AC 11
2.5.

$x \to x'$ (1)

$x+y = -x'+y$

$x', y \geq 0$

X

1	1	1	0	0	1	1	1	1	1

x_e, x_m

Y

0	0	1	0	1	0	0	0	1

y_e, y_m

$z_e = 4$

Bcd $-10 \equiv +6$
$1010 \xrightarrow{C_2} 0110 = 6$

Step 3 if sign(X) ≠ sign(Y) Complement Y_m of 2
- only Y_m has 2's complementation capabilities

$-x'+y$ x' = Cof2 of X

$x+y \to x-y$

sign(X) = 1 (−)
sign(Y) = 0 (+) } ⇒ Complement Y_m

Y

0	0	1	0	1.	0	0	1

$Y_{m_{C2}} = 0.111$

Step 4 Align Y_m by RShift with |d| bits
⇒ if Y_m was complemented of 2 in Step 3, while RShifting, introduce bits of 1s into Y_m's msb
! preserve sticky bits: g, r, s

$Y_{m_{C2}} = 0.111$ $d = -2$

$Y_{m_{C2}d} = 1.101\ \underbrace{1}_{g}\ \underbrace{1}_{r}\ \underbrace{0}_{s}$

Step 5 Add the 2 significands
- if sign(X) = sign(Y)
  - preserve Cout, if generated
- if sign(X) ≠ sign(Y)
  - if no Cout was generated
    ⇒ Z_m is negative ⇒ it needs to be complemented of 2
  - if Cout was generated
    ⇒ Z_m is positive ⇒ Cout is ignored

$$X_m = 1.111$$

$$\begin{array}{rl} Y_{m_{2\,al}} = & 1.101 \mid \overset{g}{1}\ \overset{r}{1}\ \overset{s}{0} \\ \hline Z_m = \underset{\text{Cout}}{\cancel{1}} & 1.100 \mid 1\ 1\ 0 \end{array} \; +$$

Z_m is positive

Step 6 Prenormalization
- according to rules from 2.4.
  - → determine Z_{M_n}, update Z_E
- Exception checking:
  - if $Z_E == Z_{E_{max}}$ $(2^{3}-2=6)$ and Z_M requires 1-bit RShift ⟹ Overflow

    (3 = nr of bits of Z_E)
  - if $Z_E == Z_{E_{min}}$ (1) and Z_M requires 1 or more bits LShift ⟹ Underflow

$$Z_M = \overset{\text{cout}}{0}\,1.100 \mid \underset{g}{1}\ \underset{r}{1}\ \underset{s}{0}$$

⟹ Rule 2) → $Z_{M_n} = 1.10\underset{z_{0W}}{0}$

$Z_E = 4$

Step 7 Calculate R and S:
- according to rules from 2.4.

Rule 2) → $R = g = 1$

$S = r \text{ or } s = 1 + 0 = 1$

Step 8 Round the Z_{M_n}
- according to the selected IEEE 754 rounding mode.
- according to the rounding rules from 2.4.
- if rounding generates carry out ⟹ postnormalization

Postnormalization:
- Z_M^{*} is RShifted by 1 bit
- Z_E ++

Prenormalization → exception checking (like in Step 6) (2)

if $z_E == e_{max}$ (=6) and z_M^* requires 1-bit RShift

⇒ Overflow

- Consider the round to nearest even mode.

if (R and (S or z_{0M})) then $z_M + 1$

$R = 1, \quad S = 1, \quad z_{0M} = 0.$

$R \cdot (S + z_{0M}) = 1 \cdot (1 + 0) = 1 \Rightarrow z_M + 1$

$$
\begin{array}{rl}
z_M = & 1.100 + \\
 & \quad\;\;\, 1 \\
\hline
z_M^* = & 1.101
\end{array}
$$

⇒ no carry generated

⇒ z_E not modified.

Step 9 Calculate sign of result.

- if sign(X) == sign(Y) ⇒ sign(Z) = sign(X)
- if sign(X) ≠ sign(Y) ⇒

	Swap (Step 2)	Complement (Step 5)	sign(X)	sign(Y)	sign(Z)
$Y_E > X_E$	YES		+	−	−
	YES		−	+	+
$X_E \geq Y_E$	NO	YES	+	−	−
	NO	YES	−	+	+
	NO	NO	+	−	+
	NO	NO	−	+	−

Swap (Step 2) : YES ⇒ sign(Z) = sign(Y) → before the swap

sign(Z) = 1 (−)

Step 10 Pack the result

$$
Z: \; \boxed{1 \,|\, 1 \,|\, 0 \,|\, 0 \,|\, .1 \,|\, 0 \,|\, 1}
$$

sign(Z) z_E

Verify the result:

$$X = 0.5625$$

$$Y = -3.75$$

$$X+Y = -3.1875 \text{ (infinite precision)}$$

$sign = 1$ $\qquad bias = 2^{e-1} - 1 = 3$

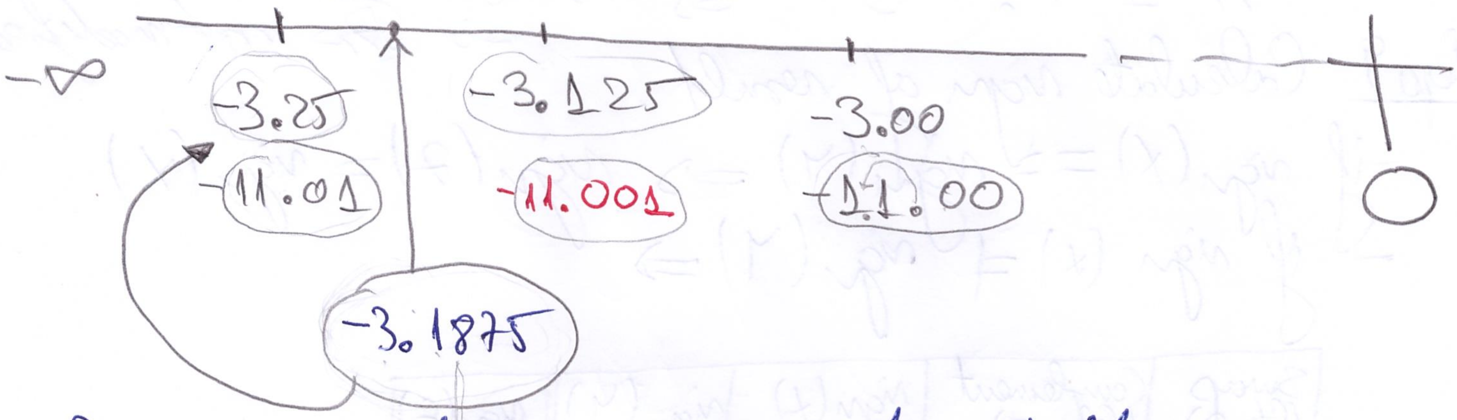
z: | 1 | 1 | 0 | 0 | . 1 | 0 | 1 | $\qquad z_E = 100 = 4$

$$z_M^* = .101$$

$$z = (-1)^{sign} * 2^{z_E - bias} * (1.z_M^*)$$

$$= (-1) * 2^{4-3} * (1.101_{(2)}) = -2 * 1.101_{(2)}$$

$$= -11.01_{(2)} = \underline{-3.25}$$



Design of the renormalization shifter:

- covers Step 6 and Step 7
  ⇒ according to the rules from 2.4.
- purely combinational.

From $\underline{\text{Step 5}}_{cout}$

$$z_M = z_4\, z_3 \,.\, z_2\, z_1\, z_0 \mid g\; r\; s$$

Output of Step 6 and Step 7

$$z_{M_n} = 1 \,.\, z_{2_n}\, z_{1_n}\, z_{0_n} \mid R\; S$$

Normalization cases:

A) z_n is normalized — (l/r_0)
B) z_n needs 1-bit LShift — (l_1)
C) z_n needs 2-bit LShift — (l_2)
D) z_n needs 3-bit LShift — (l_3)
E) z_n need 1-bit RShift — (r_1)

$z_{n_m} =$					
$1.\,z_{2_m}$	z_{1_m}	z_{0_m}	R	S	
$1.\,z_2$	z_1	z_0	g	(r or s)	
$1.\,z_1$	z_0	g	r	s	
$1.\,z_0$	g	0	0	0	
$1.\,g$	0	0	0	0	
$1.\,z_3$	z_2	z_1	z_0	(g or r or s)	

Associate a boolean variable to each of the 5 cases

$$z_{2m} = z_2 \cdot l/r_0 + z_1 \cdot l_1 + z_0 \cdot l_2 + g \cdot l_3 + z_3 \cdot r_1$$

$$z_{1m} = z_1 \cdot l/r_0 + z_0 \cdot l_1 + g \cdot l_2 + 0 \cdot l_3 + z_2 \cdot r_1$$

$$z_{0m} = z_0 \cdot l/r_0 + g \cdot l_1 + z_1 \cdot r_1$$

$$R = g \cdot l/r_0 + r \cdot l_1 + z_0 \cdot r_1$$

$$S = (r \text{ or } s) \cdot l/r_0 + s \cdot l_1 + (g \text{ or } r \text{ or } s) \cdot r_1$$

Variables l/r_0, l_1, l_2, l_3 and r_1:

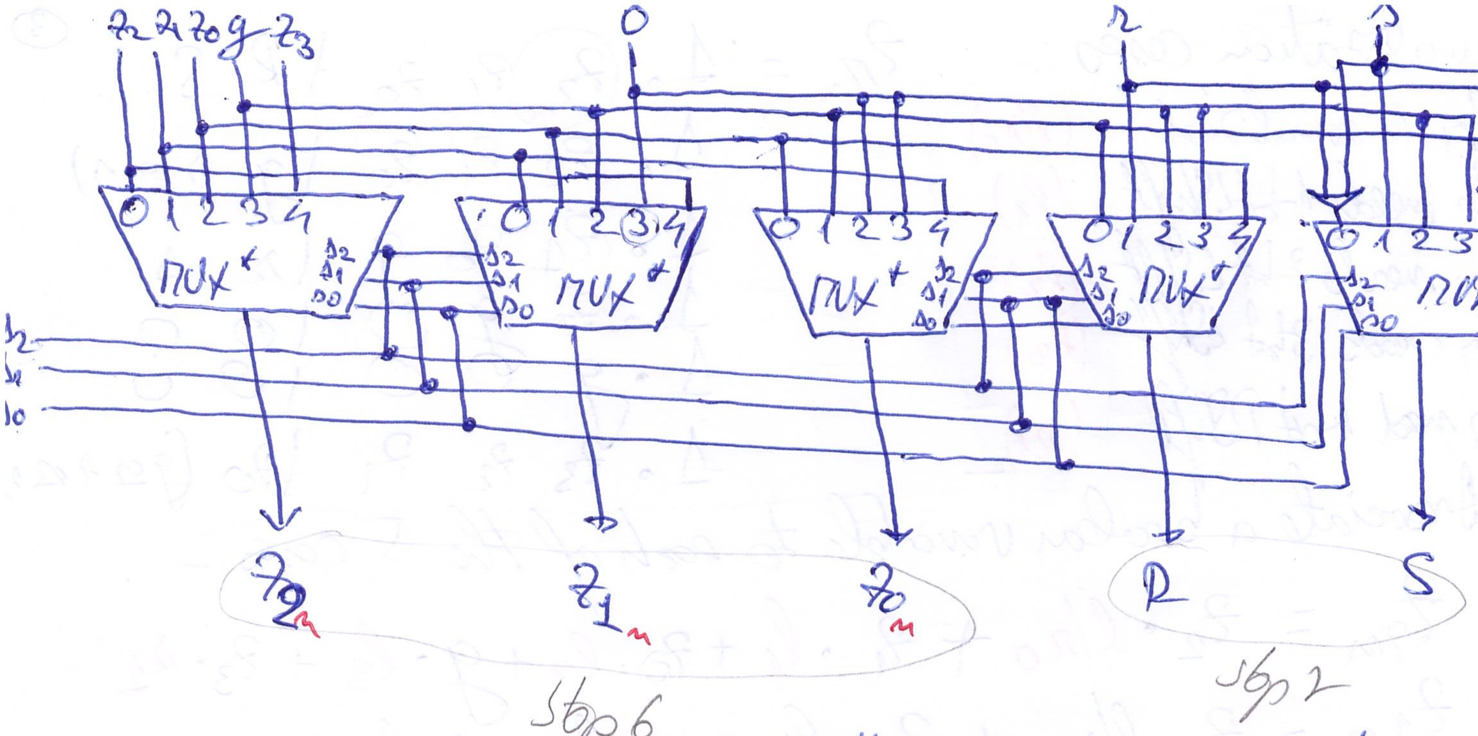
- only one can be active for a given operands pair

⇒ encode the 5 variables on the minimum nr of bits.

$\lceil \log_2 5 \rceil = 3$

→ use 3 signals s_2, s_1, s_0

Inputs					Outputs		
r_1	l_3	l_2	l_1	l/r_0	s_2	s_1	s_0
0	0	0	0	1	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	1	0	0	0	0	1	1
1	0	0	0	0	1	0*	0*



MUX^* = multiplexer with "degenerate" inputs

Chapter III Functional Analysis & Synthesis of Binary Multiplication Devices

3.1. Multiplication methods

a) paper & pencil

X = multiplier
Y = multiplicand } 4-bit, unsigned

$$
\begin{array}{r l}
1100 & \text{—— } Y \\
1011 = x_3x_2x_1x_0 & \text{—— } X \\
\hline
1100 & \text{— } x_0 \cdot Y \cdot 2^0 \\
1100 & \text{—— } x_1 \cdot Y \cdot 2^1 \\
0000 & \text{—— } x_2 \cdot Y \cdot 2^2 \\
1100 & \text{—— } x_3 \cdot Y \cdot 2^3 \\
\hline
10000100 & \text{— } P = \sum_{i=0}^{3} x_i \cdot Y \cdot 2^i
\end{array}
$$

$$= 2^7 + 2^2 = 128 + 4 = 132$$

$$
\begin{array}{r}
12 \\
11 \\
\hline
12 \\
12 \\
\hline
132
\end{array}
$$