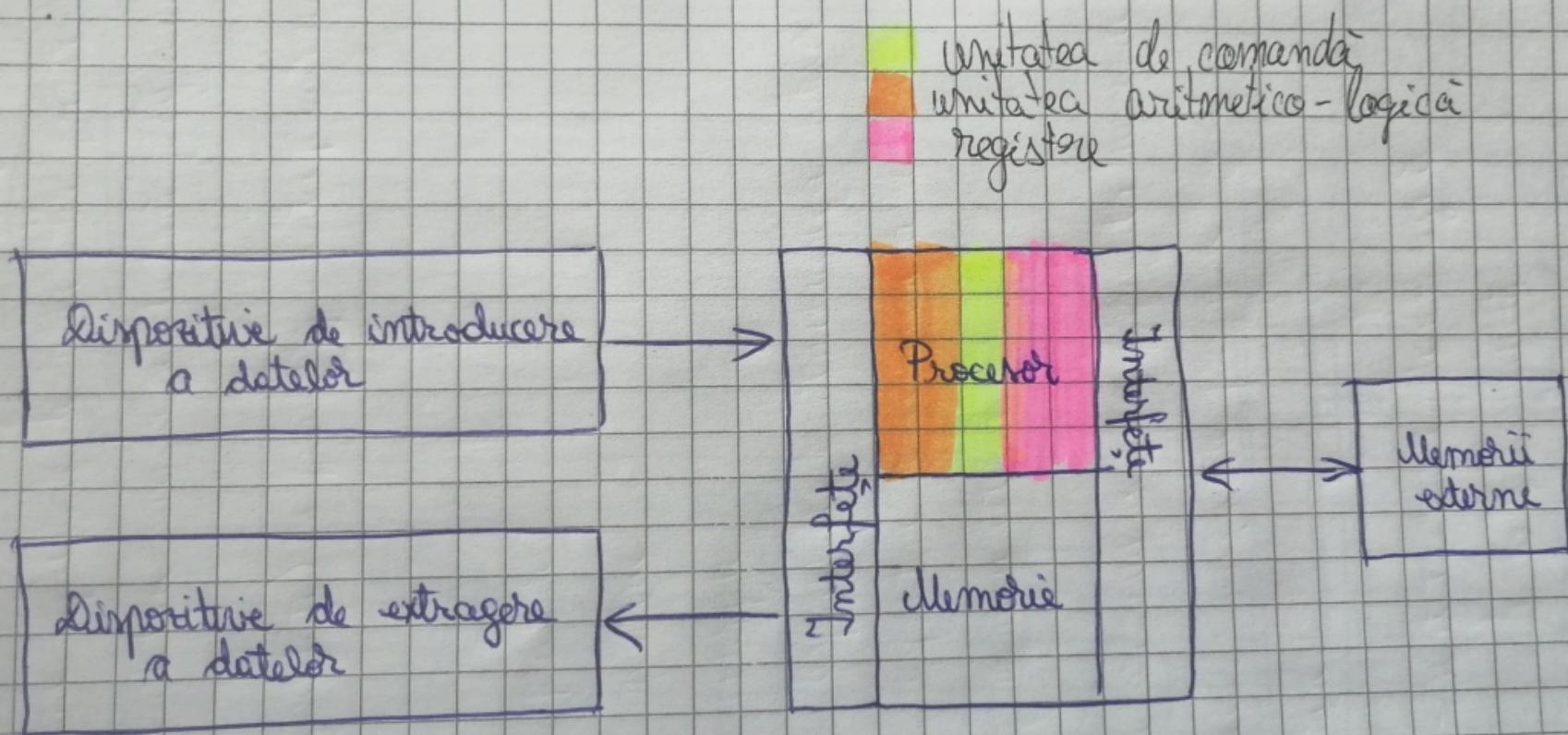
Port de intrare



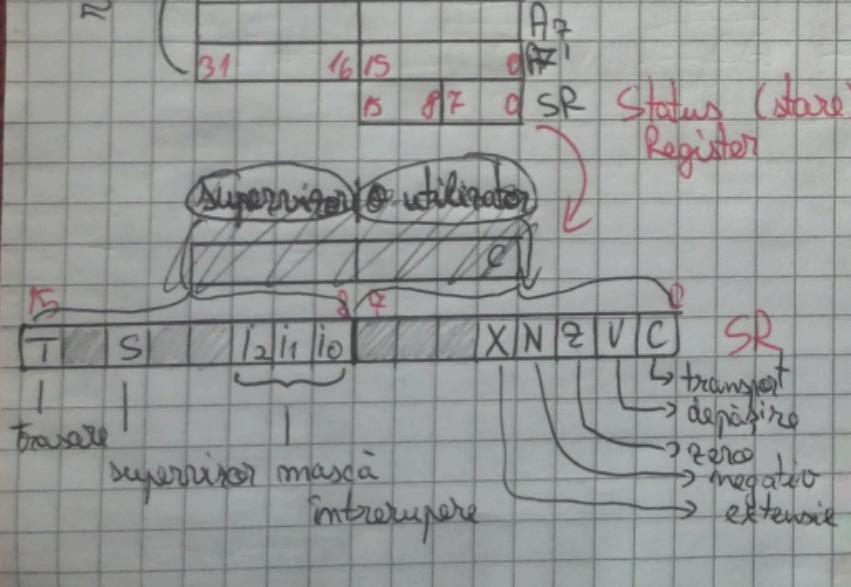
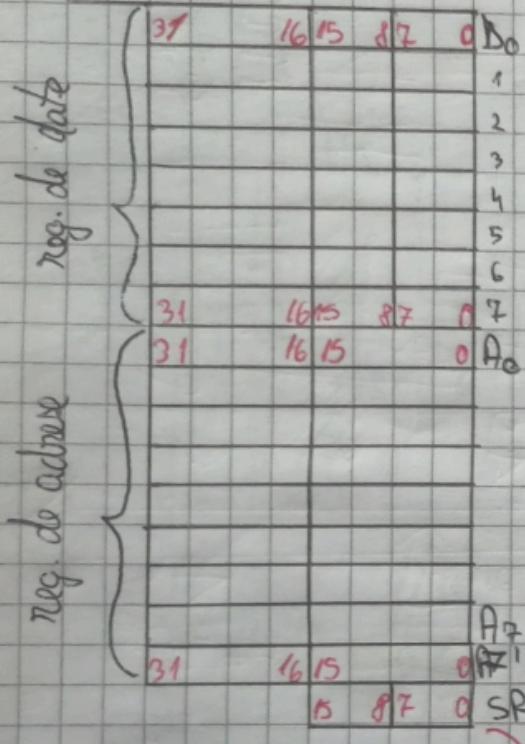
$\overline{RD} \rightarrow$ se activăază doar dacă $IBF = 0$

$\overline{EXTWR} \rightarrow$ se activăază doar când $IBF = 1$

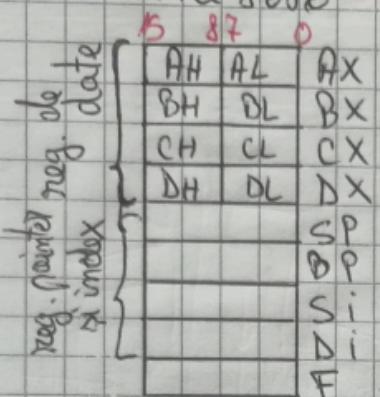
- $I_0 \dots I_7$ = borne de intrare date (preiau datele de la echipamentul periferic).
- \overline{EXTWR} = bornă de scrisoare (borne de intrare de comandă a scrierii în port a datelor de pe liniile $I_0 \dots I_7$)
- $D_0 \dots D_7$ = borne de ieșire (transmit datele spre procesor)
- \overline{RD} = bornă de citire (borne de intrare de comandă a disponibilității către procesor a datelor înscrise în port)
- IBF = bornă de remnalare (borne de ieșire de remnalare a faptului că în port există o dată înscrisă și încă necitită)
- $\overline{CE} / \overline{CS}$ = borne de selecție (borne de intrare de validare / invalidare)



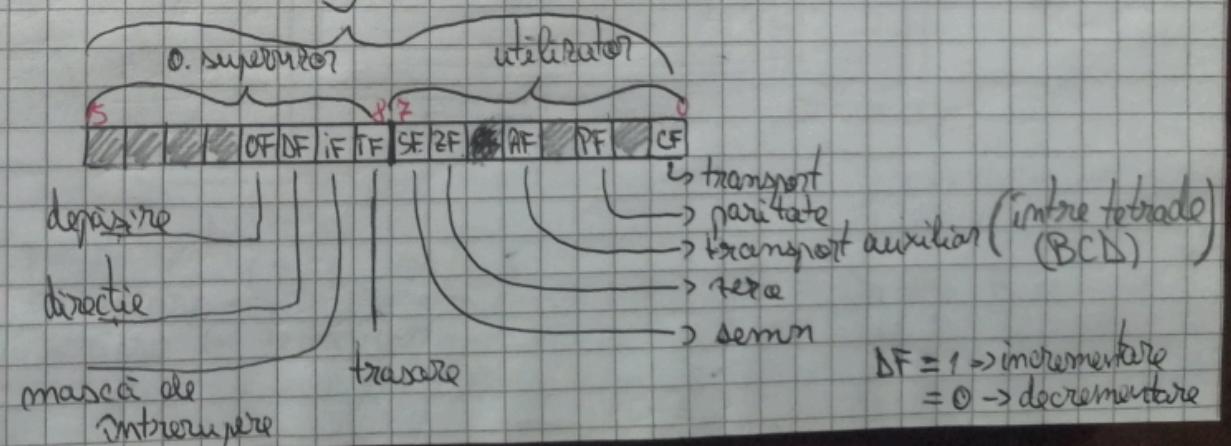
Motorola 68000



Intel 8086



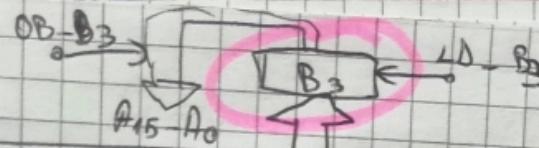
- ABC (CSAC) / Po-D₇ (M68000): Prezintă operațiuni preluare date, registru de manevră
- implementarea unor "moduri de adresare" bazată indirectă
- registru-murmurător; operații de rotire, deplasare, ciclare și transfer de sursă
- op. înmulțire, împărțire, lucru cu porturi
- stack pointer (pointer de stivă)
- punere programată (nu LIFO)
- registre cu index → calcul adresa
- Surse / destinație
- registru de punere → PSW, Program Status Word



DF = 1 → incrementare
= 0 → decrementare

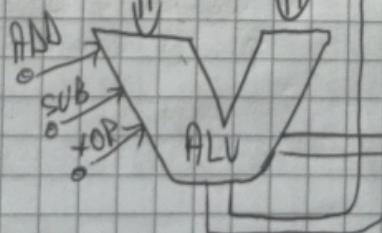
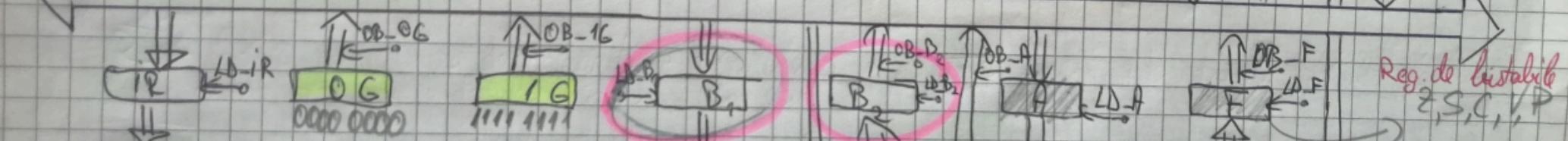
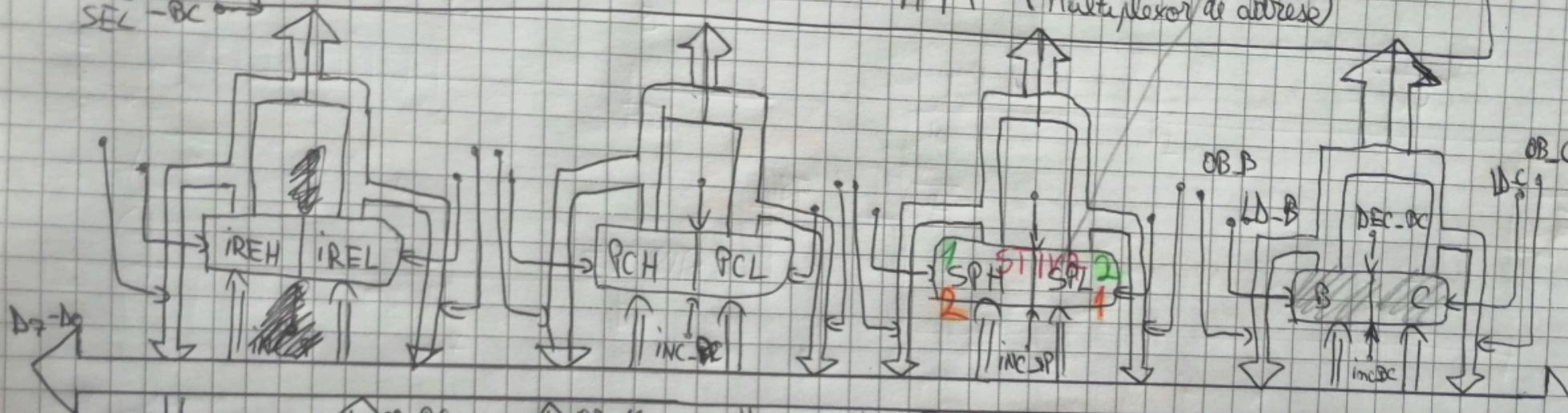
CSAC 2001

SELIRE
SELPC
SELSP
SELBC



Stack pointer
(ativa, LIFO)

AM (Multiplexor de adresa)



pseudoregister
auxiliare (B_1, B_3 - Buffer)
scriero (+ documentare)
citire (+ incrementare)

OB = On the Bus
→ activarea regizorilor

Cifra	BCD	E3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	1011
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

- Pentru numere pozitive, numărul este același în SM, C_1 , C_2 .
- Pentru numere negative:
 - SM \rightarrow împărtim succint la 2 și îl formăm cu resturile
 - C_1 \rightarrow se meagă SM (?! fără bitul de semn).
 - C_2 \rightarrow se adună 1 la C_1
- Adunarea în C_1 : se adună 1 la final (la rezultat) dacă:
 - ambele numere sunt negative
 - primul număr este pozitiv, al doilea este negativ
- Adunarea în C_2 : dacă apare Carry, se ignorează
- Adunarea în BCD:
 - se efectuează cifră cu cifră, de la dreapta spre stânga
 - dacă se obține Carry, atunci la cifra respectivă adaugă 6 (0110).
 - dacă NU se obține Carry, ci se obține o valoare între 0 și 9, atunci nu se adaugă 6
 - dacă NU se obține Carry, ci se obține o valoare între 10 și 15, (care nu este în tabel), atunci se adaugă din nou 6 (0110).

Adunarea în E3:

- adunăm dintr-un loc toate cifrele, iar dacă apare un Carry,
se adună încă un 1 pe rangul următor
- după asta, la fiecare cifră aplicăm o corecție simplă:
dacă are Carry (dacă e sub 9 în tabel: ↑) adunăm 3 (0011),
dacă nu are Carry (dacă e peste 9 în tabel: ↓) adunăm 13 (1101).

BCD ○ fiecare cifră ocupă 4 biți

- adunarea se efectuează cifră cu cifră (nu lit cu lit)
- numărul este pozitiv dacă cifra este între 0 și 4
negativ dacă cifra este între 5 și 9

E3 ○ NU constituie un cod ponderat