04.12.2024

Arhitectura calculatoarelor
(curs 10-S10)

ex curs:

$X = +6.5_{(10)} = +110.1 \cdot 2^0_{(2)} = 1.101 \cdot 2^2$

$Y = -7_{(10)} = -111.0 \cdot 2^0 = -1.110 \cdot 2^2$

$\Rightarrow X$: | 0 | 1 | 0 | 1 | .1 | 0 | 1 |

bias $= 2+3 = 5_{(10)} = 101_{(2)}$

Y: | 1 | 1 | 0 | 1 | .1 | 1 | 0 |

Y_E

1. X: S | 0 | 1 | 0 | 1 | 1 | .1 | 0 | 1 |, $\underbrace{1\,0\,1}_{X_E}$

   Y: S | 1 | 1 | 0 | 1 | 1 | .1 | 1 | 0 |, $\underbrace{1\,0\,1}_{Y_E}$

2. $d = X_E - Y_E = 5 - 5 = 0$

   $\Rightarrow$ alegem $\boxed{Z_E = X_E = 5}$

3. if sign(x) $\neq$ sign y then $Y_M \rightarrow C_2$

   Y: S | 1 | 1 | 0 | 1 | 1 | .1 | 1 | 0 |, $\underbrace{1\,0\,1}_{Y_E}$

   $Y_{M_{C_2}}$: | 0 | .0 | 1 | 0 |

4. $\rightarrow$ aliniere Y_M prin deplasare la dreapta cu $d = 0$ poziții

   $\longrightarrow$ dacă l-am deplasa, pt. că l-am complementat, am introduce 1

   $Y_{M_{C_2}}$: | 0 | .0 | 1 | 0 |

   g r s

   | 0 | .0 | 1 | 0 | 0 0 0 (nu am deplasat)

5. Se adună $x_M + y_{MAL}$ și se verifică condițiile
if (sign(x) = sign(y))
then keep generate (cout)

if (sign(x) != sign(y) && ! generate (cout))
then $z_M < 0$ && -> C_2

if (sign(x) != sign(y) && generate(cout))
then $z_M > 0$ && cout neglijat

x_M	1.101
y_{MAL}	0.010	0 000

⊕

z_M : 0 | 1.1 1 1 0 0 0

Nu avem cout și sign(x)!= sign(y)
=> $z_M < 0$

inversăm

$\xrightarrow{C_2} z_M$: 0.00 1 0 0 0 (g r s)

??

z_{m-2} z_{m-3}

6. -> caz 4
de normalizare
deplasăm la stânga cu 3 poziții z_E -= 3

R S

z_{M_M} : 1.000

$\boxed{z_E = 5-3=2}$

7. $R = g = 0$

$S = (r \text{ or } s) = 0$

8. if ($\overset{0}{R}$ and($\overset{0}{S}$ or $\overset{,0}{z_{0_M}}$)) = 0 $\boxed{z_M < 0}$

then $z_{M_M} - 1$

rămâne

z_{M_M} : | 1 | .0 | 0 | 0 | R: 0 | S: 0 |

$z_E = 2$ (din pasul 6)

9. swap(x, y) ⟶ NO

C_2 (pas 5) ⟶ DA

sign(x) ⟶ +

sign(y) ⟶ −

$\Rightarrow$ sign(z) = -

10. Impachetare

sign(z) = -

exponent(z) = $2_{(10)} = 010_{(2)}$

significand(z) = 1.000

z : | $\underline{1}$ | 0 | 1 | 0 | .0 | 0 | 0 |

$\underbrace{010}_{z_E}$ $\underbrace{000}_{z_M^*}$

Verificare

$x + y = +6.5 - 7 = -0.5_{(10)}$

$z = (-1)^{z\,sign} \cdot 2^{z_E - bias} \cdot z_M =$

$= (-1)^1 \cdot 2^{-1} \cdot 1.000 = -0.100_{(2)} = -0.5_{(10)}$ ✓

Proiectarea unui dispozitiv pentru pre-normalizare:

- acoperă pașii 6 și 7
- proiectat după regulile din secțiunea 3.3
- design combinațional

Rezultatul pasului 5, pentru formatul considerat, este:

$$5.\quad z_M = z_{4_M} z_{3_M} . z_{2_M} z_{1_M} z_{0_M} \mid g \quad r \quad s$$

$$6. + 7.\quad z_M = 1 . z_{2_M} z_{1_M} z_{0_M} \mid R \quad S$$

Pornind de la regulile de normalizare din secțiunea 3.3, pentru formatul dat, Z_M poate necesita:

- deplasare la dreapta cu 1 bit (r_1), sau
- pastrarea fără modificare (rezultat gata normalizat) (l/r_0), sau
- deplasare la stânga cu 1 bit (l_1), sau
- deplasare la stânga cu 2 biți (l_3), sau
- deplasare la stânga cu 3 biți (l_3)

Cele 5 cazuri de normalizare sunt identificate de cele 5 condiții/variabile între paranteze: r_1, l/r_0, l_1, l_2, l_3.

Una și doar una dintre cele 5 condiții va fi activă pentru o pereche dată de operanzi FP de însumat după algoritmul de adunare cu rotunjire.

Cazuri de normalizare

$$5.\quad z_M = z_{4_M} z_{3_M} . z_{2_M} z_{1_M} z_{0_M} \mid g \quad r \quad s$$

6. + 7. $z_M =$	1	. z_{2_M}	z_{1_M}	z_{0_M}	R	S
1. r_1 ; >>1 ; $z_4=1$	$\overset{z_4}{1}$	. z_3	z_2	z_1	z_0	g r r r s
2. l/r_0 ; nicio deplasare ; $z_4=0$, $z_3=1$	$\overset{z_3}{1}$	. z_2	z_1	z_0	g	r r s
3. l_1 ; <<1 ; $z_4=0$, $z_3=0$, $z_2=1$! A pt. nr. negative	$\overset{z_2}{1}$	. z_1	z_0	g	r	s

④ ℓ_2; <<2;	$z_4=0$, $z_3=0$, $z_2=0$, $z_1=1$	z_1 1.	z_0	g	0	0	0
⑤ ℓ_3; <<3	$z_4=0$, $z_3=0$, $z_2=0$, $z_1=0$, $z_0=1$	z_0 1.	g	0	0	0	0

În consecință avem următoarele ecuații booleene pentru cei 5 biți generați la ieșirea pasului 6 combinat cu pasul 7.

$$z_{2n} = z_2 \cdot \ell/r_0 + z_1 \cdot \ell_1 + z_0 \cdot \ell_2 + g \cdot \ell_3 + z_3 \cdot r_1$$

$$z_{1n} = z_1 \cdot \ell/r_0 + z_0 \cdot \ell_1 + g \cdot \ell_2 + z_2 \cdot r_1$$

$$z_{0n} = z_0 \cdot \ell/r_0 + g \cdot \ell_1 + z_1 \cdot r_1$$

$$R = g \cdot \ell/r_0 + r \cdot \ell_1 + z_0 \cdot r_1$$

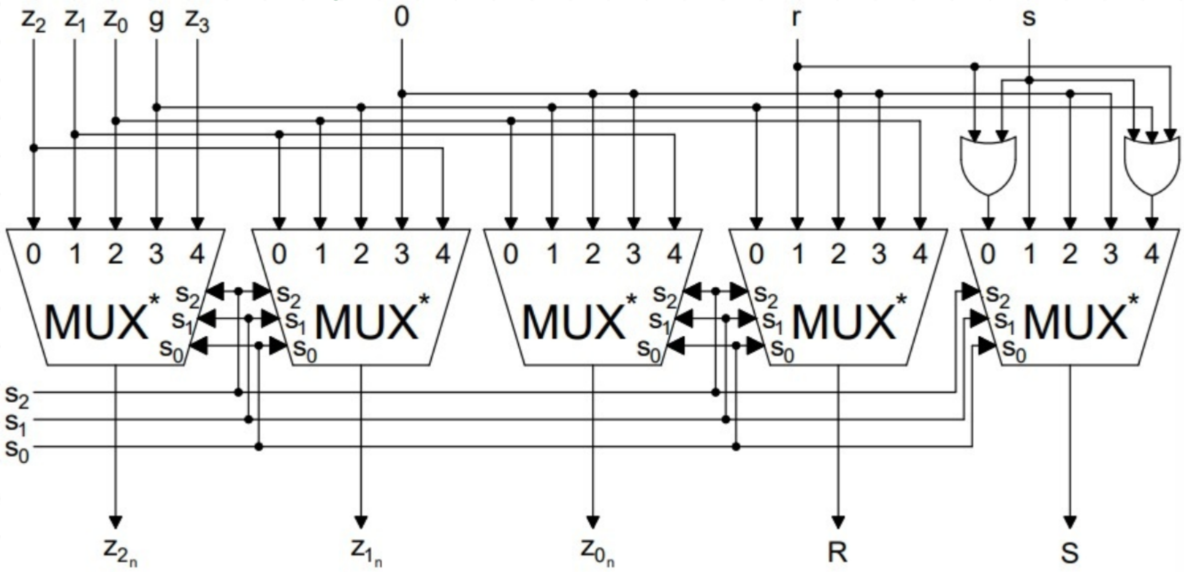
$$S = (r \text{ or } s) \cdot \ell/r_0 + s \cdot \ell_1 + (g \text{ or } r \text{ or } s) \cdot r_1$$

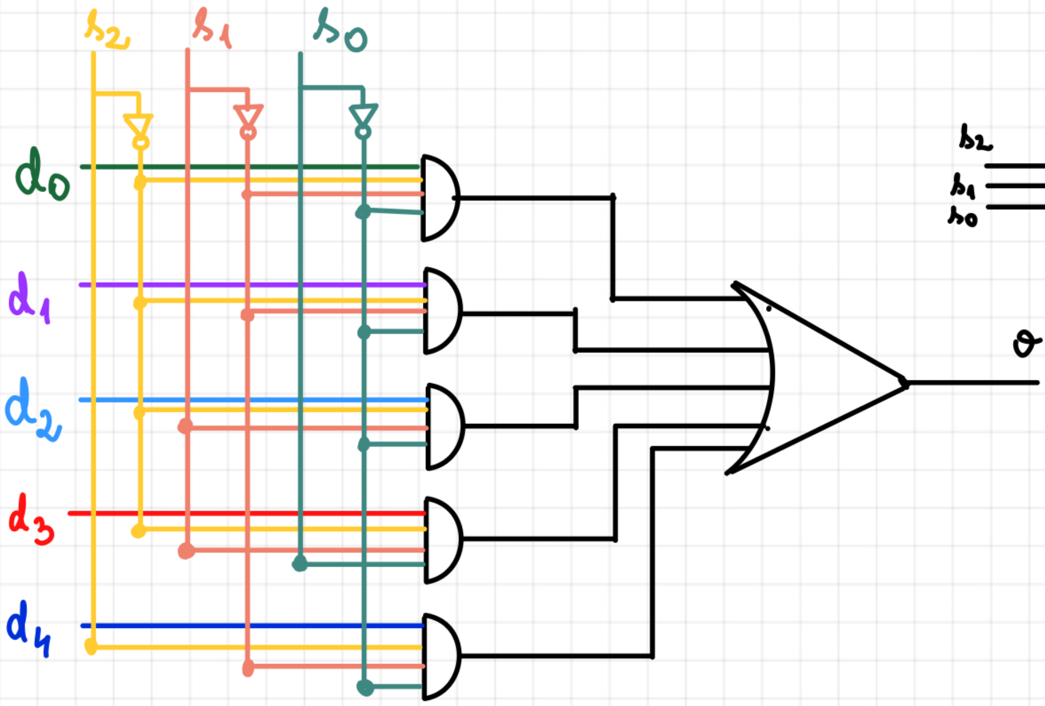
Deoarece, pentru o pereche de operanzi FP adunați prin algoritmul de adunare cu rotunjire, una și doar una din cele 5 condiții poate fi activă $\Rightarrow$ acestea pot fi codificate pe mai puțini biți.

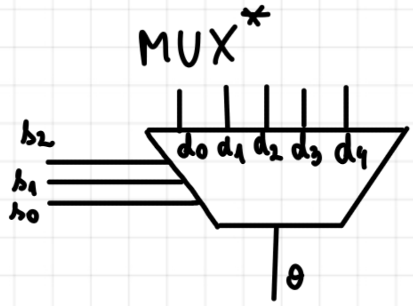
Se consideră variabilele s_2, s_1 și s_0 pentru codificarea celor 5 condiții. Condificarea este descrisă în tabelul următor:

	Intrări					Ieșiri		
	r_1	I_3	I_2	I_1	I/r_0	s_2	s_1	s_0
1	0	0	0	0	1	0	0	0
2	0	0	0	1	0	0	0	1
3	0	0	1	0	0	0	1	0
4	0	1	0	0	0	0	1	1
5	1	0	0	0	0	1	0	0
							d	d

Arhitectură dispozitiv pre-normalizare







Simbolurile *MUX** sunt multiplexoare incomplete, cu 3 linii de selecție dar având doar 5 intrări de date.

IV. Analiza funcțională și sinteza dispozitivelor de înmulțire binară

4.1. Dispozitive de înmulțire

Un înmulțitor calculează produsul:

$$P = X \cdot Y \quad , \text{ unde}$$

$X \longrightarrow$ înmulțitor

$Y \longrightarrow$ deînmulțit

$$X \cdot Y = X_M \cdot Y_M \cdot 2^{X_E + Y_E}$$

ex: Se consideră operanzii fără semn X și Y pe 4 biți

$$X = 11_{(10)} = 1011_{(2)}$$

$$Y = 12_{(10)} = 1100_{(2)}$$

A. Paper and pencil

```
          1 1 0 0 ....................... Y
          1 0 1 1 = x3x2x1x0 ............ X
        ---------------------------------
          1 1 0 0 ........... x0·Y·2^0
        1 1 0 0 ............. x1·Y·2^1
      0 0 0 0 ............... x2·Y·2^2
    1 1 0 0 ................. x3·Y·2^3
  ---------------------------------
  1 0 0 0 0 1 0 0 ........... P = Σ_{i=0}^{3} xi·Y·2^i = 132
  |         |
  128       4
```

$$P = \sum_{i=0}^{3} x_i \cdot Y \cdot 2^i = 132$$

Investiție hardware

- 2 registre pe 4 biți pentru x și y
- sumator multi-operand (CSA de regulă) are latență ridicată
- "gating"-ul deînmulțitului

Număr maxim de biți : $n + m$

B. Păstrarea fixă a produselor parțiale

1100		y
1011	$= x_3x_2x_1x_0$	x
00000000		$P_0 := 0$
1100		$x_0 \cdot y \cdot 2^0$
00001100		$P_1 := P_0 + x_0 y \cdot 2^0$
0001100		$x_1 \cdot y \cdot 2^1$
00100100		$P_2 := P_1 + x_1 \cdot y \cdot 2^1$
0000		$x_2 \cdot y \cdot 2^2$
00100100		$P_3 := P_2 + x_2 \cdot y \cdot 2^2$
1100		$x_3 \cdot y \cdot 2^3$
10000100		$P_4 := P_3 + x_3 \cdot y \cdot 2^3 = P$

Pasul de iterație

$$P_{i+1} = P_i + x_i \cdot y \cdot 2^i \text{ , pentru } i \geq 0,\ P_0 := 0$$

P_i – produs parțial

$x_i \cdot y \cdot 2^i$ – produs de un bit

Investiție hardware

- 2 registre pe 4 biți pentru x și y
- registru de 8 biți pentru produsele parțiale / rezultat
- sumator pe 8 biți
- mecanism de aliniere a produselor de 1 bit (LShift)

C. Păstrarea fixă a produselor de 1 bit

1 1 0 0 .... Y

1 0 1 1 $= x_3 x_2 x_1 x_0$ X

---

0 0 0 0   0 0 0 0 $P_0 := 0$

$\boxed{1\,1\,0\,0}$ $x_0 y$ ⊕

---

0 0 0 0   1 1 0 0 $P_0 := P_0 + x_0 \cdot y$

0 0 0   0 1 1 0 0 $P_1 := P_0 \cdot 2^{-1}$

$\boxed{1\,1\,0\,0}$ fix $x_1 \cdot y$

---

0 0 1   0 0 1 0 0 $P_1 := P_1 + x_1 \cdot y$

0 0   1 0 0 1 0 0 $P_2 := P_1 \cdot 2^{-1}$

$\boxed{0\,0\,0\,0}$ $x_2 \cdot y$

---

0 0   1 0 0 1 0 0 $P_2 := P_2 + x_2 \cdot y$

0   0 1 0 0 1 0 0 $P_3 := P_2 \cdot 2^{-1}$

$\boxed{1\,1\,0\,0}$ $x_3 \cdot y$

---

1   0 0 0 0 1 0 0 $P_3 := P_3 + x_3 y$

1 0 0 0 0 1 0 0 $P_4 := P_3 \cdot 2^{-1} = P$

Pasul de iterație: $\begin{cases} P_i := P_i + X_i \cdot Y \\ P_{i+1} := P_i \cdot 2^{-1} \end{cases}$, $i \geq 0$ și $P_0 := 0$

Investiție hardware

▷ 2 registre pe 4 biți pentru X și Y

▷ registru de 8 biți pentru produsele parțiale / rezultat cu facilitate de deplasare la dreapta

▷ sumator pe 4 biți