TEMA NR.7:
ÎNDEPLINIREA OPERAŢILOR DE ADUNARE ŞI SCĂDERE ÎN CODURILE DIRECT, INVERS ŞI COMPLEMENTAR

1 Operaţiile de adunare şi scădere în codurile direct, invers şi complimentar.

2 Depistarea depăşirii la adunarea şi scăderea numerelor binare.

3 Deplasarea numerelor binare

1. Operaţia de adunare este una din cele mai importante operaţii în orice sistem.

În sistemul binar această operaţie se efectuează după următoare regulă:

0+0=0 0+1=1 1+0=1 1+1=0(1)

Notă: Suma 1+1 =2 pentru sistemul de numerație zecimal în binar 2 →10 deci 1+1=10 care nu poate fi reprezentat pe un bit. Respectiv, la adunare acestor numere avem: 1+1=0 plus 1 transport în rangul vecin mai semnificativ pe care îl luăm în considerației la adunare în acest rang.

Operaţiile de adunare în cod direct se îndeplinesc prin metoda clasică adică cifrele din aceeaşi poziţie a două numere se adună între ele şi-n cazul când suma lor ≥ cu baza sistemului de numeraţie în poziţia cifrelor respective se scrie diferenţa dintre suma obţinută şi baza sistemului de numeraţie iar în poziţia mai puţin semnificativă se transmite unitatea de transport care se adună la cifrele din poziţia respectivă.

La scădere în cazul când descăzutului e mai < decât cifra scăzătorului din cifra mai semnificativă a descăzutului se va lua împrumut.

0-0=0, 1-0=1, 1-1=0, 0-1=1

După cum sa menţionat codul direct nu se foloseste pentru operaţiile adunare, dar se utilizează la înmulțirea numerelor binare.

Adunarea a 2 nr. binare în cod invers se efectuiază dupa regula de adunare a nr-lor în cod direct, cu excepția:

Dacă din rangul semnului se genereaza cifra de transport, ea se aduna în rangul cel mai puțin semnificativ.

Dacă rezultatul obtinut este negativ, în rangul semnului v-om avea 1, dacă pozitiv atunci în rangul semnului va fi 0.

Operaţia de scădere propriu zisă nu se realizează în dispozitivele numerice ea este înlocuită cu operaţia de adunare:

A-B=A+(-Bci)

Înainte de a efectua această operaţie valoarea scăzătorului se inversează conform regulei codului respectiv.

Exemplu: A=19 B=12

A+B

A=0.10011 B=0.01100

0.10011

0.01100

0.11111 = 31

A=14 B=9

A-B=A+(-Bci)

A= 0.1110 B=0.1001

-Bci=1.0110

0.1110

1.0110

0.0100

1

0.0101 =5

A=14 B=9

A= 0.1110 B=0.1001

B-A=B+(-Aci)

Aci=1.0001

0.1001

1.0001

1.1010 Zci=-0101=-5

A=4 B=-9 (1001)

A= 0.0100 Bci=1.0110

B-A=B+(-Aci) Aci=1.1011

1.0110

1.1011

1.0001

1

1.0010 (-1101=-13)

COD COMPLIMENTAR.

- Adunarea şi scăderea numerelor în cod complementar are loc în mare parte conform aceleaşi reguli ca adunarea şi scăderea în cod invers cu următoarea excepţia:

În cazul când din bitul semnului are loc cifra de transport ea se ignoră (mai departe nu se foloseşte).

Ca şi în cazul codului invers operaţiile de scădere este înlocuit prin operaţia de adunare a descăzutului cu scăzătorul al cărui semn se inversează (din + în – şi invers).

EXEMPLU:

A= 0110100 ($52_{(D)}$) B= 1000111($71_{(D)}$).

A = 0.0110100

B = 0.1000111 +

A+B= 0.1111011=(A+B)

Rezultatul $0.1111011_{(B)}=123_{(D)}$.

EXEMPLU:

$A= -76_{(D)}$ (.1001100) $B=127_{(D)}$

Acd=1.1001100 Acc=1.0110100 B=0.1111111

Acc= 1.0110100

Bcc= 0.1111111

A+B = 1 ← 0.0110011 51

A= 47 B= -75

A+B

A=0.0101111

Bcd=1.1001011=Bcc=1.0110101

0.0101111

1.0110101

1.1100100

- 0011011

1cc

- 0011100 = -28

A= - 47 (0101111) B=75

A-B=A+(-Bcc)

Acc=1.1010001 B=0.1001011

Bcc=1.0110101

1.1010001

1.0110101

1.0000110 (la inversare +1)

- 1111010 = -122

2. În cazul adunarii a 2 numere cu acelaşi semn sau la scăderea numerelor cu semne diferite se poate întâmpla ca rezultatul obţinut să nu să se încadreze în numărul de cifre rezervate în grila calculatorului.

Acest fenomen se numeşte fenomen de depăşire a grilei calculatorului şi în fiecare caz când se întâmplă acest lucru depăşirea trebuie semnalizată.

Exemplu:

1) A= +1011000 ($88_{(10)}$) B= +0110010 ($50_{(10)}$)

A_{CC}= 0.1011000

B_{CC}= 0.0110010 +

$\underline{1}.0001010 \neq (A+B)_{CC}$

**depăsire pozitivă*

1) A= -0111100 ($-60_{(10)}$) и B= -1000101 ($-69_{(10)}$)

A_{CC}= 1.1000100

B_{CC}= 1.0111011 +

$A_{CC}+B_{CC}= 1\underline{0}.1111111 \neq (A+B)_{CC}$

**depasire negativa*

2. DEPISTAREA DEPĂŞIRII LA ADUNAREA SĂ SCĂDEREA NUMERELOR BINARE

Sunt 2 metode de depistare a depăşirii:

a) Utilizarea codurilor modificate. În codul modificat pentru semn sunt rezervate două cifre numerice se consideră „+” dacă ambele cifre sunt egale cu zero şi „-” dacă ambele sunt egale cu 1. Dacă în urma îndeplinirii operaţii de adunare sau scădere semnul va avea unul dintre aceste două coduri 00 sau 11 aceasta va însemna că depăşirea nu a avut loc dacă însă în urma îndeplinirii operaţiei vom avea 01 aceasta înseamnă că a avut loc depăşirea pozitivă adică numărul obţinut e > decât cel mai mare număr sau 10 depăşirea negativă adică suma obţinută e mai mic număr.

Ex: 00,0101

00,1000

00,0101 - depăşire nu a avut loc deoarece avem 00

00,1010

00,1110

01,1000 - depăşire a avut loc

b) O altă metodă mai universală utilizată este metoda prin care depăşirea se depistează în urma adunării modulo 2 a biţilor de transport în şi din bitul semnului.

Dacă aceşti biţii de transport dau în sumă - 0 resepctiv depăşirea nu a avut loc, dacă suma modurilor 2 a acestor biţi de transport este 1 atunci depăşirea a avut loc.

0,0010
0,1000
0,1010 - 0⊕0=0 nu a avut loc

0,1010
0,1110
1,1000 - 1⊕0= 1 a avut loc z=0.11000

3. DEPLASAREA NUMERELOR BINARE

Pentru efectuarea anumitor operații în calculatoarele numerice poate apărea necesitatea de a deplasa cifrele codului binar.
Operația de deplasare poate fi de trei tipuri:
logică, ciclică și aritmetică.

Numerele binare pot fi deplasate la stânga și la dreaptă.

Deplasarea la stânga cu un rang corespunde cu înmulțirea numărului cu 2
Deplasarea la dreapta cu un rang corespunde cu împărțirea la 2.

Deplasarea poate fi necesară pentru efectuarea operațiilor de înmulțire, împărțire, și adunarea numerelor binare în virgulă mobilă.

Deplasarea logică:

Fie dat numărul A=00100110 ($38_{(10)}$).

DLS(A)=01001100 ($76_{(10)}$) – deplasarea logică la stânga;

DLD(A)=00010011 (19(10)) – deplasarea logică la dreapta

Notă:

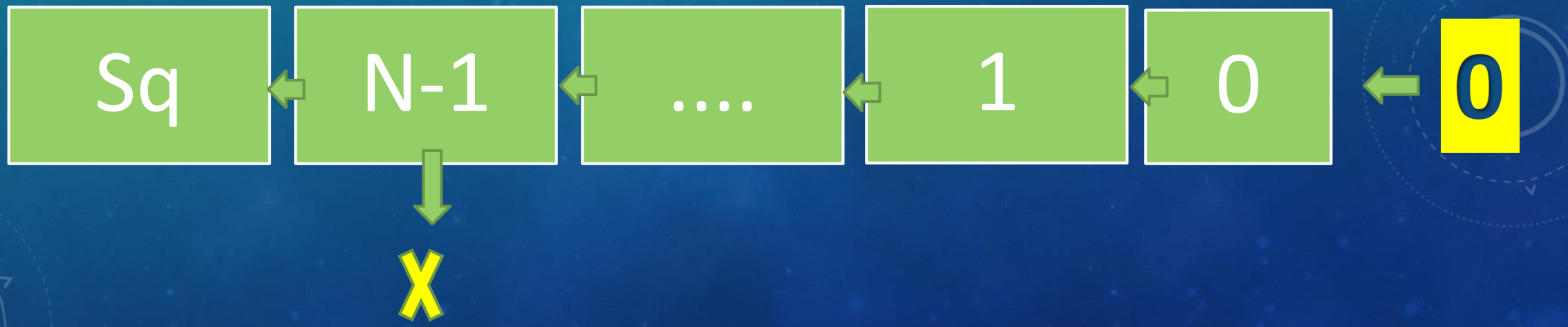
În cazul deplasării logice la stânga – se va deplasa cu o poziție toți biții la stânga și se va pierde cifra din rangul cel mai semnificativ, în bitul cel mai puțin semnificativ se încarcă- 0;

În cazul deplasării la dreapta se pierde cea mai puțin semnificativă cifră iar în cifra mai semnificativă se încarcă 0.

Deplasarea aritmetică:

Acest tip de deplasare specific este ca: bitul semnului se păstrează, la deplasarea la stânga se pierde cifra imediat următoare după bitul semnului în bitul cel mai puțin semnificativ se încarcă – 0:

A=0.0010101 (21D) DAS=0.0101010 (42D)
Acc=1.1011111 (-33D) DAScc=1.0111110 (-66D)



La deplasarea aritmetică la dreapta bitul semnului se dublează iat valoarea acestuia vine la cifra mai semnificativă respectiv cifra din rangul cel mai putin semnificativ se pierde.

A=0.0010101 (21D) DAD = 0.0001010 (10_D)
Acc=1.1011111 (-33_D) DADcc = 1.1101111 (-17_D)

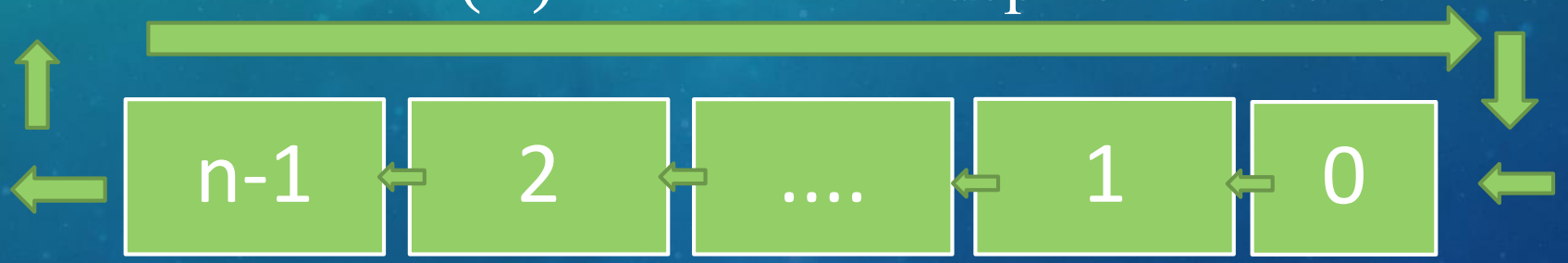
Deplasarea aritmetică poate fi realizată atît în codul complementar cît si în codul invers

Deplasarea ciclică:

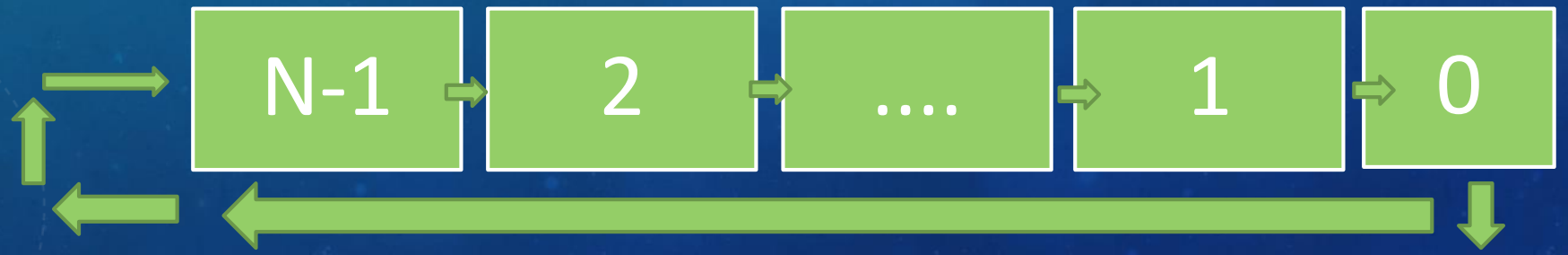
La deplasarea ciclică nu se pierde nici una din cifrele numărului deplasat, această se datorează legăturii ciclice(inelare) dintre bitul cel mai semnificativ și cel mai puțin semnificativ. În cazul deplasării la dreapta se mișcă toți biții cu o poziție la dreapta, iar cel mai puțin semnificativ bit se înscrie în locul celui mai semnificativ bit, în cazul deplasării la stînga toți biții se deplasează la stânga și cea mai semnificativă cifră se înscrie în locul celei mai puțin semnificative.

Exemplu: A=11010010

DCD(A)=10100101 – deplasarea ciclică la stînga.



DCS(A)=01101001 – deplasarea ciclică la dreapta.



A= 57 (111001) B=107

A+B

A=0.0111001 B=0.1101011

0.0111001

0.1101011

1.0100100

1 + 0=1 - depăsire - Z=0.10100100 =164

Vă mulțumesc pentru atenție!


ÎNTREBĂRI