Redundant CORDIC Timmermann (C)

20170208

Termination Algorithms
Modified CORDIC
CSD (Canonic Sign Digit) Encoding
Booth Encoding

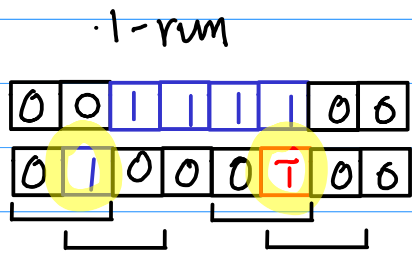
Low Latency Time CORDIC Algorithms - Timmermann (1992)
Redundant and on-line CORDIC - Ercegovac & Lang (1990)

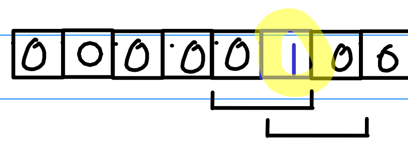
CSD (Canonic Signed Digit)

like Booth encoding (not modified Booth)

all non-zero digits are separated by zeros

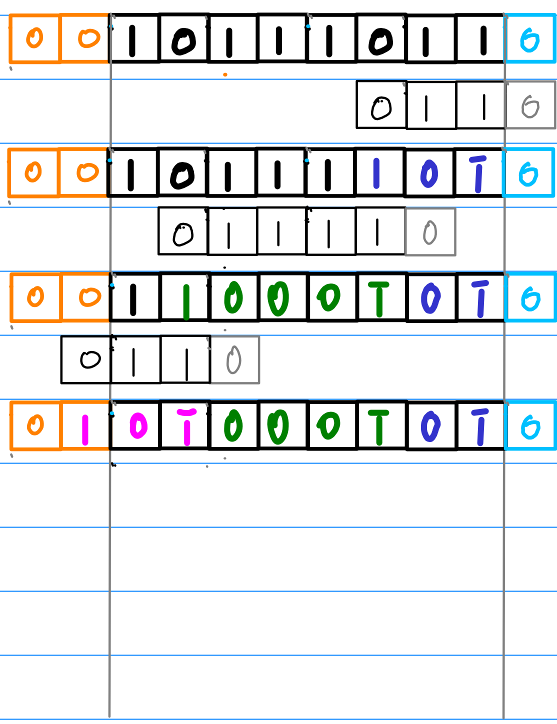
$\Rightarrow$ $\boxed{\sigma_i \sigma_{i+1} = 0}$





$0 \cdot 1 = 0$ $\qquad$ $0 \cdot \bar{1} = 0$

$1 \cdot 0 = 0$ $\qquad$ $\bar{1} \cdot 0 = 0$



Iterative Reduction of 1-runs

$\boxed{\sigma_i \sigma_{i+1}} = 0$

Unique encoding

Two successive iteration steps

$$\boxed{\begin{aligned} x_{i+1} &= x_i - m\,\sigma_i\, 2^{-S(m,i)}\, y_i \\ y_{i+1} &= y_i + \sigma_i\, 2^{-S(m,i)}\, x_i \\ z_{i+1} &= z_i - \sigma_i\, \alpha_{m,i} \end{aligned}}$$

$i+1$
$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & -m\,\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

$i+2$
$$\begin{bmatrix} x_{i+2} \\ y_{i+2} \end{bmatrix} = \begin{bmatrix} 1 & -m\,\sigma_{i+1} 2^{-i-1} \\ \sigma_{i+1} 2^{-i-1} & 1 \end{bmatrix} \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -m\,\sigma_{i+1} 2^{-i-1} \\ \sigma_{i+1} 2^{-i-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & -m\,\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

$$= \begin{bmatrix} 1 - m\,\sigma_{i+1}\sigma_i\, 2^{-2i-1} & -m\,\sigma_i 2^{-i} - m\,\sigma_{i+1} 2^{-i-1} \\ \sigma_{i+1} 2^{-i-1} + \sigma_i 2^{-i} & -m\,\sigma_{i+1}\sigma_i\, 2^{-2i-1} + 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

$$= \begin{bmatrix} 1 - m\,\sigma_i\sigma_{i+1}\, 2^{-2i-1} & -m(\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) \\ (\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) & 1 - m\,\sigma_i\sigma_{i+1}\, 2^{-2i-1} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

CSD

$$\begin{bmatrix} 1 - m\,\boxed{\sigma_i\sigma_{i+1}}\, 2^{-2i-1} & -m(\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) \\ (\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) & 1 - m\,\boxed{\sigma_i\sigma_{i+1}}\, 2^{-2i-1} \end{bmatrix}$$

$\boxed{\sigma_i \neq 0} \rightarrow \boxed{\sigma_i \sigma_{i+1} = 0}$ property of Booth encoding

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & -m\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

$$\begin{bmatrix} x_{i+2} \\ y_{i+2} \end{bmatrix} = \begin{bmatrix} 1 - m\boxed{\sigma_i \sigma_{i+1}} 2^{-2i-1} & -m(\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) \\ (\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) & 1 - m\boxed{\sigma_i \sigma_{i+1}} 2^{-2i-1} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} x_{i+2} \\ y_{i+2} \end{bmatrix} = \begin{bmatrix} 1 & -m(\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) \\ (\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

$$x_{i+2} = x_i - m(\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) y_i$$
$$y_{i+2} = (\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) x_i + y_i$$

CSD

$$\boxed{\begin{aligned} x_{i+2} &= x_i - m\sigma_i 2^{-i} \boxed{y_i} - m\sigma_{i+1} 2^{-i-1} \boxed{y_i} \\ y_{i+2} &= y_i + \sigma_i 2^{-i} \boxed{x_i} + \sigma_{i+1} 2^{-i-1} \boxed{x_i} \\ z_{i+2} &= z_i - \sigma_i \alpha_{m,i} - \sigma_{i+1} \alpha_{m,i+1} \end{aligned}}$$

$\boxed{\sigma_i = 0} \rightarrow$ inc/dec no rotation, but compensate the scale factor.

$$\boxed{\begin{aligned} x_{i+1} &= x_i + m \cdot 2^{-2i-1} x_i \\ y_{i+1} &= y_i + m \cdot 2^{-2i-1} y_i \\ z_{i+1} &= z_i \end{aligned}}$$

inc/dec

inc/dec

$m=+1$ / $m=-1$

$$\begin{aligned} x_{i+1} &= (1 + m \cdot 2^{-2i-1})\, x_i \\ y_{i+1} &= (1 + m \cdot 2^{-2i-1})\, y_i \\ z_{i+1} &= z_i \end{aligned}$$

$$\begin{aligned} x_{i+2} &= (1 + m \cdot 2^{-2i-3})\, x_{i+1} = (1 + m \cdot 2^{-2i-3})(1 + m \cdot 2^{-2i-1})\, x_i \\ y_{i+2} &= (1 + m \cdot 2^{-2i-3})\, y_{i+1} = (1 + m \cdot 2^{-2i-3})(1 + m \cdot 2^{-2i-1})\, y_i \\ z_{i+1} &= z_i \end{aligned}$$

$$2^{-2i-3} \cdot 2^{-2i-1} = 2^{-4i-4} \ll 1$$

$$\boxed{\begin{aligned} x_{i+2} &= (1 + m \cdot 2^{-2i-3} + m \cdot 2^{-2i-1})\, x_i \\ y_{i+2} &= (1 + m \cdot 2^{-2i-3} + m \cdot 2^{-2i-1})\, y_i \\ z_{i+1} &= z_i \end{aligned}}$$

$m = 1, \quad s(m, i) = i$

CSD

Cond (I) $0 \leqslant i \leqslant \frac{1}{4}(n-3)$

$$\begin{aligned} x_{i+1} &= x_i - \sigma_i 2^{-i} y_i \\ y_{i+1} &= y_i + \sigma_i 2^{-i} x_i \\ z_{i+1} &= z_i - \sigma_i \tan^{-1}(2^{-i}) \end{aligned}$$

$$\begin{aligned} x_{i+1} &= x_i - \sigma_i 2^{-i} y_i \\ y_{i+1} &= y_i + \sigma_i 2^{-i} x_i \\ z_{i+1} &= z_i - \sigma_i \tan^{-1}(2^{-i}) \end{aligned}$$

Cond (II) $\frac{1}{4}(n-3) < i \leqslant \frac{1}{2}(n+1)$

$\sigma_i \neq 0$

$$\begin{aligned} x_{i+1} &= x_i - \sigma_i 2^{-i} y_i \\ y_{i+1} &= y_i + \sigma_i 2^{-i} x_i \\ z_{i+1} &= z_i - \sigma_i \tan^{-1}(2^{-i}) \end{aligned}$$

$\sigma_i \neq 0$

$$\begin{aligned} x_{i+2} &= x_i - m\sigma_i 2^{-i} \boxed{y_i} - m\sigma_{i+1} 2^{-i-1} \boxed{y_i} \\ y_{i+2} &= y_i + \sigma_i 2^{-i} \boxed{x_i} + \sigma_{i+1} 2^{-i-1} \boxed{x_i} \\ z_{i+2} &= z_i - \sigma_i \alpha_{m,i} - \sigma_{i+1} \alpha_{m,i+1} \end{aligned}$$

$\sigma_i = 0$

$$\begin{aligned} x_{i+1} &= x_i + m \cdot 2^{-2i-1} x_i \\ y_{i+1} &= y_i + m \cdot 2^{-2i-1} y_i \\ z_{i+1} &= z_i \end{aligned}$$

$\sigma_i = 0$

$$\begin{aligned} x_{i+2} &= (1 + m \cdot 2^{-2i-3} + m \cdot 2^{-2i-1})\, x_i \\ y_{i+2} &= (1 + m \cdot 2^{-2i-3} + m \cdot 2^{-2i-1})\, y_i \\ z_{i+1} &= z_i \end{aligned}$$

Cond (III) $\frac{1}{2}(n+1) < i$

$\sigma_i \neq 0$

$$\begin{aligned} x_{i+1} &= x_i - \sigma_i 2^{-i} y_i \\ y_{i+1} &= y_i + \sigma_i 2^{-i} x_i \\ z_{i+1} &= z_i - \sigma_i \tan^{-1}(2^{-i}) \end{aligned}$$

$\sigma_i \neq 0$ or $\sigma_i = 0$

$$\begin{aligned} x_{i+2} &= x_i - m\sigma_i 2^{-i} \boxed{y_i} - m\sigma_{i+1} 2^{-i-1} \boxed{y_i} \\ y_{i+2} &= y_i + \sigma_i 2^{-i} \boxed{x_i} + \sigma_{i+1} 2^{-i-1} \boxed{x_i} \\ z_{i+2} &= z_i - \sigma_i \alpha_{m,i} - \sigma_{i+1} \alpha_{m,i+1} \end{aligned}$$

$\sigma_i = 0$

$$\begin{aligned} x_{i+1} &= x_i \\ y_{i+1} &= y_i \\ z_{i+1} &= z_i \end{aligned}$$

$i \leftarrow i+1$

$i \leftarrow i+2$

Low Latency Time CORDIC Algorithms - Timmerman (1992)

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & -m\,\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

$$\begin{bmatrix} x_{i+2} \\ y_{i+2} \end{bmatrix} = \begin{bmatrix} 1 - m\boxed{\sigma_i \sigma_{i+1}} 2^{-2i-1} & -m(\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) \\ (\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) & 1 - m\boxed{\sigma_i \sigma_{i+1}} 2^{-2i-1} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

$\boxed{\sigma_i = 0} \rightarrow \boxed{\sigma_i \sigma_{i+1} = 0}$

$$\begin{bmatrix} x_{i+2} \\ y_{i+2} \end{bmatrix} = \begin{bmatrix} 1 & -m(\cancel{\sigma_i 2^{-i}}^{\,0} + \sigma_{i+1} 2^{-i-1}) \\ (\cancel{\sigma_i 2^{-i}}^{\,0} + \sigma_{i+1} 2^{-i-1}) & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

$$x_{i+2} = x_i - m(\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1})\, y_i$$
$$y_{i+2} = (\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1})\, x_i + y_i$$

$$x_{i+2} = x_i - m\,\sigma_i 2^{-i} y_i - m\,\sigma_{i+1} 2^{-i-1} y_i$$
$$y_{i+2} = y_i + \sigma_i 2^{-i} x_i + \sigma_{i+1} 2^{-i-1} x_i$$

$\boxed{\sigma_i = 0}$ $\boxed{\sigma_{i+1} \neq 0}$

$$x_{i+2} = \boxed{x_i} \quad \boxed{- m\,\sigma_{i+1} 2^{-i-1} y_i}$$
$$y_{i+2} = \boxed{y_i} \quad \boxed{+ \sigma_{i+1} 2^{-i-1} x_i}$$

$\boxed{\sigma_i \neq 0}$ $\boxed{\sigma_{i+1} = 0}$

$$x_{i+2} = \boxed{x_i} \quad \boxed{- m\,\sigma_i 2^{-i} y_i}$$
$$y_{i+2} = \boxed{y_i} \quad \boxed{+ \sigma_i 2^{-i} x_i}$$

$\boxed{\sigma_i = 0}$ $\boxed{\sigma_{i+1} = 0}$

$$x_{i+2} = \boxed{x_i}$$
$$y_{i+2} = \boxed{y_i}$$

Cond (II) $\frac{1}{4}(n-3) < i \leq \frac{1}{2}(n+1)$

$\sigma_i \neq 0$

parallel $\leftarrow$

$$\boxed{\begin{aligned} x_{i+2} &= x_i - m\sigma_i 2^{-i}\boxed{y_i} - m\sigma_{i+1} 2^{-i-1}\boxed{y_i} \\ y_{i+2} &= y_i + \sigma_i 2^{-i}\boxed{x_i} + \sigma_{i+1} 2^{-i-1}\boxed{x_i} \\ z_{i+2} &= z_i - \sigma_i \alpha_{m,i} - \sigma_{i+1} \alpha_{m,i+1} \end{aligned}}$$

$\sigma_i = 0$

parallel $\leftarrow$

$$\boxed{\begin{aligned} x_{i+2} &= (1 + m \cdot 2^{-2i-3} + m \cdot 2^{-2i-1})\, x_i \\ y_{i+2} &= (1 + m \cdot 2^{-2i-3} + m \cdot 2^{-2i-1})\, y_i \\ z_{i+1} &= z_i \end{aligned}}$$

$\sigma_i = 0$

- postphone this computation until the whole iteration process has been completed
- Use a Wallace tree to implement this in parallel

$\sigma_i \neq 0$

- executes a rotation by either $\alpha_{m,i}$ or $\alpha_{m,i+1}$
- multiplex the different shifts, $\boxed{\text{4-to-2 cell}}$

Cond (iii) $\cdot\ \frac{1}{2}(n+1) < i$

- analogous to the termination algorithm
- but improved design regularity
  : tree structure are not necessary

- the original iterations are paired
  always two subsequent iterations are
  merged into a new single iteration

- a 4-to-2 adder cell (x_i, y_i)
  a 3-to-2 adder cell (z_i)
  a 3:1 multiplexer

σ_i	σ_{i+1}
0	0
0	1
1	0

Termination Algorithm

quit the iteration process as early as possible

termination algorithm
- T.C. Chen IBM Journal of Research and Development 1972
  Automatic computation of exponentials logarithms, ratios, and square roots

- Timmermann Modified CORDIC algorithms

the 2^{nd} half of the n iterations
- can be substituted by 2 multications in parallel

A fully parallel n-bit wallace tree multiplier
- $2 \cdot \log_2(n)$ FA time units

algorithm + prediction + termination
$$(n+1) + 2\log_2\left(\frac{n}{2}\right) + \log_2(n) = n + 3\log_2(n) - 1$$

Modified CORDIC

Modified CORDIC

Timmermann 1989 Electronics Letters

$$x_n = k_m \{ \quad x_0 \cos[\sqrt{(m)}\,\alpha] - \sqrt{(m)}\, y_0 \sin[\sqrt{(m)}\,\alpha] \}$$
$$y_n = k_m \{ 1/\sqrt{(m)}\, x_0 \sin[\sqrt{(m)}\,\alpha] + \quad y_0 \cos[\sqrt{(m)}\,\alpha] \}$$
$$z_n = z_0 + \alpha$$

k_m : the scaling factor

m : the coordinate system $(0, -1, +1)$

α : the rotation angle

$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}$: the initial values depends on the iteration goal

Data dependency across iteration

→ CSA no benefit

① 1st half iterations : the most significant contribution

the rotation angle $\boxed{\alpha_i = \frac{1}{\sqrt{(m)}} \tan^{-1}\left[\sqrt{(m)}\, 2^{-s(m,i)}\right]}$

$s(m, i)$ the iteration shift values

α_i decreases with the increasing iteration index i

② 2nd half iterations : can improve the accuracy only by one bit each

Cond Ⅰ $0 \leq i \leq \frac{1}{4}(n-3)$ } 1st half

Cond Ⅱ $\frac{1}{4}(n-3) < i \leq \frac{1}{2}(n+1)$

Cond Ⅲ $\frac{1}{2}(n+1) < i$ } 2nd half

$\boxed{\text{rotation} \quad z_n \to 0}$

$$x_n = k_m \{ x_0 \cos[\sqrt{(m)}\,\alpha] - \sqrt{(m)}\, y_0 \sin[\sqrt{(m)}\,\alpha] \}$$
$$y_n = k_m \{ 1/\sqrt{(m)}\, x_0 \sin[\sqrt{(m)}\,\alpha] + y_0 \cos[\sqrt{(m)}\,\alpha] \}$$

$\boxed{\text{Vectoring} \quad y_n \to 0}$

$$x_n = k_m \sqrt{x_0^2 + m\, y_0^2}$$
$$z_n = z_0 + 1/\sqrt{(m)} \tan^{-1}[\sqrt{(m)}\, y_0 / x_0]$$

2nd half iterations : can improve the accuracy
only by one bit each

replace these iterations by a single rotation
after the remaining rotation angle
has been reduced using a fixed number of
pure CORDIC iterations

this **truncation** reduces the latency time and saves area
although the truncation requires extra hardware

the necessary minimum number of iterations

Rotation mode ($z \to 0$)

after j CORDIC rotations have been performed (x_j, y_j) the z path contains the remaining rotation angle z_j

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} \cos[\sqrt{(m)}\, z_j] & -\sqrt{(m)}\, \sin[\sqrt{(m)}\, z_j] \\ 1/\sqrt{(m)}\, \sin[\sqrt{(m)}\, z_j] & \cos[\sqrt{(m)}\, z_j] \end{bmatrix} \begin{bmatrix} x_j \\ y_j \end{bmatrix}$$

$$x_n = k_m \{ x_0 \cos[\sqrt{(m)}\, \alpha] - \sqrt{(m)}\, y_0 \sin[\sqrt{(m)}\, \alpha] \}$$
$$y_n = k_m \{ 1/\sqrt{(m)}\, x_0 \sin[\sqrt{(m)}\, \alpha] + y_0 \cos[\sqrt{(m)}\, \alpha] \}$$
$$z_n = z_0 + \alpha$$

assume $k_m = 1$ $\leftarrow$ 2nd half iteration does not affect scaling factors

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} \cos[\sqrt{(m)}\, z_j] & -\sqrt{(m)}\, \sin[\sqrt{(m)}\, z_j] \\ 1/\sqrt{(m)}\, \sin[\sqrt{(m)}\, z_j] & \cos[\sqrt{(m)}\, z_j] \end{bmatrix} \begin{bmatrix} x_j \\ y_j \end{bmatrix}$$

Taylor Series expansions to $\sin\theta$, $\cos\theta$

take only the first terms

$$\sin\theta = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \cdots$$

$$\cos\theta = 1 - \frac{1}{2!}x^2 + \frac{1}{4!}x^4 - \frac{1}{6!}x^6 + \cdots$$

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & -\sqrt{(m)} \cdot \sqrt{(m)}\, z_j \\ 1/\sqrt{(m)} \cdot \sqrt{(m)}\, z_j & 1 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \end{bmatrix}$$

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & -m\, z_j \\ z_j & 1 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \end{bmatrix}$$

for a sufficiently small z_j

the required precision of n-bit

the upper limit on the maximal remainder

$$z_j \leqslant \frac{1}{\sqrt{(m)}} \tan^{-1}[\sqrt{(m)}\, 2^{-j+1}]$$

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & -m z_j \\ z_j & 1 \end{bmatrix} \begin{bmatrix} x_j \\ y_j \end{bmatrix}$$

for a sufficiently small z_j

the required precision of n-bit
the upper limit on the maximal remainder

$$\frac{1}{2} z_j^2 \leq 2^{-n} \rightarrow z_j^{\,2} \leq 2^{-n+1} \rightarrow z_j \leq 2^{\frac{-n+1}{2}}$$

$$z_j \leq \frac{1}{\sqrt{(m)}} \tan^{-1}\left[\sqrt{(m)}\, 2^{-j+1}\right]$$

$$j > \frac{n+1}{2}$$

Cond (I) $0 \leq i \leq \frac{1}{4}(n-3)$ } 1st half
Cond (II) $\frac{1}{4}(n-3) < i \leq \frac{1}{2}(n+1)$

Cond (III) $\frac{1}{2}(n+1) < i$ } 2nd half

Rotation mode

$$x_n = x_j - m\, z_j y_j \quad (j > (n+1)/2)$$
$$y_n = z_j x_j + y_j \quad (j > (n+1)/2)$$

Vectoring mode

$$x_n = z_j \qquad j > (n+1)/2$$
$$z_n = z_j + y_j / x_j \qquad j > (n/3) + 0.472$$

the prediction algorithm : rotation mode (OK)
vectoring mode (X)

2nd half of the n iterations in rotation mode
~ replaced by 2 multiplications in parallel

$$\begin{cases} z_j * x_j \\ z_j * y_j \end{cases}$$

A fully parallel n-bit Wallace tree multiplier : $2\log_2(n)$ FA time unit

prediction + termination.

the Truncated CORDIC Algorithm

- reduces the number of CORDIC iterations
- multiplication / division hardware
  - Booth Technique halves the amount of partial products
  - Carry Save Architecture

$k_m \neq 1 \Rightarrow$ multiplication $\Rightarrow$ multiplier anyway

Modified Booth Encoding CSD

Low Latency CORDIC IEEE Trans. on Computers Aug 1992

9 8	7 6	5 4	3 2	1 0

0	1	0	$\bar{1}$	0	0	0	$\bar{1}$	0	$\bar{1}$	0

each bit pair contains at least one zero

$\sigma_i \sigma_{i+1} = 0$ $\quad i, i+2, i+4, \cdots$

CSD property

✱ In Timmermann's paper, the modified Booth encoding refers to CSD (Canonic Signed Digit) $\boxed{\sigma_i \sigma_{i+1} = 0}$ not the generally known modified Booth encoding.

the algorithm depends on i

$$S(m, i) = i$$

$$\begin{cases} \lambda(t) = 1 & \text{for } |t| = 0 \\ \lambda(t) = 0 & \text{for } |t| = 1 \end{cases}$$

$$\begin{cases} \lambda(\sigma_i) = 1 & \text{for } |\sigma_i| = 0 \\ \lambda(\sigma_i) = 0 & \text{for } |\sigma_i| = 1 \end{cases} \qquad \begin{cases} \lambda(\sigma_{i+1}) = 1 & \text{for } |\sigma_i| = 0 \\ \lambda(\sigma_{i+1}) = 0 & \text{for } |\sigma_i| = 1 \end{cases}$$

σ_i	σ_{i+1}
0	0
0	1
1	0

$\lambda(\sigma_i)$	$\lambda(\sigma_{i+1})$
1	1
1	0
0	1

Simple Approximation $\boxed{\sigma_i \sigma_{i+1}} \Rightarrow 0$

$$\begin{bmatrix} x_{i+2} \\ y_{i+2} \end{bmatrix} = \begin{bmatrix} 1 - m\boxed{\sigma_i \sigma_{i+1}}2^{-2i-1} & -m(\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) \\ (\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) & 1 - m\boxed{\sigma_i \sigma_{i+1}}2^{-2i-1} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

$$\begin{bmatrix} x_{i+2} \\ y_{i+2} \end{bmatrix} = \begin{bmatrix} 1 & -m(\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) \\ (\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}) & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

$$\begin{aligned} x_{i+2} &= \boxed{(x_i - m\sigma_i 2^{-i} y_i - m\sigma_{i+1} 2^{-i-1} y_i)} \\ y_{i+2} &= \boxed{(y_i + \sigma_i 2^{-i} x_i + \sigma_{i+1} 2^{-i-1} x_i)} \end{aligned}$$

$$\begin{aligned} x_{i+2} &= \boxed{(x_i - m\sigma_i 2^{-i} y_i - m\sigma_{i+1} 2^{-i-1} y_i)} \cdot \boxed{(1 + \lambda(\sigma_i)\, m\, 2^{-2i-1} x_i + \lambda(\sigma_{i+1})\, m\, 2^{-2i-3} x_i)} \\ y_{i+2} &= \boxed{(y_i + \sigma_i 2^{-i} x_i + \sigma_{i+1} 2^{-i-1} x_i)} \cdot \boxed{(1 + \lambda(\sigma_i)\, m\, 2^{-2i-1} y_i + \lambda(\sigma_{i+1})\, m\, 2^{-2i-3} y_i)} \end{aligned}$$

scaling factor compensation.

$$\begin{bmatrix} x_{i+2} \\ y_{i+2} \end{bmatrix} = \begin{bmatrix} 1 - m\boxed{\sigma_i \sigma_{i+1}} 2^{-2i-1} & -m\left(\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}\right) \\ \left(\sigma_i 2^{-i} + \sigma_{i+1} 2^{-i-1}\right) & 1 - m\boxed{\sigma_i \sigma_{i+1}} 2^{-2i-1} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

$$x_{i+2} = \left(x_i - m\sigma_i 2^{-i} y_i - m\sigma_{i+1} 2^{-i-1} y_i\right) - m\boxed{\sigma_i \sigma_{i+1}} 2^{-2i-1} x_i$$

$$y_{i+2} = \left(y_i + \sigma_i 2^{-i} x_i + \sigma_{i+1} 2^{-i-1} x_i\right) - m\boxed{\sigma_i \sigma_{i+1}} 2^{-2i-1} y_i$$

$$x_{i+2} = \left(x_i - m\sigma_i 2^{-i} y_i - m\sigma_{i+1} 2^{-i-1} y_i\right)\left(1 + \lambda(\sigma_i) m 2^{-2i-1} x_i + \lambda(\sigma_{i+1}) m 2^{-2i-3} x_i\right)$$

$$y_{i+2} = \left(y_i + \sigma_i 2^{-i} x_i + \sigma_{i+1} 2^{-i-1} x_i\right)\left(1 + \lambda(\sigma_i) m 2^{-2i-1} y_i + \lambda(\sigma_{i+1}) m 2^{-2i-3} y_i\right)$$

$$\lambda(\sigma_i) = 1 \quad \text{for } |\sigma_i| = 0 \qquad \{0\}$$

$$\lambda(\sigma_i) = 0 \quad \text{for } |\sigma_i| = 1 \qquad \{1, \bar{1}\}$$

$$\lambda(\sigma_{i+1}) = 1 \quad \text{for } |\sigma_{i+1}| = 0 \qquad \{0\}$$

$$\lambda(\sigma_{i+1}) = 0 \quad \text{for } |\sigma_{i+1}| = 1 \qquad \{1, \bar{1}\}$$

$$x_{i+2} = \boxed{(x_i - m\,\sigma_i\, 2^{-i}\, y_i - m\,\sigma_{i+1}\, 2^{-i-1}\, y_i)}\;\boxed{(1 + \lambda(\sigma_i)\, m\, 2^{-2i-1}\, x_i + \lambda(\sigma_{i+1})\, m\, 2^{-2i-3}\, x_i)}$$

$$y_{i+2} = \boxed{(y_i + \sigma_i\, 2^{-i}\, x_i + \sigma_{i+1}\, 2^{-i-1}\, x_i)}\;\boxed{(1 + \lambda(\sigma_i)\, m\, 2^{-2i-1}\, y_i + \lambda(\sigma_{i+1})\, m\, 2^{-2i-3}\, y_i)}$$

rotation by $\alpha_{m,i}$ or $\alpha_{m,i+1}$

multiplex the diff. shifts

$\Rightarrow$ 4-to-2 cells x_{i+2}, y_{i+2}

3-to-2 cells z_{i+2}

Timmermann's constant scaling factor

for n-bit precision.

parallelizable

late evaluation

after all iterations

Wallace Tree

σ_i's are recoded in parallel

# of non-zero σ_i's at most half of max value w/o recoding

$\boxed{\sigma_i \sigma_{i+1}} = 0$

	$\boxed{\sigma_i \sigma_{i+1}}$
case 1	0 1
case 2	1 0
case 3	0 0

$$(1+m2^{-2i-1})\cdot(1+m2^{-2i-3}) = 1 + m2^{-2i-1} + m2^{-2i-3}$$

$$(1+m2^{-2i-1})\cdot(1+m2^{-2i-3})\cdot(1+m2^{-2i-5}) = 1 + m2^{-2i-1} + m2^{-2i-3} + m2^{-2i-5}$$

$$(1+m2^{-2i-0-1})\cdot(1+m2^{-2i-2-1})\cdot(1+m2^{-2i-4-1})\cdot(1+m2^{-2i-6-1})\cdots$$
$$= 1 + m2^{-2i-0-1} + m2^{-2i-2-1} + m2^{-2i-4-1} + m2^{-2i-6-1}$$

$$\prod_{j=0}^{n}(1+m2^{-2i-2j-1}) = 1 + \sum_{j=0}^{n} m2^{-2i-2j-1}$$

Constrained by n-bit accuracy

$\alpha_{m,i}$ $\alpha_{m,i+1}$

$$(1+m2^{-2i-1})\cdot(1+m2^{-2i-3}) = 1 + m2^{-2i-1} + m2^{-2i-3}$$

	$\sigma_i = 1$, $\sigma_{i+1} = 0$	$\sigma_i = 0$, $\sigma_{i+1} = 1$	$\sigma_i = 0$, $\sigma_{i+1} = 0$
no rotation by	$\alpha_{m,i+1}$	$\alpha_{m,i}$	$\alpha_{m,i}$ & $\alpha_{m,i+1}$
SF compensation	$(1+m2^{-2i-3})$	$(1+m2^{-2i-1})$	$1+m2^{-2i-1}+m2^{-2i-3}$

Modified Booth Encoding

$\lambda(t) = 1 \quad \text{for } |t| = 0$
$\lambda(t) = 0 \quad \text{for } |t| = 1 \quad \{1, \bar{1}\}$

$$x_{i+2} = (x_i - m\,\sigma_i\, 2^{-i}\, y_i - m\,\sigma_{i+1}\, 2^{-i-1}\, y_i\,)$$
$$= (1 + \lambda(\sigma_i)\, m\, 2^{-2i-1}\, x_i + \lambda(\sigma_{i+1})\, m\, 2^{-2i-3}\, x_i$$
$$y_{i+2} = (y_i + \sigma_i\, 2^{-i}\, x_i + \sigma_{i+1}\, 2^{-i-1}\, x_i)$$
$$= (1 + \lambda(\sigma_i)\, m\, 2^{-2i-1}\, y_i + \lambda(\sigma_{i+1})\, m\, 2^{-2i-3}\, y_i)$$

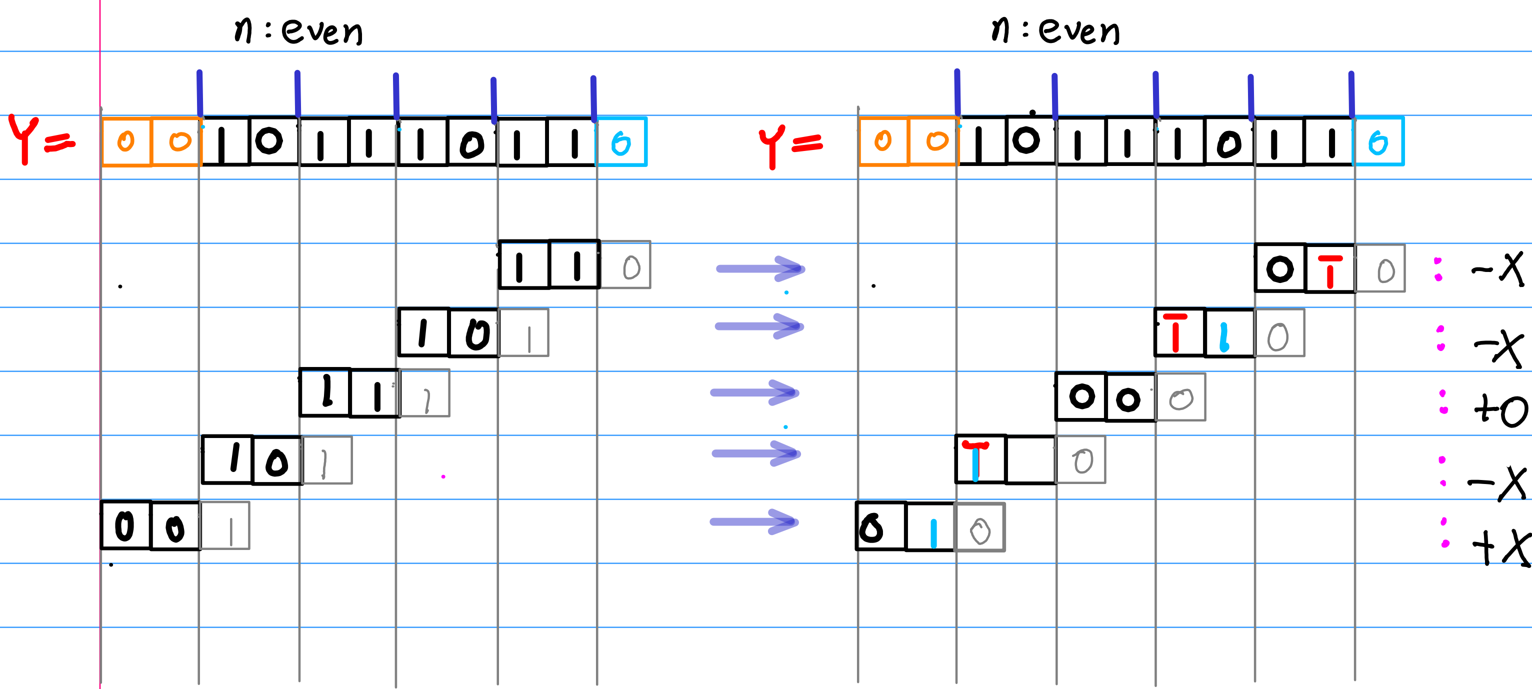
Modified Booth Encoding

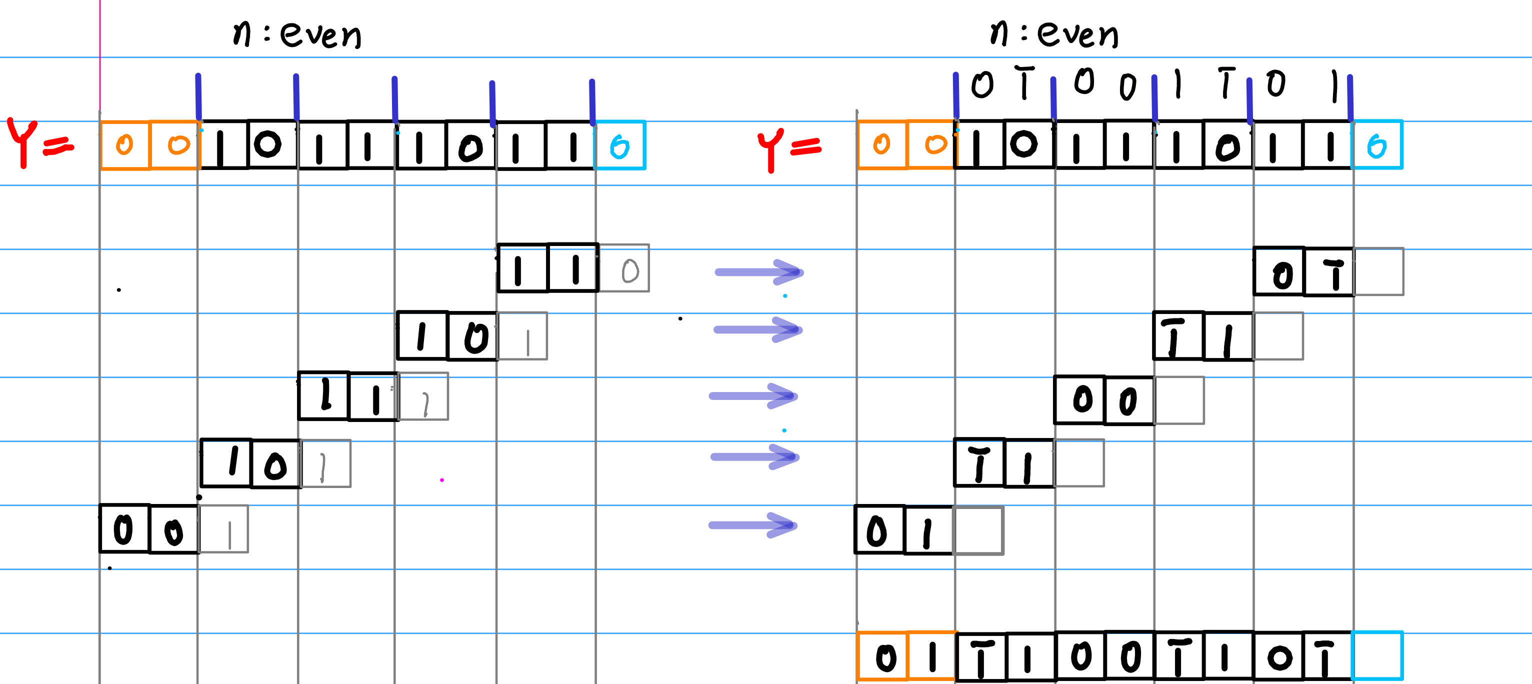
2-bit encoding		2^1 2^0	scale factor	
all zero's	0 0 0	0 0 0	$0 \cdot 2 + 0 = +0$	$+0$
end of 1's	0 0 1	0 1 0	$0 \cdot 2 + 1 = +1$	$+X$
isolated 1	0 1 0	1 $\bar{1}$ 0	$1 \cdot 2 + \bar{1} = +1$	$+X$
end of 1's	0 1 1	1 0 0	$1 \cdot 2 + 0 = +2$	$+2X$
start of 1's	1 0 0	$\bar{1}$ 0 0	$\bar{1} \cdot 2 + 0 = -2$	$-2X$
isolated 0	1 0 1	$\bar{1}$ 1 0	$\bar{1} \cdot 2 + 1 = -1$	$-X$
start of 1's	1 1 0	0 $\bar{1}$ 0	$0 \cdot 2 + \bar{1} = -1$	$-X$
all 1's	1 1 1	0 0 0	$0 \cdot 2 + 0 = 0$	$+0$

Scale factor $\{0, \pm 1, \pm 2\}$

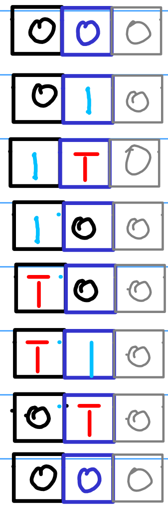
not the one Timmermann's paper refers to

Original Booth Encoding

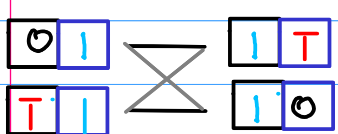


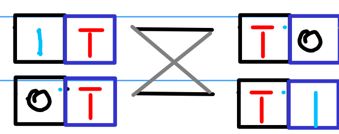


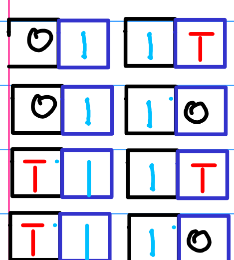
After the 1st Pass

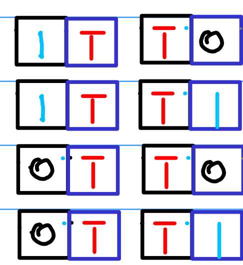


possible boundary cases

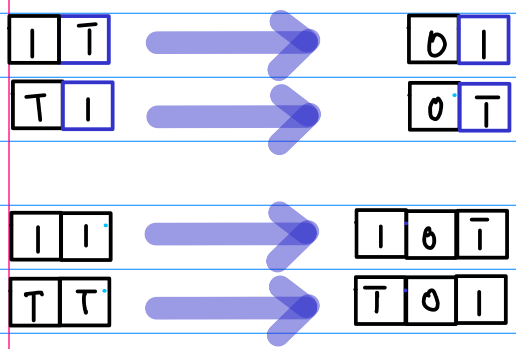




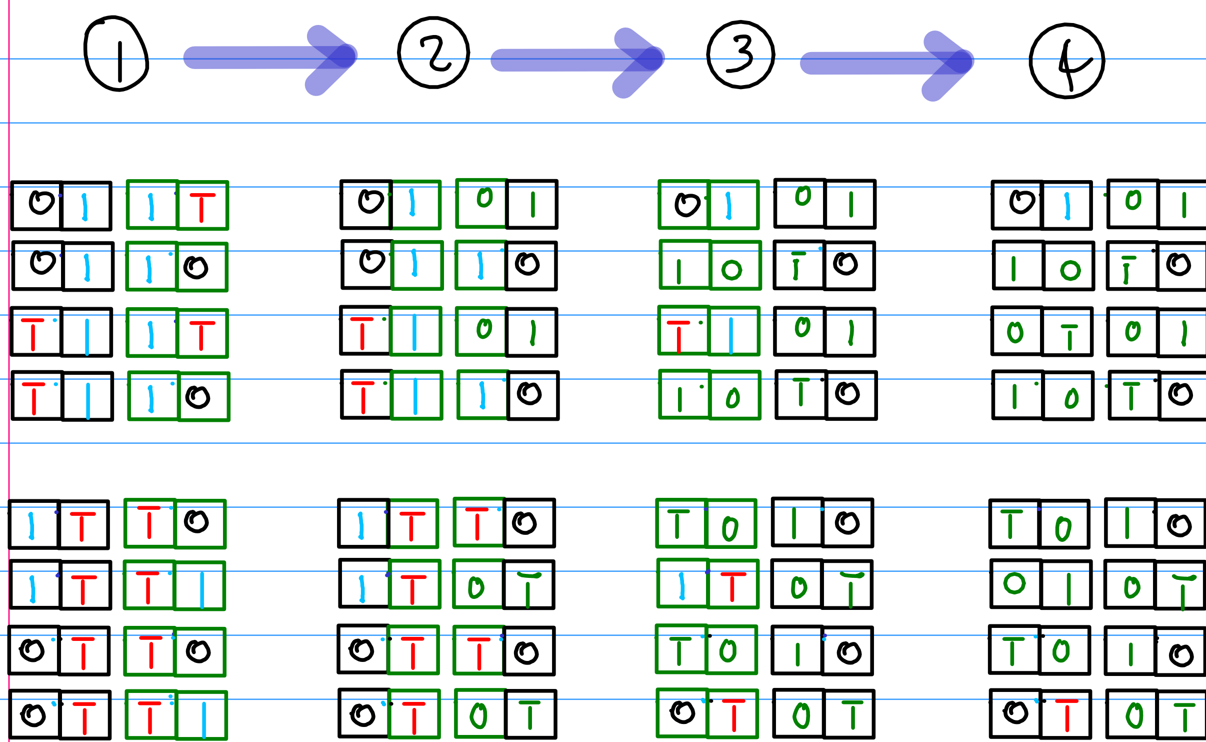




Pass 2 Operation



iterative application

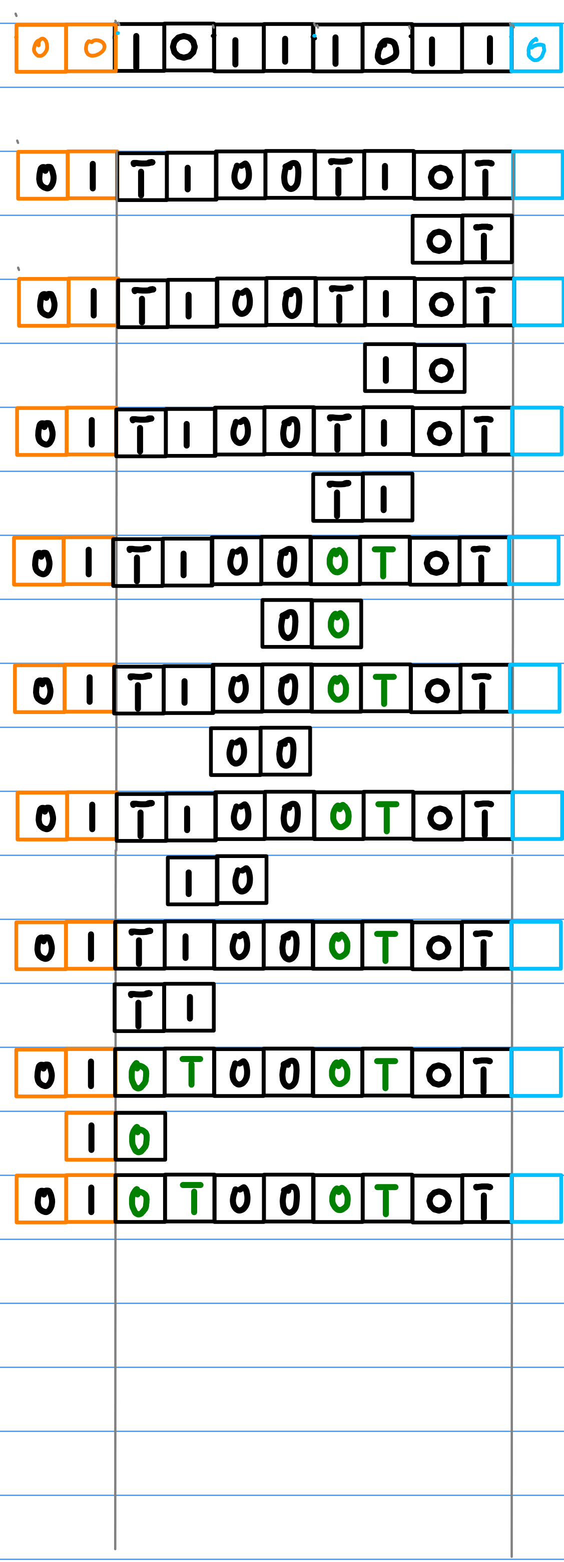


$$\boxed{\sigma_i \sigma_{i+1}} = 0$$

the 2^{nd} pass

$Y =$

$\boxed{\sigma_i \sigma_{i+1}} \cancel{\neq} 0$



$\boxed{\sigma_i \sigma_{i+1}} = 0$

CSD approach

Efficient canonic signed digit recoding
Gustavo A. Ruiz n , Mercedes Granda
Microelectronics Journal 42 (2011)

Iterative Reduction of 1-runs

$\boxed{\sigma_i \sigma_{i+1}} = 0$

Unique encoding

$0\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 0$

$0\ 1\ 1\ 0$

$0\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ \bar{1}\ 0$

$0\ 1\ 1\ 1\ 1\ 0$

$0\ 0\ 1\ 1\ 0\ 0\ 0\ \bar{1}\ 0\ \bar{1}\ 0$

$0\ 1\ 1\ 0$

$0\ 1\ 0\ \bar{1}\ 0\ 0\ 0\ \bar{1}\ 0\ \bar{1}\ 0$

$0\ 1\ 1\ 0$

$1\ 0\ \bar{1}\ 0$

$0\ 1\ 1\ 1\ 0$

$1\ 0\ 0\ \bar{1}\ 0$

$0\ 1\ 1\ 1\ 1\ 0$

$1\ 0\ 0\ 0\ \bar{1}\ 0$

$0\ 1\ 1\ 1\ 1\ 1\ 0$

$1\ 0\ 0\ 0\ 0\ \bar{1}\ 0$

$0\ 1\ 1\ 1\ 1\ 1\ 1\ 0$

$1\ 0\ 0\ 0\ 0\ 0\ \bar{1}\ 0$

Verification

		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
0	0	1	0	1	1	1	0	1	1	0
0	1	$\bar{1}$	1	0	0	$\bar{1}$	1	0	$\bar{1}$	
0	1	0	$\bar{1}$	0	0	0	$\bar{1}$	0	$\bar{1}$	
0	1	0	$\bar{1}$	0	0	0	$\bar{1}$	0	$\bar{1}$	0

$2^7 + 2^5 + 2^4 + 2^3 + 2^1 + 2^0 = 128 + 32 + 16 + 8 + 2 + 1 = 187$

$2^8 - 2^7 + 2^6 - 2^3 + 2^2 - 2^0 = 256 - 128 + 64 - 8 + 4 - 1 = 187$

$2^8 - 2^6 - 2^2 - 2^0 = 256 - 64 - 4 - 1 = 187$

$2^8 - 2^6 - 2^2 - 2^0 = 256 - 64 - 4 - 1 = 187$

0	0	1	0	1	0	1	1	1	0
					0	1	1	1	0
0	0	1	0	1	1	0	0	1	0
			0	1	1	0			
0	0	1	1	0	1	0	0	1	0
	0	1	1	0					
0	1	0	1	0	1	0	0	1	0

Canonical Signed Digit (CSD)

(1) the number of non-zero digits is minimal

(2) no two consecutive digits are both non-zero
two non-zero digits are not adjacent