IV.4.6.
Reprezentarea în complement faţă de 2

Complement

- Fie baza $d > 1$
- Complementul unei cifre:
  - Pentru o cifră $a \in \{0,1,\ldots d-1\}$

    $c_d(a) = (d - 1) - a$
  - Pentru $d = 2$ şi $b \in \{0,1\}$:

    $c_2(b) = (2 - 1) - b = 1 - b \rightarrow \overline{b}$
- Dar complementul unui şir de biţi?

Complement faţă de bază şi faţă de cifra maximă

- Extinderea definiţiei complementului la un şir de biţi se poate face în două moduri:
  - Conform definiţiei pentru un bit (complement faţă de cifra maximă)

    $C_1(1011) = 0100$ $\qquad$ $C_1(0100) = 1011$
  - Adaptând definiţia pentru şiruri (complement faţă de bază)

    $C_2(1011) = 0100 + 0001 = 0101$

    $C_2(0101) = 1010 + 0001 = 1011$

Reprezentarea în complement faţă de 2

- Cel mai frecvent utilizată
- Adunarea şi scăderea cu acelaşi algoritm / circuit
- Testarea automată a depăşirilor

C_2 - definiţie

- Complement faţă de 2: C_2
- $Val_{C2}^{n,m}(a_{n-1}a_{n-2}\ldots.a_1a_0a_{-1}\ldots..a_{-m}) =$

$$\begin{cases} a_{n-2}\times 2^{n-2}+\ldots+a_{-m}\times 2^{-m}, & \text{dacă } a_{n-1}=0 \\ (a_{n-2}\times 2^{n-2}+\ldots+a_{-m}\times 2^{-m}) - 2^{n-1}, & \text{dacă } a_{n-1}=1 \end{cases}$$

- Temă: strict negativ pentru $a_{n-1}=1$
  - Deci a_{n-1} reprezintă semnul

Reprezentarea în complement faţă de 2

- Cele 2^{n+m} reprezentări diferite (şiruri diferite de biţi) corespund la 2^{n+m} numere diferite
  - Neredundantă: $0 = Val_{C2}^{n+m}(00\ldots0)$
  - Temă: 0 nu poate fi reprezentat ca număr negativ
- Cel mai mic număr reprezentabil este

  $$min_{C2}^{n,m} = val_{C2}^{n,m}(10\ldots0) = -2^{n-1}$$
- Cel mai mare număr reprezentabil este

  $$max_{C2}^{n,m} = val_{C2}^{n,m}(01\ldots1) = 2^{n-1} - 2^{-m}$$
- Intervalul pe care se află numerele reprezentabile este deci $[-2^{n-1}; +(2^{n-1} - 2^{-m})]$

Reprezentarea în complement faţă de 2

- Numerele reprezentabile **exact** sunt cele începând cu min = -2^{n-1}, cu pasul 2^{-m}
- Celelalte numere din interval se reprezintă aproximativ, cu eroare de cel mult 2^{-m}
- **Precizia** reprezentării este 2^{-m}
  - pentru numere întregi, m=0, deci precizia este 1
- Pentru n+m fixat
  - creşterea magnitudinii duce la aproximare mai slabă
  - precizie mai bună duce la magnitudine scăzută

Reprezentarea în complement faţă de 2

- $Val_{C2}^{8,0}(00110011) = 51$

$00110011 \rightarrow + (2^0 + 2^1 + 2^4 + 2^5) = 51$

- $Val_{C2}^{6,2}(00110011) = 12,75 = 51 : 2^{\mathbf{2}}$

$00110011 \rightarrow + (2^{-2} + 2^{-1} + 2^2 + 2^3) = 12,75$

- $Val_{C2}^{4,4}(00110011) = 3,1875 = 51 : 2^{\mathbf{4}}$

$00110011 \rightarrow + (2^{-4} + 2^{-3} + 2^0 + 2^1) = 3,1875$

Reprezentarea în complement faţă de 2

- $Val_{C2}^{8,0}(10110011) = -77$

$10110011 \rightarrow (2^0 + 2^1 + 2^4 + 2^5) - 2^7 = 51 – 128 = -77$

- $min_{C2}^{8,0} = val_{C2}^{8,0}(10000000) = 0 - 2^7 = 0 - 128 = -128$
- $max_{C2}^{8,0} = val_{C2}^{8,0}(01111111) = 2^7 - 2^0 = 128 - 1 = 127$
- $[-128; 127] \rightarrow$ 256 numere, din 1 în 1
- $Val_{C2}^{4,4}(10110011) = -4{,}8125 = -77 : 2^4$
- $10110011 \rightarrow (2^{-4} + 2^{-3} + 2^0 + 2^1) - 2^3 = 3{,}1875 - 8 = -4{,}8125$
- $min_{C2}^{4,4} = val_{C2}^{4,4}(10000000) = 0 - 2^3 = -8 = -128 : 2^4$
- $max_{C2}^{4,4} = val_{C2}^{4,4}(01111111) = 2^3 - 2^{-4} = 8 - 0{,}0625 = 7{,}9375 = 127 : 2^4$
- $[-8; 7{,}9375] \rightarrow$ 256 numere din 0,0625 în 0,0625

C_2 - complementare

- Dată reprezentarea lui q, se poate afla automat reprezentarea lui -q?
- Dacă da, atunci scăderea p-q devine adunare, după generarea automată a reprezentării lui -q: p - q = p + (-q)
- Reprezentarea lui -q: complementul faţă de 2 al reprezentării lui q
- Exemplu: $q = -77 = Val_{C2}^{8,0}(10110011)$

  $-q = 77 = Val_{C2}^{8,0}(01001100 + 00000001)$

  $Val_{C2}^{8,0}(01001101) = 64 + 8 + 4 + 1$

Temă

- Reprezentarea în C_2 pe N biţi a numărului întreg negativ q este de fapt reprezentarea pe N biţi a numărului $q + 2^N = 2^N - |q|$
- $a + \overline{a} = 11....11 \rightarrow -1$, deci $(-a) = \overline{a} + 1$
  - $\overline{a}$ notează negaţia bit cu bit a reprezentării numărului a
  - a notează atât numărul, cât şi reprezentarea sa; se foloseşte aici implicit faptul că numerele pozitive se reprezintă ca în baza 2

Reprezentări în virgulă fixă, ^{4,0}

şirul de biţi	A+S	C_1	C_2	XS-7
0000	+0	+0	0	-7
0001	+1	+1	+1	-6
0010	+2	+2	+2	-5
0011	+3	+3	+3	-4
0100	+4	+4	+4	-3
0101	+5	+5	+5	-2
0110	+6	+6	+6	-1
0111	+7	+7	+7	0
1000	-0	-7	-8	+1
1001	-1	-6	-7	+2
1010	-2	-5	-6	+3
1011	-3	-4	-5	+4
1100	-4	-3	-4	+5
1101	-5	-2	-3	+6
1110	-6	-1	-2	+7
1111	-7	-0	-1	+8

IV.4.7.
Depăşiri pentru operaţii cu reprezentări în virgulă fixă

Trecerea la reprezentări mai lungi

- Adăugare de cifre nesemnificative *la partea întreagă*
- Ce înseamnă cifră nesemnificativă ?
  - A+S: zerouri imediat după cifra semn
    - » $Val_{A+S}^{8,0}(00110011) = 51$
    - » $Val_{A+S}^{16,0}(0000000000110011) = 51$
    - » $Val_{A+S}^{8,0}(10110011) = -51$
    - » $Val_{A+S}^{16,0}(1000000000110011) = -51$
  - C_1, C_2: repetări ale cifrei semn imediat după ea
    - » $Val_{C2}^{8,0}(0110011) = 51 = Val_{C2}^{16,0}(000000000110011)$
    - » $Val_{C2}^{8,0}(10110011) = -77$
    - » $Val_{C2}^{16,0}(1111111110110011) = -77$
    - » $Val_{C1}^{8,0}(10110011) = -76$
    - » $Val_{C1}^{16,0}(1111111110110011) = -76$

- Interesează mai mult trecerea la reprezentări mai scurte: încape reprezentarea corectă a rezultatului pe n+m biţi?

Mulţimea numerelor reprezentabile

- Submulţime finită a mulţimii numerelor raţionale
- $Q_{rep}^{n,m}$, unde rep $\in$ {A+S, C_1, C_2}
- $\mathbf{Q_{rep}^{n,m} \subset [min; max]}$ ←
- $Q_{C2}^{8,0}$ = {-128; -127; -126;…-1; 0; 1; ...127}
- $Q_{C2}^{4,4}$ = {-8; -7,9375; … -0,0625; 0; 0,0625; … 7,9375}
- $Q_{A+S}^{n,m}$ ={-(2^{n-1}-2^{-m});-(2^{n-1}-2^{-m+1});-(2^{n-1}-3×2^{-m}); …-2^{-m}; 0; +2^{-m}; … (2^{n-1}-2^{-m})}
- $\mathbf{Q_{rep}^{n,m} \subset Q_{rep}^{n+k,m}}$, k=1,2,… ←

Importanţa reprezentării în complement faţă de 2

- Cel mai simplu circuit de adunare a două reprezentări: sumator complet la fiecare rang şi ignorarea transportului de la rangul cel mai semnificativ
  - **Temă**: adunarea în C_1 implică o operaţie suplimentară
- Testare simplă a depăşirilor
- Un număr în plus reprezentat pe aceeaşi lungime (reprezentare neredundantă)

Operaţii în C_2

- Vom prezenta pe scurt doar proprietăţi ale adunării şi scăderii (suma algebrică) în C_2
- Termenii sumei şi rezultatul se reprezintă pe acelaşi număr de biţi
    - Uneori această restricţie cauzează depăşiri
- Înmulţirea şi împărţirea ar trebui să aibă, pentru reprezentarea rezultatului, o lungime dublă faţă de lungimea comună a operanzilor
    - Aceeaşi definiţie a depăşirii

Depăşiri

- Fie *rep* o reprezentare şi *op* o operaţie cu numere. Fie a, b $\in Q_{rep}^{n,m}$
- Spunem că operaţia *op* aplicată *numerelor* a şi b produce depăşire dacă:

$$a \; op \; b \notin [min; max] \leftarrow$$

  - Este una din situaţiile **repr(n1) *op* repr(n2) ≠ repr (n1 *op* n2)**
  - Anume, eroare la partea întreagă
  - Depăşire înseamnă **rezultat eronat**
  - Nu se poate produce depăşire la partea fracţionară
    - ci doar *aproximare* (rezultat aproximativ, nu eronat)

Depăşire - exemplul 1

- În reprezentarea C_2, cu n = 4, m = 0:

1111 + 1111 = 11110 → 1110

» Se "pierde" o cifră (rezultatul nu "încape" pe 4 biţi)

$val_{C2}^{4,0}(1111) = -1$

$(-1) + (-1) = -2$

$val_{C2}^{4,0}(1110) = -2$

» Rezultat corect, deci **nu se produce depăşire**

» Testarea depăşirii cu definiţia:

» $Q_{C2}^{4,0} \subset [-8; 7]$ $\quad$ $-1 \in Q_{C2}^{4,0}$ $\quad$ $-2 \in [-8; 7]$

» Calculatorul nu poate însă testa definiţia, întrucât "ştie" reprezentările, nu numerele reprezentate

Depăşire - exemplul 2

- În reprezentarea C_2, cu n = 4, m = 0 :

0111 + 0111 = 1110

» Nu se "pierde" nici o cifră

$val_{C2}^{4,0}(0111) = 7$

$7 + 7 = 14$

$val_{C2}^{4,0}(1110) = -2$

» Rezultat eronat, deci **se produce depăşire**

» Testarea depăşirii cu definiţia:

» $Q_{C2}^{4,0} \subset [-8; 7]$ $7 \in Q_{C2}^{4,0}$ $14 \notin [-8; 7]$

» Calculatorul nu poate însă testa definiţia, întrucât nu "ştie" numerele reprezentate

Depăşirea la adunare în C_2

- De fapt, cifra de transport de la rangul cel mai semnificativ (fie 0, fie 1) se pierde întotdeauna din rezultat
  - Dar se reţine în indicatorul C (transport general)
- Trebuie găsită o condiţie *asupra reprezentărilor* care, testată, să semnaleze depăşirea
  - Fără a folosi definiţia - numerele nu sunt disponibile
  - Testarea va fi făcută de un circuit ataşat sumatorului

Condiţia de depăşire

- Un alt mod de a vedea depăşirea la adunare:
  - Depăşire se produce numai atunci când ambii operanzi au acelaşi semn, iar reprezentarea rezultatului indică semn opus.
- Temă:
  - Nu este posibil ca definiţia depăşirii (cu numere) să fie satisfăcută când numerele au semn opus.
  - Studiul adunării reprezentărilor pentru cazurile: ambele numere pozitive, ambele numere negative. Concluzie.

Rezultate asupra sumei algebrice în C_2

- **Teorema 1.**

  Dacă a, b $\in Q_{C2}^{n,m}$, atunci $a \pm b \in Q_{C2}^{n+1,m}$

- **Lema 2.**

  Dacă

$a = Val_{C2}^{n+1,m}(\alpha_n\alpha_{n-1}\ldots\alpha_1\alpha_0\alpha_{-1}\ldots\alpha_{-m})$, cu $\alpha_n = \alpha_{n-1}$

  atunci

$a = Val_{C2}^{n,m}(\alpha_{n-1}\alpha_{n-2}\ldots\alpha_1\alpha_0\alpha_{-1}\ldots\alpha_{-m})$

Rezultate asupra sumei algebrice în C_2

- **Definiţie.** Date reprezentările $\boldsymbol{\alpha}$ şi $\boldsymbol{\beta}$,
  $\boldsymbol{\alpha} = \alpha_{n-1}\ldots\alpha_1\alpha_0\alpha_{-1}\ldots\alpha_{-m}$ şi
  $\boldsymbol{\beta} = \beta_{n-1}\ldots\beta_1\beta_0\beta_{-1}\ldots\beta_{-m}$ $(\alpha_i, \beta_i \in \{0,1\})$,
  suma lor formală $\boldsymbol{\gamma} = \boldsymbol{\alpha} + \boldsymbol{\beta}$ este
  $\boldsymbol{\gamma} = \gamma_n \gamma_{n-1} \ldots\gamma_1\gamma_0 \gamma_{-1}\ldots \gamma_{-m}$, unde
  $$\Sigma_{i=-m}^{n}(\gamma_i \times 2^i) = \Sigma_{i=-m}^{n-1}((\alpha_i + \beta_i)\times 2^i)$$
- **Teorema 3.** Dacă suma algebrică a numerelor reprezentate de $\boldsymbol{\alpha}$ şi $\boldsymbol{\beta}$ nu produce depăşire, atunci reprezentarea rezultatului este
  $$\gamma_{n-1} \ldots\gamma_1\gamma_0 \gamma_{-1}\ldots \gamma_{-m}$$
- **Consecinţă:** sumatorul serial efectuează adunări corecte
  - semnul se adună ca oricare altă cifră

Testarea depăşirii

- **Teorema 4.**
- Suma algebrică a numerelor reprezentate de **α** şi **β** nu produce depăşire dacă cifrele transport C_{n-1} şi C_n coincid.
  - 1111 + 1111 : cifrele-transport C_3 şi C_4 sunt identice (egale cu 1)
  - 0111 + 0111 : cifrele-transport C_3 şi C_4 diferă (prima este 1, a doua este 0)
- **Consecinţă:** testarea depăşirii în UAL prin adăugarea la sumatorul serial a unei porţi NXOR
  - în care intră cifrele-transport C_{n-1} şi C_n
  - şi din care iese bitul-flag **O** (Overflow)
  - cifra-transport C_n poziţionează şi bitul-flag **C** (carry)