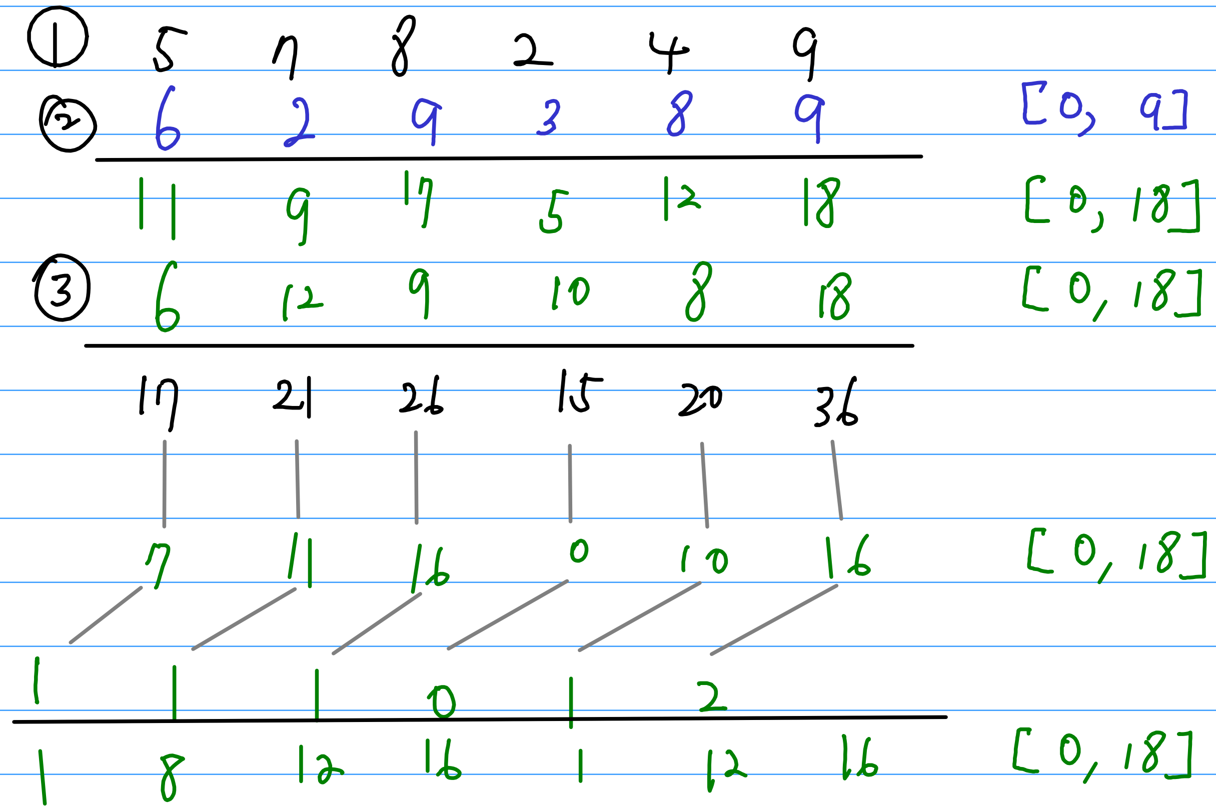
CORDIC Literature
Lookahead

Redundant Adders

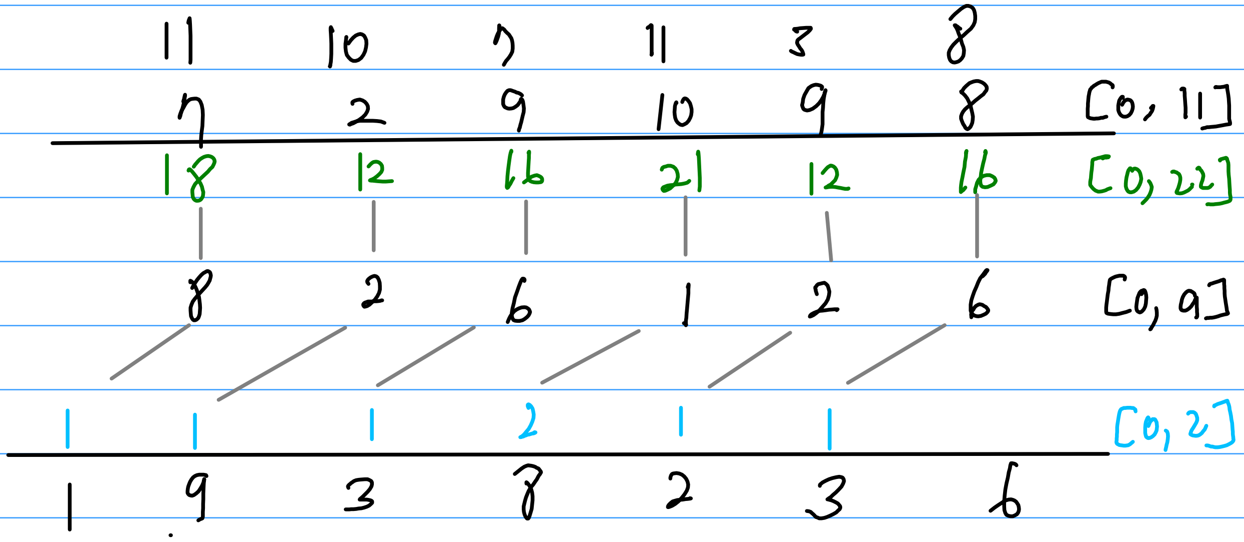
20160519

Based on

Computer Arithmetic : Algorithm and Hardware Designs
B. Parhami



adding radix-10 numbers
with the digit set [0, 18]



adding radix-10 numbers
with the digit set [0, 11]

① A 0 0 1 0 0 1 [0, 1]

② B 0 1 1 1 1 0 [0, 1]

0 1 2 1 1 1 [0, 2]

③ C 0 1 1 1 0 1

0 2 3 2 1 2 [0, 3]

0 0 1 0 1. 0 [0, 1]

0 1 1 1 0 1 [0, 1]

1 1 2 0 2 0 [0, 2]

④ D 0 0 1 0 1 1 [0, 1]

1 1 3 0 3 1 [0, 3]

1 1 1 0 1. 1 [0, 1]

0 0 1 0 1 0 [0, 1]

1 2 1 1 1 1 [0, 2]

Digit Sets and Digit Set Conversion ①

radix 10 number
with the digit set
[0, 18]

→

radix 10 number
with the digit set
[0, 9]

	11	9	17	10	12	18	[0, 18]
	11	9	17	10	13	8	
	11	9	17	11	3	8	↓ ⑩
	11	9	18	1	3	8	
	11	10	8	1	3	8	
	12	0	8	1	3	8	
1	2	0	8	1	3	8	[0, 9]

Digit Sets and Digit Set Conversion ②

radix 2 number with the digit set [0, 2] → radix 2 number with the digit set [0, 1]

	1	1	2	0	②	0	[0, 2]	
	1	1	2	①	0	0	↓	
	1	1	②	1	0	0		
	1	②	0	1	0	0		②
	②	0	0	1	0	0		
1	0	0	0	1	0	0	[0, 1]	

Digit Sets and Digit Set Conversion ③

radix 10 number with the digit set [0, 18] → radix 10 number with the digit set [-6, 5]

asymmetric

	11	9	17	10	12	⑱	[0, 18]
	11	9	17	10	⑭	-2	
	11	9	17	⑪	4	-2	-7 ↓ ⑥
	11	9	⑱	1	4	-2	
	11	⑪	-2	1	4	-2	
	⑫	1	-2	1	4	-2	
1	2	1	-2	1	4	-2	[-6, 5]

⑱ = 10 + 8 = 20 − 2 → carry 2, -2.

⑭ = 10 + 4 → carry 1, 4

⑪ = 10 + 1 → carry 1, 1

⑱ = 20 − 2 → carry 2, -2

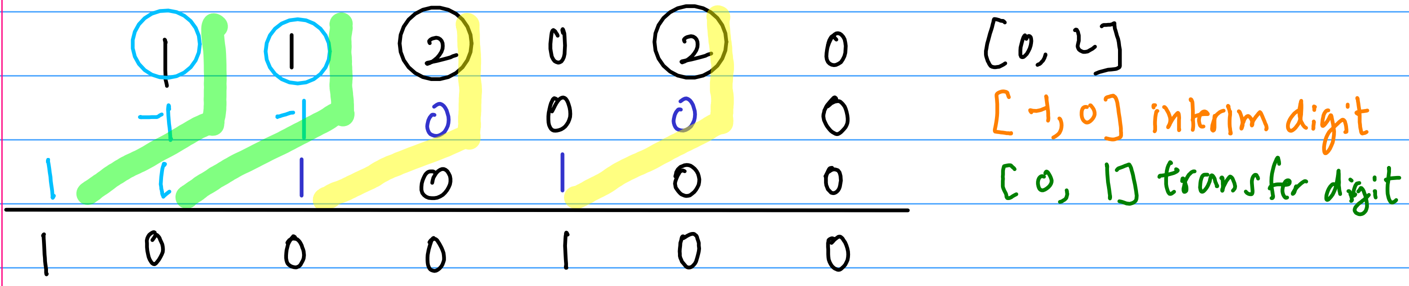
⑪ = 10 + 1 → carry 1, 1

⑫ = 10 + 2 → carry 1, 2

Digit Sets and Digit Set Conversion ④

radix 2 number with the digit set [0, 1] → radix 2 number with the digit set [-1, 1]

- digit-serial conversion, as before
- carry-free conversion, this case



	1	1	2	0	2	0	[0, 2]
	-1	-1	0	0	0	0	[-1, 0] interim digit
1	1	1	0	1	0	0	[0, 1] transfer digit
1	0	0	0	1	0	0	

$2 = 2 + 0$ carry 1, 0

$1 = 2 - 1$ carry 1, -1

resulting interim digit [-1, 0] can absorb incoming carry of 1 without further progation.

Binary Signed Digit (BSD)

radix 2 number with the digit set [-1, 1]

many possible encodings

	2^4	2^3	2^2	2^1	2^0	
x_i	1	-1	0	-1	0	16-8-2=6
(s, v)	01	11	00	11	00	Sign-Value
2's	01	11	00	11	00	2-bit 2's
(n, p)	01	10	00	10	00	neg-pos flags
(n, z, p)	001	100	010	100	010	1-out-of-3

Hybrid Signed Digit

trade off { algorithmic speed
implementation cost

example BSD digits in every third position

	BSD	B	B	BSD	B	B	BSD	B	B		
Ⓐ	1	0	1	-1	0	1	-1	0	1		
Ⓑ	0	1	1	-1	0	0	0	1	0		
	1	1	2	-2	0	1	-1	1	1	p_i	position sum
	-1			0			-1			w_i	interim sum
1			-1			0				t_{in}	transfer
1	-1	1	1	0	0	1	-1	1	1		

Generalized Signed Digit (GSD)

① OSD (Ordinary Signed Digit)

Auizienis 1961

Signed digit number
with symmetric digit sets $[-\alpha, \alpha]$
radix $r > 2$

α any integer $\lfloor r/2 \rfloor + 1 \leqslant \alpha \leqslant r - 1$

$2 \lfloor r/2 \rfloor + 3$ digit values

$\alpha < r < 2\alpha$

② GSD (Generalized Signed Digit)

asymmetric digit sets $[\underset{\uparrow}{-\alpha}, \underset{\uparrow}{\beta}]$

Ordinary Signed Digit

* Avizienis 1961

Signed-digit Number Representation for
Fast Parallel Arithmetic

integer radix r

in each digit q values are allowed

$$r+2 \leqslant q \leqslant 2r-1$$

both pos & neg

redundancy → a method of fast add/Sub

Each sum/diff digit depends on
only two adjacent digital positions
of the operands

Totally-parallel addition & subtraction

requirements

totally-parallel

: the minimum redundancy $(r+2) \le q$

unique zero representation

the upper bound $q \le 2r-1$

the magnitude of the allowed digit values $\leq \boxed{r-1}$

Signed-digit meets these requirements $r > 2$

digit : both pos & neg values
contain sign info ⇒ Signed-digit
no special sign digit

$[x_{n-1}, \cdots, x_0]$ $\qquad$ $x_i \in \{0, 1, \cdots, \beta-1\}$

$\{0, 1, \cdots, \boxed{r-1}\}$

$$\langle [x_{n-1}, \cdots, x_0] \rangle_\beta = \sum_{i=0}^{n-1} x_i \cdot \beta^i$$

$$= \sum_{i=0}^{n-1} x_i r^i$$

digit sets $\{-D, \cdots, +D\}$

$\{-\alpha, \cdots, +\alpha\}$

Theorem 1 Uniqueness of Division with Remainder

$x, y \in \mathbb{Z}$ $\quad y \neq 0$

unique numbers $q, r \in \mathbb{Z}$

$x = q \cdot y + r \qquad 0 \leq r < |y|$

q : x div y
r : x mod y

Lemma 1 SD Number Representations

$$x = \langle [x_{n-1}, \cdots, x_1, x_0] \rangle_{D,B}$$
$$= \langle [x'_{n-1}, \cdots, x'_1, x'_0] \rangle_{D,B}$$

$\Rightarrow$ $x_0' = x_0 + k \cdot B \qquad k \in \mathbb{Z}$

$y_1 \quad \langle [\boxed{x_{n-1}, \cdots, x_1}, x_0] \rangle_{D,B}$

$y_1' \quad \langle [\boxed{x'_{n-1}, \cdots, x'_1}, x'_0] \rangle_{D,B}$

$$x = y_1 \cdot B + x_0 = y_1' \cdot B + x_0'$$

$$x_0 - x_0' = (y_1' - y_1) B = k \cdot B$$

redundant representation of a number

not possible to reduce equality testing to checking the equality of the corresponding digits

$$\begin{array}{ccccccc} \langle [& x_{n-1}, & \cdots, & x_1, & x_0 &] \rangle_{D,B} \\ & \| & & \| & \| & \\ \langle [& x'_{n-1}, & \cdots, & x'_1, & x'_0 &] \rangle_{D,B} \end{array} \qquad \times$$

constant depth reduction possible

$$x = y \iff x - y = 0$$

possible to reduce equality testing to checking whether the result is zero

to be able to check equality of SD numbers

$D < B$ $\qquad$ $\alpha < r$

Theorem 2 Unique Representation of Zero

The number 0 has a unique representation

as SD number $\langle [x_{n-1}, \cdots, x_1, x_0] \rangle_{D,B}$

iff $D < B$ $\boxed{\alpha < r}$

$x = \langle [x_{n-1}, \cdots, x_1, x_0] \rangle_{D\ B} = 0$ ($x_i = 0$ for any n)

if any other representation

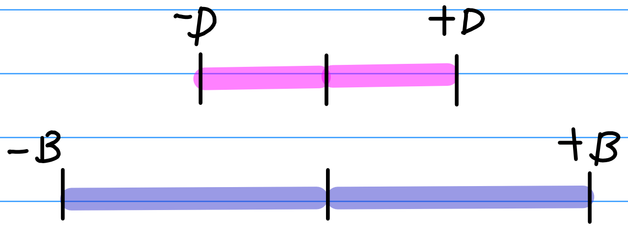
$\langle [x'_{n-1}, \cdots, x'_1, x'_0] \rangle_{D\ B} = 0$

$x'_0 \neq x_0$

then we would have $x'_0 - x_0 = x'_0 = k \cdot B$ $(k \neq 0)$

this cannot happen iff $x'_0 \in [-D, +D]$

$\subset [-B, +B]$



x_0 is uniquely determined for $x = 0$ iff $D < B$

similarly x_1 $\langle [x_{n-1}, \cdots, x_1 \quad] \rangle_{D\ B} = 0$

Schneider & Willenbucher

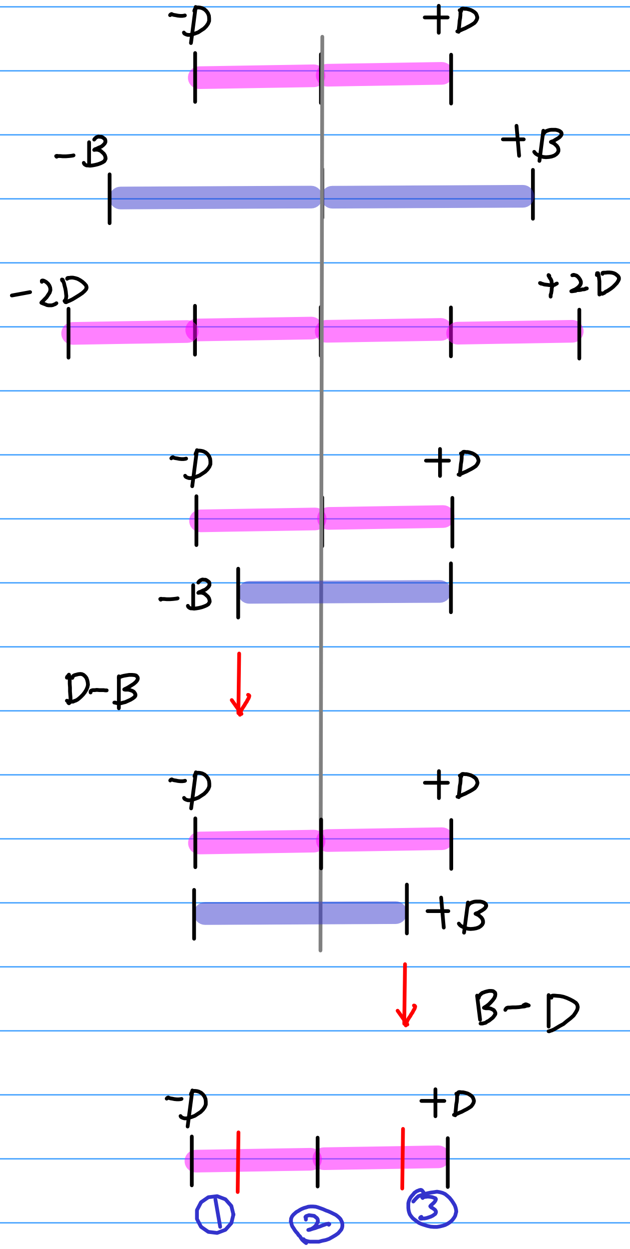
$\langle [1, -B] \rangle_{D,B} = B - B = 0$

$\langle [-1, +B] \rangle_{D,B} = -B + B = 0$

$\leftarrow \boxed{D = B}$

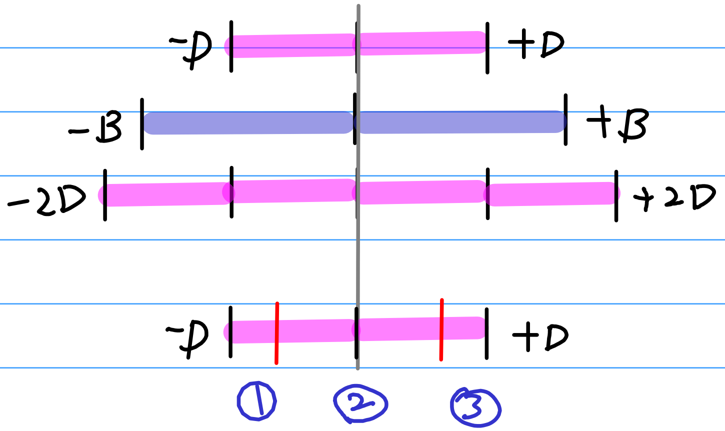
not unique

Assume $\boxed{B < 2 \cdot D}$ $\quad B - D \leq D$



Schneider & Willenbucher

$-D \cdots D-B$	$D-B+1 \cdots B-D-1$	$B-D \cdots +D$
①	②	③



-D ··· D-B	D-B+1 ··· B-D-1	B-D ··· +D
①	②	③

the digits $D_{-1} = [-D, D-B]$

$D_{+1} = [B-D, +D]$ +B -B

adding B / subtracting B

increment / decrement the next digit x_{i+1}

$D_0 = [D-B+1, B-D-1]$

uniquely determined

D_{-1}, D_{+1} exactly one alternative

$$\langle [x_{n-1}, \cdots, x_0] \rangle_{D,B} = \sum_{i=0}^{n-1} x_i B^i$$

$x_i \in \{0, 1, \cdots, B-1\}$ radix B

$x_i \in \{-D, \cdots, +D\}$ redundant radix B

$D < B$ ← $(+, -)$, unique zero

$B < 2 \cdot D$ ← to ensure reduntancy

$$\boxed{D < B < 2 \cdot D}$$

$$\langle [x_{n-1}, \cdots, x_0] \rangle_{D,B} = \sum_{i=0}^{n-1} x_i r^i$$

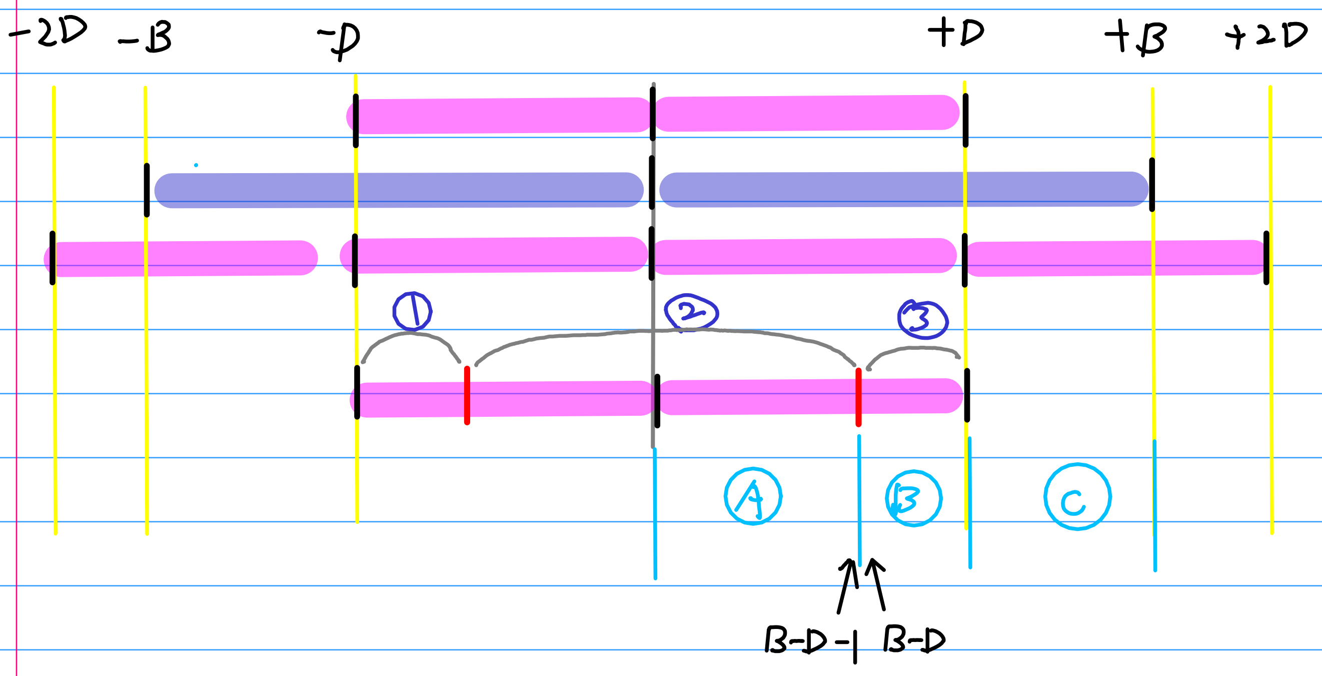
$x_i \in \{0, 1, \cdots, r-1\}$ radix r

$x_i \in \{-\alpha, \cdots, +\alpha\}$ redundant radix r

Lemma 2 for any sp num

$x = \langle [x_{n-1}, \dots, x_0] \rangle_{D,B}$

$D < B < 2 \cdot D$



-D ⋯ D-B	D-B+1 ⋯ B-D-1	B-D ⋯ +D
①	②	③

Ⓐ $(x \bmod B) \in \{0, \dots, B-D-1\}$

unique $x_0 = (x \bmod B)$

Ⓑ $(x \bmod B) \in \{B-D, \dots, D\}$

two $\begin{cases} x_0 = (x \bmod B) \\ x_0 = (x \bmod B) - B \end{cases}$

Ⓒ $(x \bmod B) \in \{D+1, \dots, B\}$

unique $x_0 = (x \bmod B) - B$

$$\begin{array}{l} x = \langle [x_n, \dots, x_0] \rangle_{D,B} \\ y = \langle [y_n, \dots, y_0] \rangle_{D,B} \\ \hline u = \langle [u_n, \dots, u_0] \rangle_{D,B} \end{array}$$

$$-2\cdot D \leq \boxed{u_i = x_i + y_i} \leq 2\cdot D$$

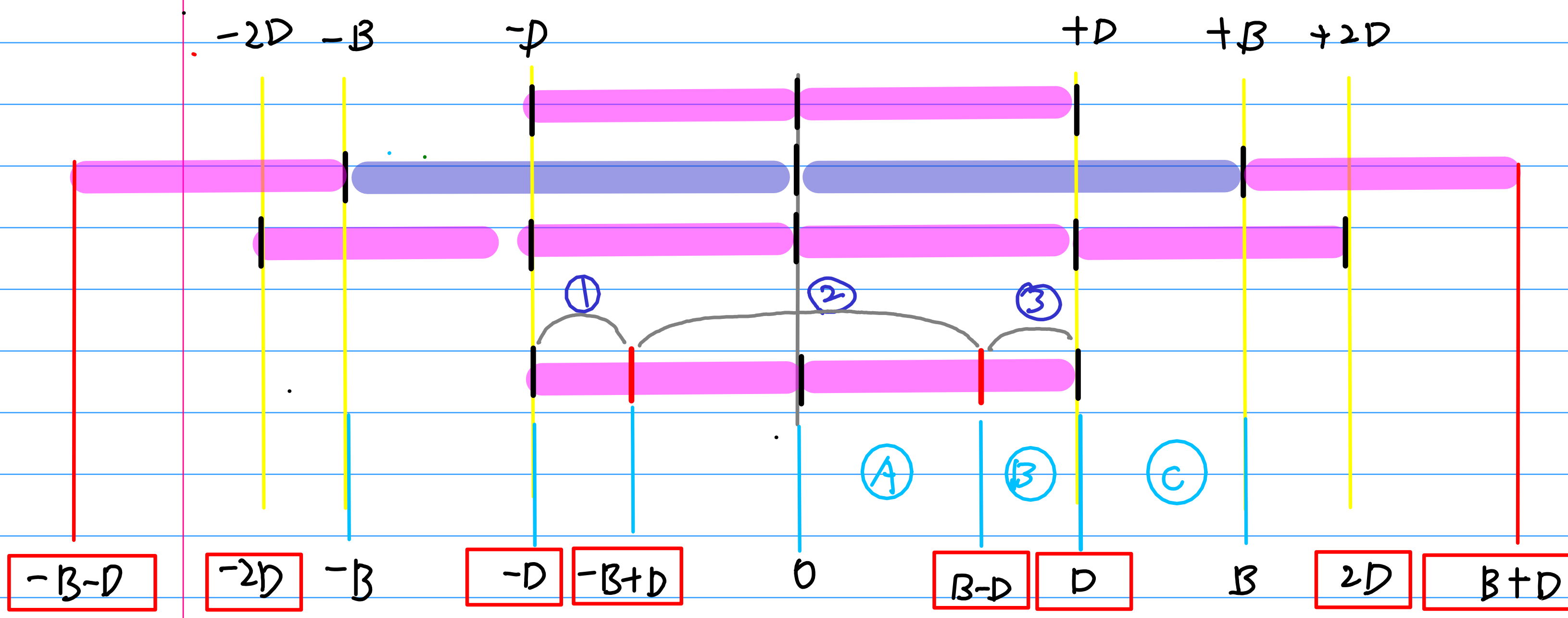
$$\Big\downarrow \text{decompose}$$

$$\underline{-2\cdot B} < -2\cdot D \leq \boxed{t_{i+1}\cdot B + w_i} \leq 2\cdot D < \underline{2\cdot B}$$

$$(-2\cdot B,\ 2\cdot B) \rightarrow t_{i+1} \in \{-1, 0, +1\}$$

$$\boxed{D < B < 2\cdot D}$$

$$\boxed{D < B < 2\cdot D}$$



$$\boxed{D < B < 2\cdot D} \Rightarrow$$

$$\boxed{-B-D} < \boxed{-2D} < \boxed{-D} \le \boxed{-B+D} < \boxed{0} < \boxed{B-D} \le \boxed{D} < \boxed{2D} < \boxed{B+D}$$

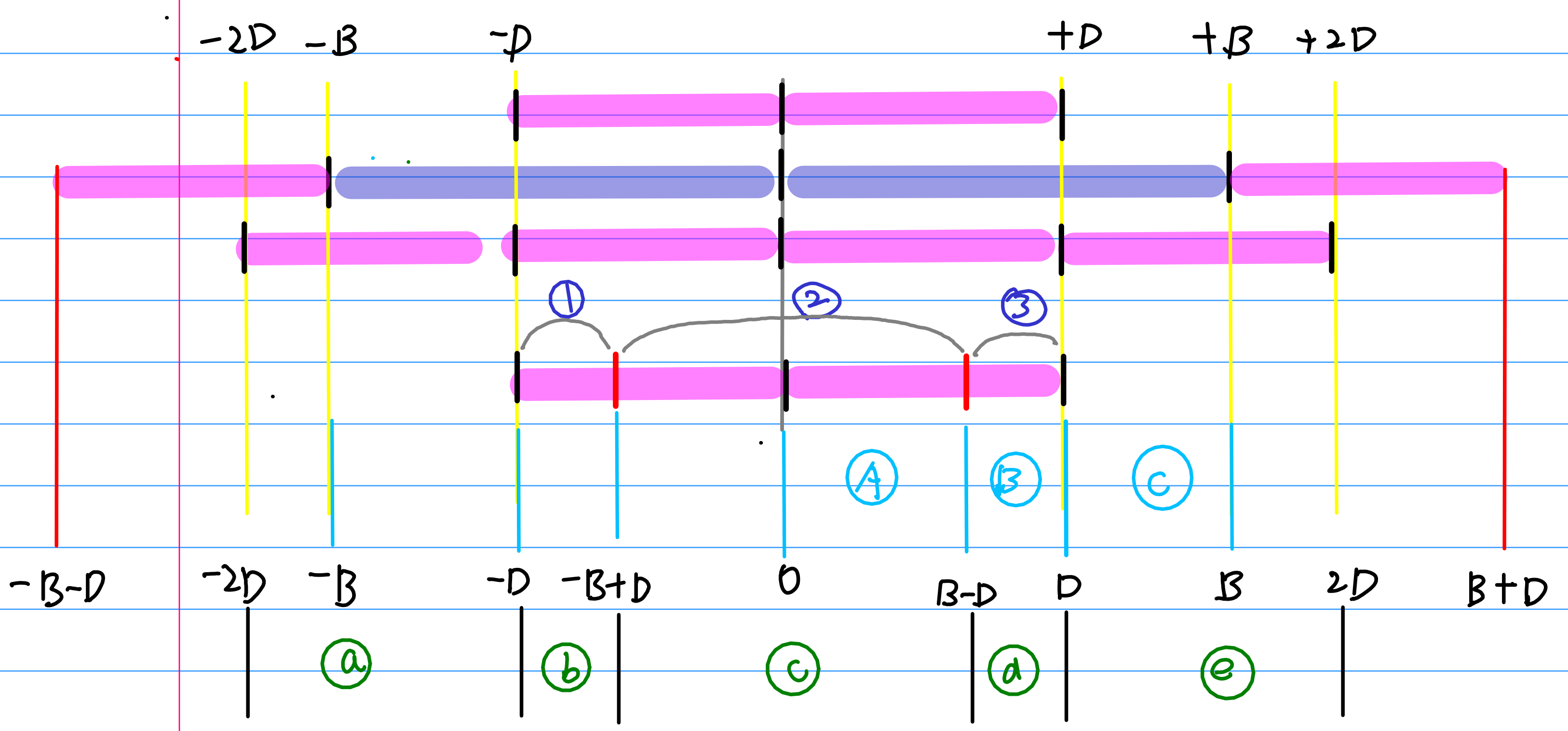
Lemma 3

for given digits

$$\boxed{u_i = x_i + y_i} \xrightarrow{\text{decompose}} \boxed{t_{i+1}\cdot B + w_i}$$

$x_i, y_i \in \{-D, \cdots, +D\}$ $w_i \in \{-D, \cdots, +D\}$

$D < B < 2\cdot D$ $t_i \in \{-1, 0, +1\}$

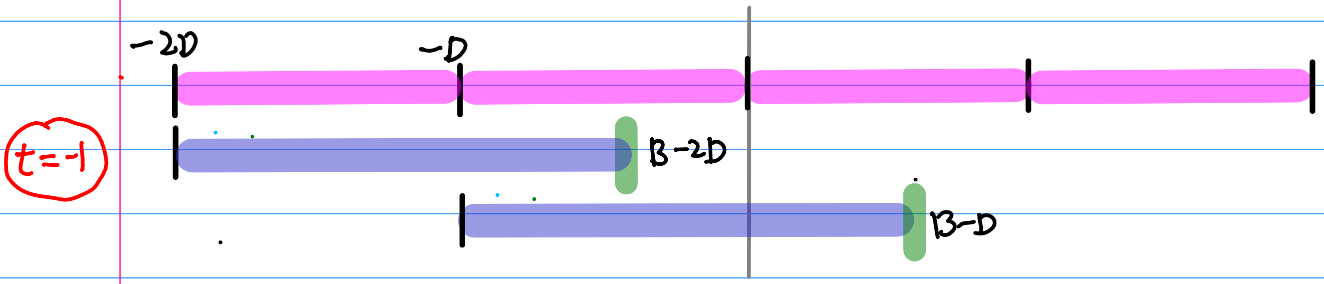


$[-2D, -D)$ $[-D, -B+D]$ $(-B+D, B-D)$ $[B-D, D]$ $(D, 2D]$

ⓐ	$[-2D, -D)$	$[-B+B-2D, -B+B-D) = [B-2D, B-D) \subset (-D, D)$
ⓑ	$[-D, -B+D]$	$[-B+B-D, -B+B-B+D] = [B-D, D] \subset (-D, D]$
	$[-D, -B+D]$	$[-D, -B+D] \subset [-D, D)$
ⓒ	$(-B+D, B-D)$	$(-B+D, B-D) \subset (-D, D)$
ⓓ	$[B-D, D]$	$[B-D, D] \subset (-D, D]$
	$[B-D, D]$	$[+B-B+B-D, +B-B+D] = [-D, -B+D] \subset [-D, D)$
ⓔ	$(D, 2D]$	$(+B-B+D, +B-B+2D] = (-B+D, -B+2D] \subset (-D, D)$

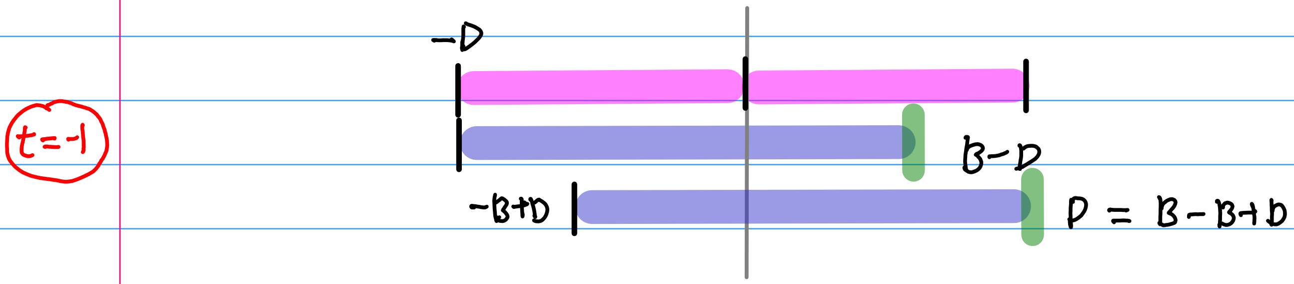
$$\boxed{D < B < 2 \cdot D}$$

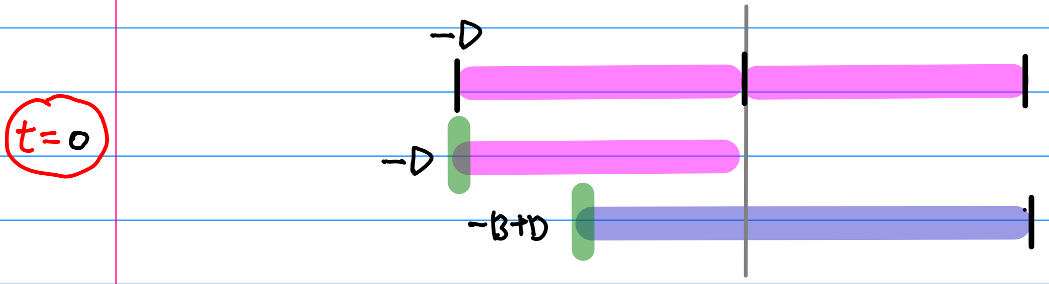
ⓐ $[-2D, -D) = [-B+B-2D, -B+B-D) = [B-2D, B-D) \subset (-D, D)$



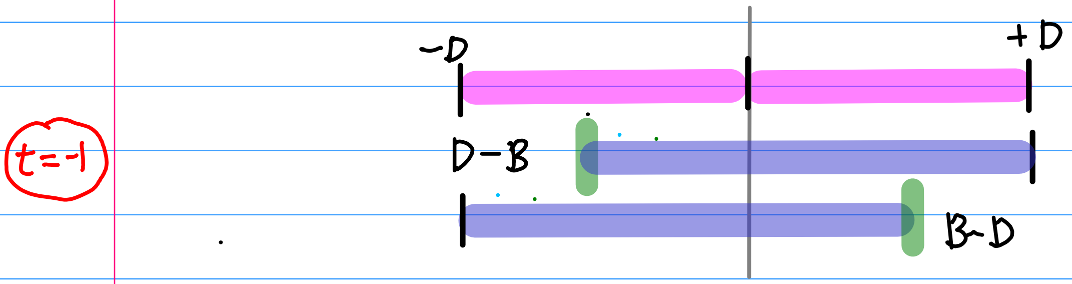
ⓑ $[-D, -B+D] = [-B+B-D, -B+B \quad = [B-D, D] \subset (-D, D]$

$[-D, -B+D] \quad [-D, -B+D] \subset [-D, D)$





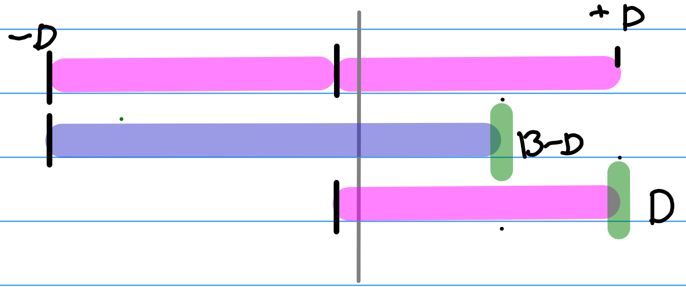
ⓒ $(-B+D, B-D) = \quad (-B+D, B-D) \subset (-D, D)$



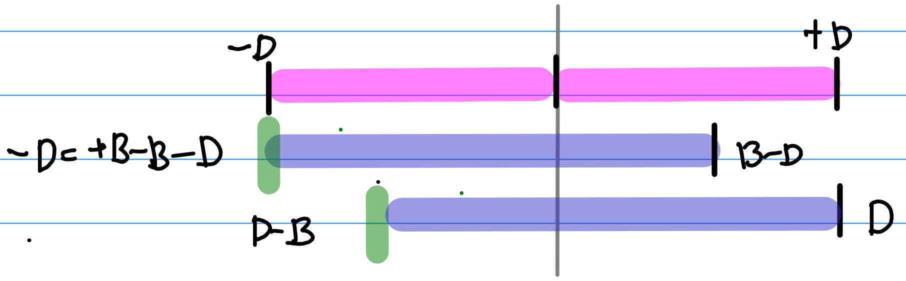
ⓓ $[B-D, D]$ $[B-D, D] \subset (-D, D]$

$[B-D, D] = [+B-B+B-D, +B-B+D] = [-D, -B+D] \subset [-D, D)$

$t=0$

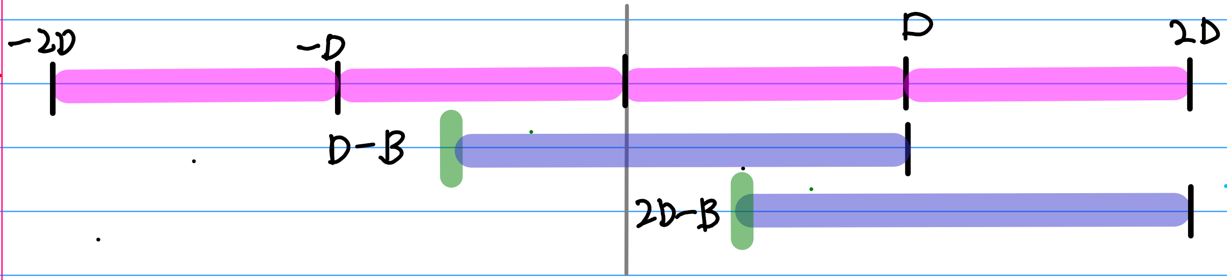


$t=+1$



ⓔ $(D, 2D] = (+B-B+D, +B-B+2D] = (-B+D, -B+2D] \subset (-D, D)$

$t=+1$



Lemma 3

for given digits

$$\boxed{u_i = x_i + y_i} \xrightarrow{\text{decompose}} \boxed{t_{i+1}\cdot B + w_i}$$

$x_i, y_i \in \{-D, \dots, +D\}$ $\qquad$ $w_i \in \{-D, \dots, +D\}$

$D < B \le 2\cdot D$ $\qquad$ $t_i \in \{-1, 0, +1\}$

To make sure $S_i = w_i + t_i$ (transfer + interim sum) will not produce a carry, Avizienis uses more strict requirements

Lemma 4

for given digits

$$\boxed{u_i = x_i + y_i} \xrightarrow{\text{decompose}} \boxed{t_{i+1}\cdot B + w_i}$$

$x_i, y_i \in \{-D, \dots, +D\}$ $\qquad$ $w_i \in \{\boxed{-D+1}, \dots, \boxed{+D-1}\}$

$D < B < 2\cdot D$ $\qquad$ $t_i \in \{-1, 0, +1\}$

ⓑ $\boxed{\begin{matrix}[-D, -B+D] \\ [-D, -B+D]\end{matrix}}$

$[-B+B-D, -B+B-B+D] = [B-D, D] \subset (-D, D]$

$[-D, -B+D] \subset [-D, D)$

ⓓ $\boxed{\begin{matrix}[B-D, D] \\ [B-D, D]\end{matrix}}$

$[B-D, D] \subset (-D, D]$

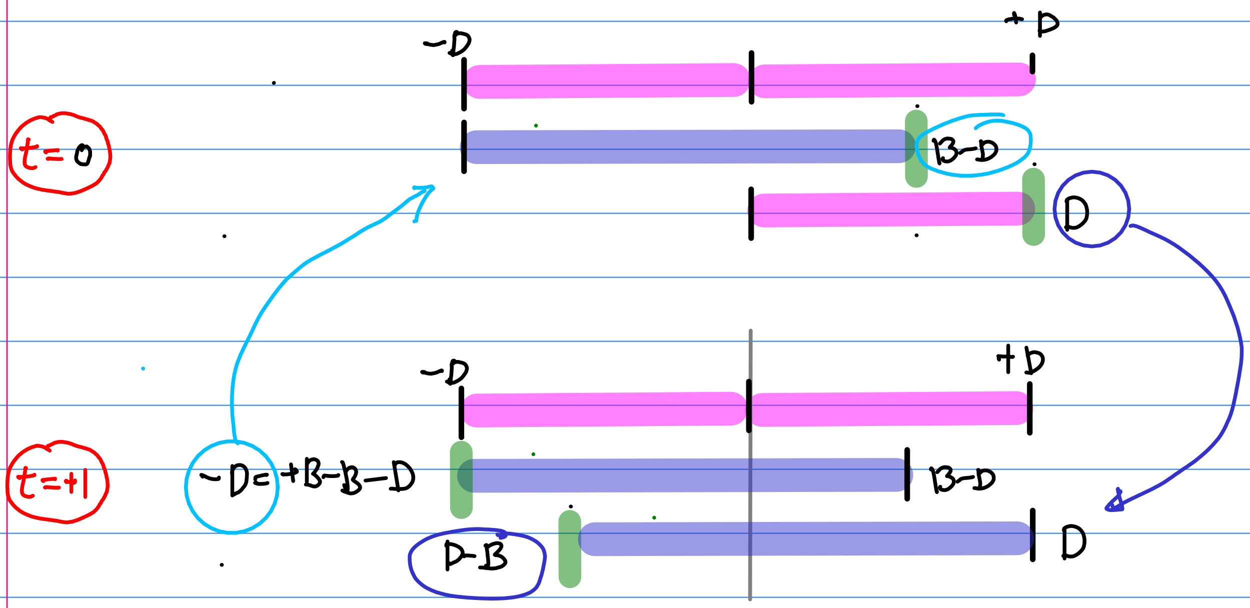
$[+B-B+B-D, +B-B+D] = [-D, -B+D] \subset [-D, D)$

2 different decompositions

each decomposition either starts $[-D$
or ends $+D]$

the interim sum $w_i \notin (-D, +D)$

mutually exclusive cases



for each case, there is one u_i
that produces an interim sum $w_i \notin (-D, +D)$

In that case, however,
we use the other possible decomposition
and therefore ensure $w_i \in (-D, +D)$

ⓑ

$$\boxed{\begin{array}{l}[-D, -B+D] \\ [-D, -B+D]\end{array}} \qquad \begin{array}{l}[B-D, D] \\ [-D, -B+D]\end{array} \qquad \begin{array}{l}\subset (-D, D] \\ \subset [-D, D)\end{array}$$

if $w_i = D$, use $-B+D$
if $w_i = -D$, use $B-D$

ⓐ

$$\boxed{\begin{array}{l}[B-D, D] \\ [B-D, D]\end{array}} \qquad \begin{array}{l}[B-D, D] \\ [-D, D-B]\end{array} \qquad \begin{array}{l}\subset (-D, D] \\ \subset [-D, D)\end{array}$$

if $w_i = D$, use $-B+D$
if $w_i = -D$, use $B-D$

Theorem 3 Carry Free Addition by Avizienis

The addition of SD numbers $x = \langle [x_{n-1}, \dots, x_0] \rangle_{D,B}$

$y = \langle [y_{n-1}, \dots, y_0] \rangle_{D,B}$

$D < B < 2 \cdot D$

can be computed in depth $O(1)$ with $O(n)$ work (gates)

1) for $i \in \{0, \dots, n-1\}$ compute u_i

$$u_i = x_i + y_i$$

2) for $i \in \{0, \dots, n-1\}$ compute t_{i+1}

$$t_{i+1} = \begin{cases} +1 & \text{if } u_i \geqslant +D \\ -1 & \text{if } u_i \leqslant -D \\ 0 & \text{if } -D < u_i < +D \end{cases}$$

3) for $i \in \{0, \dots, n-1\}$ compute s_i

$$s_i = t_i + \boxed{w_i} \qquad (t_0 = 0)$$

$$= t_i + \boxed{u_i - t_{i+1} \cdot B}$$

ⓑ $[-D, -B+D]$ $\quad [-B+B-D, -B+B-B+D] = [B-D, D] \subset (-D, D]$

$[-D, -B+D]$ $\quad [-D, -B+D] \subset [-D, D)$

ⓓ $[B-D, D]$ $\quad [B-D, D] \subset (-D, D]$

$[B-D, D]$ $\quad [+B-B+B-D, +B-B+D] = [-D, -B+D] \subset [-D, D)$

$u_i \in [-D, -B+D]$ or $[B-D, D]$

prefers the decomposition with $t_{i+1} = 0$
except for the case $u_i = \pm D$

$u_i \in (-D, -B+D]$ or $[B-D, D)$

$t_{i+1} = 0 \quad W_i = (-D, -B+D]$ or $[B-D, D)$

$u_i = -D$

$t_{i+1} = +1 \quad W_i = B-D$

$u_i = +D$

$t_{i+1} = -1 \quad W_i = -B+D$

① Subtraction of x and y $\qquad x+y$

addition of x and $-y = \langle [-y_{n-1}, \cdots, -y_0] \rangle_{D,B}$

depth $O(1)$ work $O(n)^2$

② Checking equality of x and y $\qquad x == y$

Check $x - y = 0$

subtraction depth $O(1)$ and work $O(n)$

checking all digits are zero depth $O(\log n)$ and work $O(n)$

③ Comparing $x < y$ $\qquad x < y$

Testing $x - y < 0$

subtraction depth $O(1)$ and work $O(n)$

checking signs depth $O(\log n)$ and work $O(n)$

Some of the leading digits can be zero

the sign of <u>the first</u> <u>non-zero digit</u> determines the sign

④ **Multiplication** x and y $x * y$

determined by adding partial products $x \cdot y_i \cdot B^i$

can be arranged with a depth $O(\log(n))$

work $O(n^2)$

⑤ **Division** x / y

multiplication of the integer reciprocal

a depth $O(\log(n))$ work $O(n^2)$

Binary SD Numbers

Avezenis already noted that
his algorithm does not work for binary SD numbers

using weaker constraints

$D < B \leqslant 2 \cdot D \qquad 1 < B \leqslant 2$

ⓐ	$[-2D, -D)$	$[-B+B-2D, -B+B-D) = [B-2D, B-\cdot\cdot) \subset (-D, D)$
ⓑ	$[-D, -B+D]$ $[-D, -B+D]$	$[-B+B-D, -B+B-B+D] = [B-D, \;;] \subset (-D, D]$ $[-D, -B+D] \subset [-D, D)$
ⓒ	$(-B+D, B-D)$	$(-B+D, B-D) \subset (-D, D)$
ⓓ	$[B-D, D]$ $[B-D, D]$	$[B-D, D] \subset (-D, D]$ $[+B-B+B-D, +B-B+D] = [-D, -B+D] \subset [-D, D)$
ⓔ	$(D, 2D]$	$(+B-B+D, +B-B+2D] = (-B+D, -B+2D] \subset (-D, D)$

ⓐ	$[-2D, -D)$	$[-2, -1) = -2 = u_i$	$t_{i+1} = -1$	$w_i = 0$
ⓑ	$[-D, -B+D]$	$[-1, -1] = -1 = u_i$	$t_{i+1} = \left(\begin{array}{l} 0 \\ -1 \end{array} \right.$	$w_i = \left(\begin{array}{l} -1 \\ +1 \end{array} \right.$
ⓒ	$(-B+D, B-D)$	$(-1, 1) = 0 = u_i$	$t_{i+1} = 0$	$w_i = 0$
ⓓ	$[B-D, D]$	$[1, 1] = +1 = u_i$	$t_{i+1} = \left(\begin{array}{l} 0 \\ +1 \end{array} \right.$	$w_i = \left(\begin{array}{l} +1 \\ -1 \end{array} \right.$
ⓔ	$(D, 2D]$	$(1, 2] = +2 = u_i$	$t_{i+1} = 1$	$w_i = 0$

$D < B \leq 2 \cdot D$

$D = 1 \quad B = 2$

$u_i = -2$	$(t_{i+1}, w_i) = (-1, 0)$
$u_i = -1$	$(t_{i+1}, w_i) = (0, -1)$ or $(-1, +1)$
$u_i = 0$	$(t_{i+1}, w_i) = (0, 0)$
$u_i = +1$	$(t_{i+1}, w_i) = (0, +1)$ or $(+1, -1)$
$u_i = +2$	$(t_{i+1}, w_i) = (+1, 0)$

no decomposition that always allows
$w_i \in [-D+1, +D-1] = 0$

→ generally accepted that there is
no carry free addition
for general binary SD numbers

One possible solution
consider a radix $B = 2^k$
represent digits x_i
as 2's complement numbers with $(k+1)$ bits

adding 2's complement numbers with k-bit
depth is increased to $O(\log(k))$

Jaberipur & Ghodsi
"High Radix Signed Digit Number System"
Scientia Iranica 2003

Since the small number k can be chosen
→ may still be a practical solution

Asymmetric digits sets

Gorgin & **Jaberipur**
"A family of high radix signed digit adders"
Symposium on Computer Arithmetic

Parhami: Carry-free addition of recorded Signed digit numbers (1988)

recording

a given binary SD number x of length n
to an equivalent SD number x' of length $n+1$

Such that there are no two neighboring digits

$$x'_{i+1} \cdot x'_{i} = \underline{1}$$

the output of this addition
does not satisfy this condition

⇒ record again before another addition

not increase area
but increase latency

Addition of digits z_i and y_i is totally-parallel if the following two conditions are met

1) the sum digit s_i : i-th digit of sum $s = z + y$ is the function only of z_i & y_i and the transfer digit t_i from the $(i+1)$th position on the right $s_i = f(z_i, y_i, t_i)$

2; the transfer digit t_{i-1} to the $(i\ 1)$ th position on the left is a function of z_i & y_i

$$t_{i-1} = f(z_i, y_i)$$

Totally-parallel subtraction of $z_i - x_i$ is performed as the totally-parallel addition of the additive inverse of y_i

$$z_i - y_i = z_i + (-y_i)$$

addition of two digits performed in 2 successive steps

① the outgoing transfer digit t_{i-1} and
an interim sum digit w_i

$$z_i + y_i = r t_{i-1} + w_i .$$

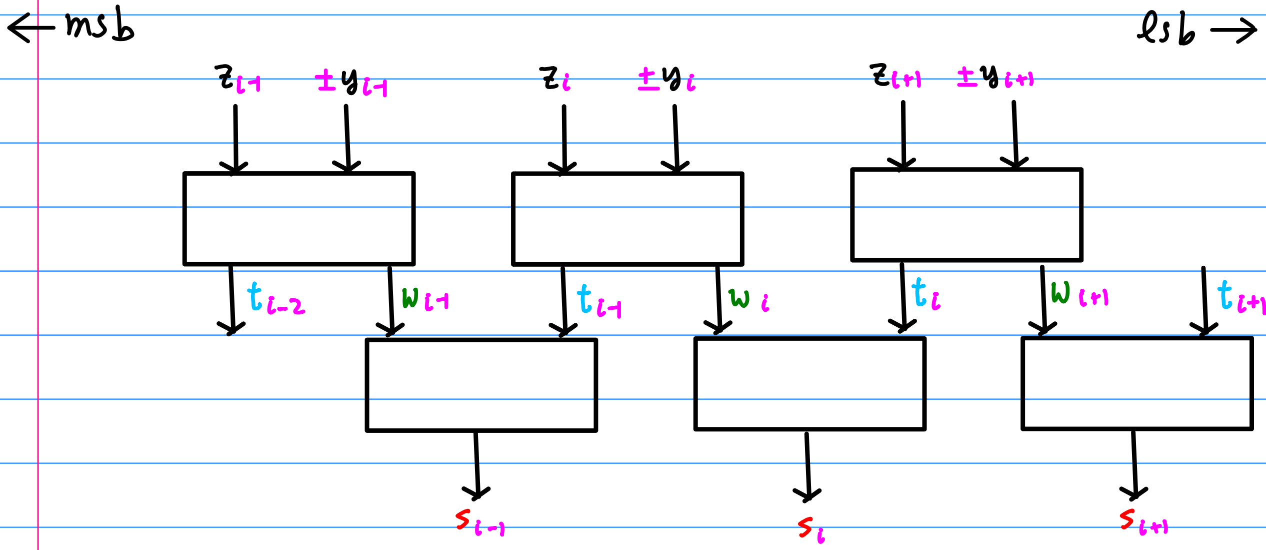
the sum digit $s_i = w_i + t_i .$

the range values are the same
for s_i, z_i, y_i

$$Z = \sum_{i=-n}^{m} z_i \, r^{-i}$$

msb lsb

$$z_{-n} r^{n} + z_{-n+1} r^{n-1} + \cdots + z_{-1} r^{1} + z_0 + z_1 r^{-1} + \cdots + z_m r^{-m}$$



Generalized Signed Digit

Signed digit number
with symmetric digit sets $[-\alpha, \alpha]$
radix $r > 2$

α any integer
$\lfloor r/2 \rfloor + 1 \leq \alpha \leq r - 1$

$2 \lfloor r/2 \rfloor + 3 \leq$ digit values

$r = 3$ $\lfloor 3/2 \rfloor + 1 \leq \alpha \leq 3 - 1$ $2 \leq \alpha \leq 2$ $\alpha = 2$

$[-2, 2]$ 5 digit values

$r = 4$ $\lfloor 4/2 \rfloor + 1 \leq \alpha \leq 4 - 1$ $3 \leq \alpha \leq 3$ $\alpha = 3$

$[-3, 3]$ 7 digit values

$r = 5$ $\lfloor 5/2 \rfloor + 1 \leq \alpha \leq 5 - 1$ $3 \leq \alpha \leq 4$ $\alpha = 3, 4$

$[-3, 3]$ 7 digit values
$[-4, 4]$ 9 digit values

$r = 6$ $\lfloor 6/2 \rfloor + 1 \leq \alpha \leq 6 - 1$ $4 \leq \alpha \leq 5$ $\alpha = 4, 5$

$[-4, 4]$ 9 digit values
$[-5, 5]$ 11 digit values

Carry Free Addition Algorithms

Ⓐ Carry Free Addition Algorithms for GSD numbers

① compute the <u>position sums</u> $\boxed{p_i} = x_i + y_i$

② decompose p_i into a <u>transfer</u> $\boxed{t_i}$ and
an <u>interim sum</u> $\boxed{w_i} = p_i - r\,t_{i+1}$

③ add the incoming transfers
to obtain the <u>sum digit</u> $\boxed{s_i} = w_i + t_i$

the transfer digits $t_i \in [-\lambda, \mu]$

no new transfer in the last step

$$
\begin{array}{rcccl}
-\alpha & \le & p_i & \le & \beta \\
-\lambda & \le & r\,t_{i+1} & \le & \mu \\
\hline
-\alpha-\mu & \le & \boxed{p_i - r\,t_{i+1}} & \le & \beta+\lambda
\end{array}
$$

$\boxed{w_i}$: interim sum

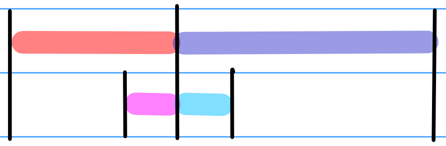
$$\underline{-\alpha+\lambda} \le p_i - r\,t_{i+1} \le \underline{\beta-\mu}$$

the smallest w_i
if a transfer of $-\lambda$
is to be absorbed

the largest w_i
if a transfer of $+\mu$
is to be absorbed

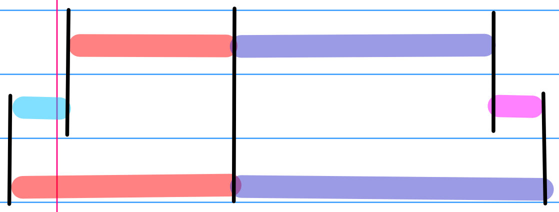
the transfer digits $t_i \in [-\lambda, \mu]$

no new transfer in the last step



$$-\alpha \le p_i \le \beta$$
$$-\lambda \le r t_{i+1} \le \mu$$

☆ for the given $-\lambda$, μ
the range of $\boxed{w_i}$ interim sum

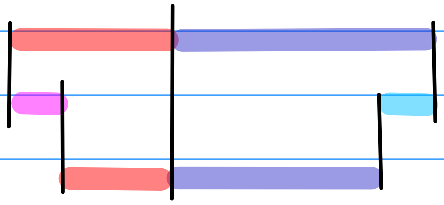


$$\begin{array}{rcccl} -\alpha & \le & p_i & \le & \beta \\ -\mu & \le & -r t_{i+1} & \le & \lambda \\ \hline -\alpha-\mu & \le & \boxed{p_i - r t_{i+1}} & \le & \beta+\lambda \end{array}$$

We need the following case for carry free addition

☆ Conditions for no new transfer in $\boxed{s_i} = w_i + t_i$

$\Downarrow$



$$\underline{-\alpha+\lambda} \le \boxed{p_i - r t_{i+1}} \le \underline{\beta-\mu}$$

the smallest w_i
if a transfer of $-\lambda$
is to be absorbed

the largest w_i
if a transfer of $+\mu$
is to be absorbed.

$$-\alpha \leq p_i \leq \beta$$
$$-\lambda \leq r t_{i+1} \leq \mu$$

$$\underline{-\alpha + \lambda} \leq \boxed{p_i - r t_{i+1}} \leq \underline{\beta - \mu}$$

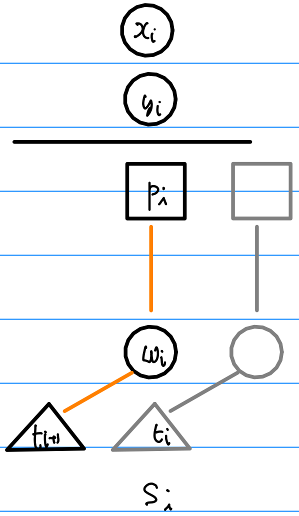
$$\lambda \geqslant \frac{\alpha}{(r-1)}$$
$$\mu \geqslant \frac{\beta}{(r-1)}$$

$(r-1)\lambda \geqslant \alpha$ $(r1)\mu \geqslant \beta$

$[-\alpha, \beta]$
$[-\lambda, -\mu]$

$p_i = x_i + y_i$: position sum
$= r t_{i+1} + w_i$: decompose transfer and interim sum

$s_i = w_i + t_i$



$-\alpha \leq x_i, y_i \leq \beta$

$-2\alpha \leq p_i \leq 2\beta$

$-\alpha \leq s_i \leq \beta$

$-\alpha \leqslant \boxed{w_i} + t_i \leq \beta$

$-\alpha \leqslant \boxed{p_i - r t_{i+1}} + t_i \leq \beta$

$-\alpha - p_i - t_i \leq -r t_{i+1} \leq \beta - p_i - t_i$

$$\frac{-\beta + p_i + t_i}{r} \leqslant t_{i+1} \leqslant \frac{\alpha + p_i + t_i}{r}$$

$$\frac{p_i - (\beta - t_i)}{r} \leqslant t_{i+1} \leqslant \frac{p_i + (\alpha + t_i)}{r}$$

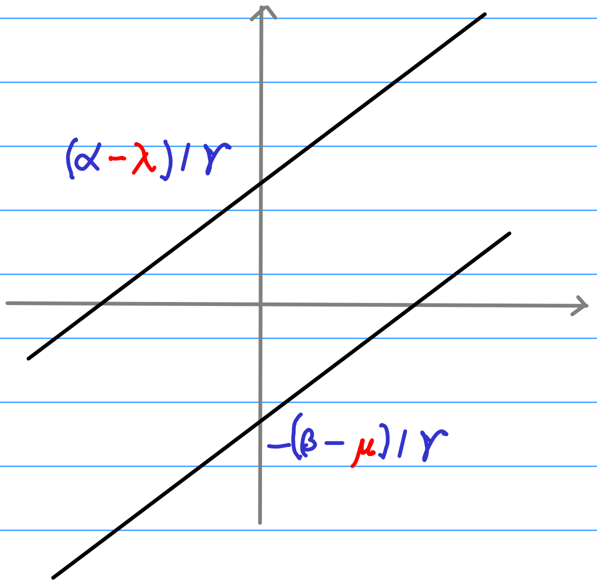
$$\frac{p_i - (\beta - t_i)}{r} \leqslant t_{i+1} \leqslant \frac{p_i + (\alpha + t_i)}{r} \qquad -\lambda \leqslant t_i \leqslant \mu$$

$$\frac{p_i - (\beta - \mu)}{r} \leqslant t_{i+1} \leqslant \frac{p_i + (\alpha - \lambda)}{r} \qquad \lambda \leq \alpha, \quad \mu \leq \beta$$

the worst case : smallest intersection

$$\underset{-\alpha}{[} \quad x_i, y_i \quad \underset{\beta}{]}$$

$$\underset{-\lambda}{[} \, t_j \, \underset{\mu}{]}$$



Comparison constants C_k $\quad -\lambda \le k \le \mu$

$$kr-(\alpha-\lambda) \leqslant C_k \leqslant (k-1)r+\beta-\mu+1$$

$$\frac{p_i-(\beta-\mu)}{r} \leqslant t_{i+1} \leqslant \frac{p_i+(\alpha-\lambda)}{r}$$

$t_{i+1} = k-1 \qquad p_i \leqslant (k-1)r+\beta-\mu$

$$k-1 \Leftarrow \frac{p_i-(\beta-\mu)}{r} = \frac{(k-1)r+\beta-\mu-(\beta-\mu)}{r}$$

$t_{i+1} = k \qquad p_i \leqslant kr-(\alpha-\lambda)$

$$k \Leftarrow \frac{p_i+(\alpha-\lambda)}{r} = \frac{kr-(\alpha-\lambda)+(\alpha-\lambda)}{r}$$