Curs 2: Reprezentarea numerelor si aritmetica binara

Vlad-Cristian Miclea

Universitatea Tehnica of Cluj-Napoca

October 13, 2022

Cuprins

1. Introducere
2. Reprezentarea numerelor
   - Reprezentarea in virgula fixa
     - Marime si semn
     - Complement fata de 2
     - Complement fata de 1
   - Reprezentarea in virgula mobila
3. Artimetica binara
   - Operatii intregi, fara semn
   - Operatii – reprezentare in virgula fixa
4. Concluzii

Lecture syllabus

1. Sisteme de numeratie si coduri
2. **Reprezentarea numerelor. Aritmetica binara**
3. Algebra booleana
4. Metode de minimizare a functiilor booleene
5. Circuite logice combinationale
6. Sinteza circuitelor digitale cu SSI, MSI, LSI and VLSI circuits
7. Circuite logice secventiale. Bistabile
8. Numaratoare
9. Registre. Memorii
10. Sinteza circuitelor digitale folosind bistabile
11. Sinteza circuitelor digitale folosind memorii, numaratoare etc.
12. Sisteme secventiale sincrone
13. Sinteza circuitelor digitale folosind circuite logice programabile
14. Probleme finale

Reprezentarea numerelor in calculator

Generalitati

- Reprezentarea numerelor in calculator – sisteme binare
- Numere pozitive si negative – cum sa specificam semnul?
  - Numere fara semn – numere binare sau BCD
  - Numere cu semn – pozitia cea mai semnificativa (MSB) – bit special de semn!
  - Conventie: 0 – pozitiv; 1 – negativ
  - numar pe n-biti $\rightarrow$ $n + 1$ biti
- Numere intregi si fractionare
  - Numere fractionare – pozitia virgulei $\rightarrow$ reprezentarea
  - Reprezentarea in virgula fixa (Fixed-point)
  - Reprezentarea in virgula mobila (Floating-point)

Reprezentarea in virgula fixa

Generalitati

- Calculatoarele pot opera cu numere de lungime fixa (pastrate in memorie)
- Numarul de cifre – predefinit (32/64 pozitii binare)
- Pozitia virgulei
  - Se stabileste initial, la proiectare
  - Nu se reprezinta fizic, dar locatia ei este cunoscuta
- Blocuri aritmetice care folosesc virgula fixa
  - Pozitia virgulei e considerata inainte bitului cel mai semnificativ MSB
  - Toate numerele sunt sub-unitare (transformate inainte de operatii)

Tipuri de reprezentare – date de semn

- Marime si semn
- Complement fata de 2
- Complement fata de 1

Complement

Definitii si concepte

- $\overline{(N)_b} \rightarrow$ complement fata de baza b a numarului $(N)_b$

$$\overline{(N)_b} = b^n - (N)_b \tag{1}$$

- $\overline{\overline{(N)_b}} \rightarrow$ complement fata de baza $b-1$ a numarului $(N)_b$

$$\overline{\overline{(N)_b}} = b^n - (N)_b - b^{-m} \tag{2}$$

- n $\rightarrow$ numarul de cifre a partii intregi ale numarului N
- m $\rightarrow$ numarul de cifre a partii fractionare ale numarului N
- b^n nu poate fi reprezentat folosind doar n cifre, asadar b^n este echivalat cu numarul 0

Complement

Exemplu

- $N_1 = (123,45)_{10}$, $n = 3$ si $m = 2$

$$\overline{(N_1)_{10}} = 10^n - N_1 = 10^3 - 123.45 = 876.55$$
$$\overline{\overline{(N_1)_{10}}} = 10^n - N_1 - 10^{-m} = 10^3 - 123.45 - 10^{-2} = 876.54$$

- $N_2 = (1101,011)_2$, $n = 4$ si $m = 3$

$$\overline{(N_2)_2} = 2^n - N_2 = 2^4 - 1101,011 = 0010,101$$
$$\overline{\overline{(N_2)_2}} = 2^n - N_2 - 2^{-m} = 2^4 - 1101,011 - 2^{-3} = 0010,100$$

Complement

000000 −
101011

Determinarea complementului fata de 2 – 3 metode

- $\overline{(N)_2} = 0 - N$ (eq 3)
- $\overline{(N)_2} = \overline{\overline{N}} + 2^{-m}$ (eq 4)
- Procedura (eq 5):
  - Pornim de la dreapta la stanga
  - Pastram toate 0-urile neschimbate, pana la prima valoare de 1
  - Pastram prima valoare de 1 neschimbata
  - Pornind de la aceasta pozitie, inversam toate cifrele

Exemplu: $(N)_2 = 101011$

$$\overline{N_2} = 000000 - 101011 = 010101 \quad (3)$$

$$\overline{N_2} = 010100 + 000001 = 010101 \quad (4)$$

$$\overline{N_2} = 010101 \quad (5)$$

101011
010101

Complement

Determinarea complementului fata de $1 - 3$ metode

- $\overline{\overline{(N)_2}} = 0 - N - 2^{-m}$ (eq 6)
- $\overline{\overline{(N)_2}} = \overline{N} - 2^{-m}$ (eq 7)
- Inversam toate cifrele (eq 8)

Exemplu: $(N)_2 = 101011$

$$\begin{aligned}
\overline{\overline{N_2}} &= 000000 - \\
&\quad 101011 - \\
&\quad 000001 = \\
&\quad 010100 \qquad (6)\\
\overline{\overline{N_2}} &= 010101 - \\
&\quad 000001 = \\
&\quad 010100 \qquad (7)\\
\overline{\overline{N_2}} &= 010100 \qquad (8)
\end{aligned}$$

Marime si semn

Generalitati

- Relatia de reprezentare:

$$N = a_n 2^n + \sum_{i=-m}^{n-1} a_i 2^i$$

- a_n - bit de semn
  - N - pozitiv $\rightarrow a_n = 0$
  - N - negativ $\rightarrow a_n = 1$
- a_i – cifrele binare ale numarului N

Advantaje si dezavantaje

- Advantaj: Similar cu scrierea manuala
- Dezavantaje pentru realizarea calculelor aritmetice
  - Evaluarea semnului e necesara inainte de efectuarea operatiilor
  - Adunarea si scaderea au nevoie de blocuri diferite
- Exemplu: $+6 = 00110$; $-6 = 10110$

Complement fata de 2

Generalitati

- Relatiile de reprezentare:

$$N = 0 \times 2^n + \sum_{i=-m}^{n-1} a_i 2^i \text{ if } N > 0$$

$$N = 1 \times 2^n + \sum_{i=-m}^{n-1} \overline{a_i} 2^i + 2^{-m} \text{ if } N < 0$$

- $\overline{a_i}$ e complement-ul cifrei a_i
- Exemplu:
  - $+6 = 00110$
  - $-6 = 11010$ (1010 – complementul fata de 2 al cifrei 6)

Complement fata de 1

Generalitati

- Relatiile de reprezentare:

$$N = 0 \times 2^n + \sum_{i=-m}^{n-1} a_i 2^i \text{ if } N > 0$$

$$N = 1 \times 2^n + \sum_{i=-m}^{n-1} \overline{a_i} 2^i \text{ if } N < 0$$

- $\overline{a_i}$ e complementul cifrei binare a_i
- Exemplu:
  - $+6 = 00110$
  - $-6 = 11001$ (1001 – e complementul fata de 1 al cifrei 6)
- Cum reprezentam numarul 0 in complement fata de 1 si fata de 2?

Reprezentarea in virgula fixa

Rezumat

- In general, putem scrie fiecare numar $N = a_n 2^n + N^*$
- a_n - bit de semn
- N^* e reprezentat:
  - Marime si semn: $N^* = \sum_{i=0}^{n-1} a_i 2^i$
  - $N < 0$, in complement fata de 2: $N^* = \sum_{i=0}^{n-1} \overline{a_i} 2^i + 2^0$
  - $N < 0$, in complement fata de 1: $N^* = \sum_{i=0}^{n-1} \overline{a_i} 2^i$
- a_i - cifre; n - numar de cifre (dupa virgula); $\overline{a_i} = 1 - a_i$

Pozitionarea virgulei dupa prima cifra binara

- Virgula va avea aceeasi pozitie dupa o inmultire
- Plasarea poate fi usor memorata

Reprezentarea in virgula mobila

Generalities

- Reprezentarea in virgula fixa – probleme la reprezentarea unor numere foarte mari sau foarte mici
- Reprezentarea in virgula mobila – permite modificarea pozitiei virgulei
- **Exponent** – indica ordinul de marime a numarului, printr-o putere
- **Mantisa** – indica valoarea exacta a numarului, in cadrul ordinului respectiv

Exemplu – reprezentare folosind 32-biti

SE	E	E	E	E	E	E	SM	M . . . M	M
0	1	2	3	4	5	6	7	. . .	31

- bit 0 = SE - semnul exponentului; bit 7 = SM - semnul mantisei;
- biti 1-6: exponent; biti 8-31: mantisa

Reprezentarea in virgula mobila

Exemplu

- Consideram numarul zecimal $+12.34$
- Its representation can be:
  - $+1100.01010111$ – folosind algoritmul de saptamana trecuta (precizie de 12-biti)
  - $+0.110001010111 \times 2^{+4}$
  - $+0.00110001010111 \times 2^{+6}$
- Folosind reprezentare in virgula mobila pe 32-biti:
  - $0\,000100\,0\,11000101011100000000000$
  - $0\,000110\,0\,00110001010111000000000$

Reprezentarea in virgula mobila

Caracteristica

- Reprezentarea prezentata anterior poate fi eficientizata
- Exponentul nu are nevoie de semn $\rightarrow$ se foloseste marimiea numita caracteristica
- Toate operatiile legate de carcateristica vor fi pozitive
- Caracteristica e obtinuta ca C = E + deplasament
  - Deplasamentul e ales incat sa rezulte intotdeauna o caracteristica pozitiva
- Problema: E nevoie de o scadere suplimentara: E = C - deplasament

Reprezentarea in virgula mobila

Caracteristica

- Reprezentare in precizie simpla

S	E			M			
0	1		7	8	. . .		31

- Reprezentare in precizie dubla

S	E			M			
0	1		7	8	. . .		63

- Exemplu
  - Pentru caracteristica C = 7 bits: numerele sunt intre 0 si $2^7 - 1$ $\rightarrow 0 \leq C \leq 127$
  - Daca deplasamentuls = 64: exponent E = C-64, deci $-64 \leq E \leq 63$

Reprezentarea in virgula mobila

Forma normalizata

- Reprezentarea poate fi si mai eficienta
- Daca exponentul/caracteristica e corect selectata, nu avem nevoie de primul bit din mantisa (e intotdeauna 1) – vezi exemplu pt 12.34
- Aceasta varianta a reprezentarii e numita **forma normalizata**
- Putem astfel creste gama numerelor si precizia operatiilor
- Problema: reprezentarea valorii 0 – e nevoie de o forma speciala

Reprezentarea in virgula mobila

Gama numerelor reprezentate in precizie simpla – numere de 32-biti reprezentate in complement fata de 2

- Numere pozitive: intre 0.5×2^{-128} si $(1 - 2^{-24} \times 2^{127})$
- Numere negative: intre $-(1 - 2^{-24}) \times 2^{127}$ si -0.5×2^{-128}
- 5 regiuni care nu sunt cuprine in aceste domenii:
  - Depasire inferioara negativa (prea "mici"): $< -(1 - 2^{-24}) \times 2^{127}$
  - Depasire superioara negativa (prea aproape de 0): $> -0.5 \times 2^{-128}$
  - Zero
  - Depasire inferioara pozitiva: $< 0.5 \times 2^{-128}$
  - Depasire superioara pozitiva: $> (1 - 2^{-24}) \times 2^{127}$
- In general, exista mecanisme speciale pentru detectia, semnalizarea si tratarea acestor cazuri.

Reprezentarea in virgula mobila

Compromis intre dimensiunea exponentului si a mantisei

- Mai multi biti pentru mantisa: Precizie ridicata
- Mai multi biti pentru exponent: Gama mai larga de reprezentare

Standarde

- Exista standarde care specifica deciziile de design pentru reprezentarea numerelor in virgula mobila
- IEEE 754, din 1985 (modificat in 2008) – pt numere binare
- IEEE 854, din 1987 – pentru alte baze
- Aceste standarde specifica: formate, codificari, rotunjiri, operatii, exceptii etc.

Artimetica binara: operatii fara semn

Adunarea binara

- La baza: Operatie modulo 2 – adunare
- Cea mai mare cifra: 1
- Daca rezultatul adunarii a e cifre de rang i e mai mare decat 1 $\rightarrow$ avem nevoie de un **transport** spre pozitia (rangul) $i+1$, care e adunata la suma cifrelor de rang $i+1$;
- Problema la adunare, la cifra de pe pozitia cea mai semnificativa – de ce?

x	y	Transport	Suma
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Exemplu:

$$22_{10} = 10110_2 +$$
$$19_{10} = 10011_2$$
$$\overline{\qquad\qquad\qquad}$$
$$41_{10} = 101001_2$$

Artimetica binara: operatii fara semn

Scaderea binara

- Cand scadem 2 cifre de rang i, am putea avea nevoie de un **imprumut** de la cifra de rang $i + 1$;
- In general, nu ne place sa efectuam scaderea in acest fel

x	y	Imprumut	Diferenta
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Exemplu:

$$22_{10} = 10110_2 -$$
$$19_{10} = 10011_2$$

$$3_{10} = 00011_2$$

Artimetica binara: operatii fara semn

Inmultirea binara

- Operatia e realizata prin adunarea mai multor produse partiale

x	y	Produs
0	0	0
0	1	0
1	0	0
1	1	1

Exemplu:

$$12_{10} = 1100_2 \times$$
$$6_{10} = 0110_2$$

$$0000$$
$$1100$$
$$1100$$
$$0000$$

$$1001000_2$$
$$(64 + 8 = 72_{10})$$

Artimetica binara: operatii fara semn

Impartirea binara

- Nu putem imparti la 0!
- Trebuie satisfacuta relatia: $X = Q \times Y + R$
- X – deimpartit; Q – cat; Y – impartitor; R – rest
- Scaderi multiple ale impartitorului din resturile partiale (RP)
- Daca $RP > Y$ atunci Catul e 1
- Altfel, Catul e 0

Exemplu: $147_{10} = 10010011_2$; $11_{10} = 1011_2$

$$
\begin{array}{ll}
10010011 & : 1011 = 1101_2 = 13_{10} \\
1011 & \\
\hline
1110 & \\
1011 & \\
\hline
1111 & \\
1011 & \\
\hline
100 = 4_{10} &
\end{array}
$$

Operatii – reprezentare in virgula fixa

Adunarea pentru numere in complement fata de 2

- Numerele se aduna bit-cu-bit, inclusiv bitii de semn
- Se ignora bitii de transport la bitii de semn
- Daca rezultatul e negativ, apare ca un numar reprezentat in complement fata de 2
- Daca rezultatul e mai mare decat valoarea maxima reprezentabila $\rightarrow$ depasire
- Cand adunam 2 numere cu semne similare, apare depasire daca rezultatul are un semn diferit de operanzi

Operatii – reprezentare in virgula fixa

Adunarea pentru numere in complement fata de 2 – exemplu

$+9_{10}$ $0\,1001_2+$
$+5_{10}$ $0\,0101_2 =$
$+14_{10}$ $0\,1110_2$

$+9_{10}$ $0\,1001_2+$
$+11_{10}$ $0\,1011_2 =$
$+20_{10}$ $1\,0100_2 \rightarrow$ Rezultat incorect

-9_{10} $1\,1001_2+$
-5_{10} $1\,1011_2 =$
-14_{10} $11\,0010_2$

-9_{10} $1\,0111_2+$
-11_{10} $1\,0101_2 =$
-20_{10} $10\,1100_2 \rightarrow$ Rezultat incorect

$+7_{10}$ $0\,0111_2+$
-4_{10} $1\,1100_2 =$
$+3_{10}$ $10\,0011_2$

-7_{10} $1\,1001_2+$
$+4_{10}$ $0\,0100_2 =$
-3_{10} $1\,1101_2$

Operatii – reprezentare in virgula fixa

Scaderea pentru numere in complement fata de 2

- Exista 2 metode:
  - Prin scadere directa, daca exista scazatoare elementare
  - Prin adunarea cu complementul fata de 2 a scazatorului ($a - b \Leftrightarrow a + (-b)$)
- Pot sa apara depasiri, care trebuie detectate
- La scaderea unor numere de semne contrare pot sa apara depasiri daca si numai daca rezultatul are acelasi semn cu scazatorul

Concluzii

Rezumat

- Reprezentarea numerelor in calculator
  - Numere fara semn
  - Complement – Concept, complement fata de 1, complement fata de 2
  - Marime si semn
  - Reprezentarea in virgula fixa
  - Reprezentarea in virgula mobila
- Aritmetica binara
  - Numere fara semn: adunare, scadere, inmultire, impartire
  - Operatii in virgula fixa – adunare si scadere in complement fata de 2

Next week

- Algebra booleana. Functii booleene

Multumesc pentru atentie!