Redundant CORDIC
Timmermann (C)

20170118

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

Copyright (c) 2015 - 2016 Young W. Lim.
  Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

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

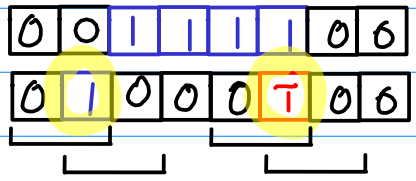
CSD (Canonic Signed Digit)

like Booth encoding (not modified Booth)

all non-zero digits are separated by zeros

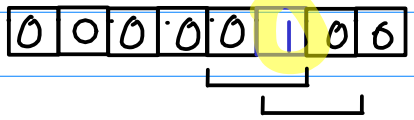
$\Rightarrow$ $\boxed{\sigma_i \sigma_{i+1} = 0}$

1-run



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

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



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

$i+1$

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

$i+2$

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

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

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

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

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

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

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

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

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

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

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

$\sigma_i = 0$ → inc/dec no rotation, but compensate the scale factor.

$$
\begin{aligned}
x_{i+1} &= x_i + m \cdot 2^{-2i-1} x_i \quad \text{inc/dec} \\
y_{i+1} &= y_i + m \cdot 2^{-2i-1} y_i \quad \text{inc/dec} \\
z_{i+1} &= z_i \qquad\qquad m=+1 \,/\, m=-1
\end{aligned}
$$

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

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

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

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

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

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

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

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

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

$\sigma_i \neq 0$

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

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

$\sigma_i = 0$

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

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

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

$\sigma_i \neq 0$

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

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

$\sigma_i = 0$

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

– $0 \le i \le (n-3)/4$:

use the prediction algorithm, generate σ_i from z_i using Table 1 in the same manner as in Fig. 1, $\sigma_i \in \{-1,1\}$, execute iterations according to Eqs. 1–3

– $(n-3)/4 < i \le (n+1)/2$:

use the prediction algorithm, generate σ_i from z_i by special recoding (explained later), $\sigma_i \in \{0,1,-1\}$, after each iteration increment i by 2

$\sigma_i <> 0$:
$$x_{i+2} = x_i - m\,\sigma_i\,2^{-i}\,y_i - m\,\sigma_{i+1}\,2^{-i-1}\,y_i$$
$$y_{i+2} = y_i + \sigma_i 2^{-i}\,x_i + \sigma_{i+1}2^{-i-1}x_i$$
$$z_{i+2} = z_i - \sigma_i\alpha_{m,i} - \sigma_{i+1}\alpha_{m,i+1}$$

$\sigma_i = 0$:
$$x_{i+2} = x_i + m\,2^{-2i-1}x_i + m\,2^{-2i-2}x_i \quad (11)$$
$$y_{i+2} = y_i + m\,2^{-2i-1}y_i + m\,2^{-2i-2}y_i \quad (12)$$
$$z_{i+2} = z_i \quad (13)$$

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

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

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

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

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

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

– $0 \le i \le (n-3)/4$:
use the prediction algorithm, generate σ_i from z_i using Table 1 in the same manner as in Fig. 1, $\sigma_i \in \{-1,1\}$, execute iterations according to Eqs. 1–3

– $(n-3)/4 < i \le (n+1)/2$:
use the prediction algorithm, generate σ_i from z_i by special recoding (explained later), $\sigma_i \in \{0,1,-1\}$, after each iteration increment i by 2

$\sigma_i <> 0$:
$$\begin{aligned} x_{i+2} &= x_i - m\,\sigma_i\,2^{-i}\,y_i - m\,\sigma_{i+1}\,2^{-i-1}\,y_i \\ y_{i+2} &= y_i + \sigma_i 2^{-i}\,x_i + \sigma_{i+1} 2^{-i-1} x_i \\ z_{i+2} &= z_i - \sigma_i \alpha_{m,i} - \sigma_{i+1} \alpha_{m,i+1} \end{aligned}$$

$\sigma_i = 0$:
$$\begin{aligned} x_{i+2} &= x_i + m\,2^{-2i-1} x_i + m\,2^{-2i-2} x_i \quad (11) \\ y_{i+2} &= y_i + m\,2^{-2i-1} y_i + m\,2^{-2i-2} y_i \quad (12) \\ z_{i+2} &= z_i \quad (13) \end{aligned}$$

$$\lambda(-1) = 0$$

$$\lambda(1) = 0$$

$$\lambda(0) = 1$$

$$\lambda(\sigma_i) = 0 \quad \sigma_i \in \{1, -1\}$$

$$= 1 \quad \sigma_i = 0$$

Modified CORDIC

Timmermann 1989 Electronics Letters

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

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

$$z_n = z_0 + \alpha$$

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

1st half iterations : the most significant contribution

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

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

α_i decreases with the increasing iteration index i

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

rotation $z_n \to 0$

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

Vectoring $y_n \to 0$

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

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 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 $k_m = 1$

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

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

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

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

for a sufficiently small z_j

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

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

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

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

Rotation mode

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

multiplex the diff. shifts

⇒ 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

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

	σ_i	σ_{i+1}
case ①	0	1
case ②	1	0
case ③	0	0

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

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

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

$$= 1 + m2^{-2i-0-1} + m2^{-2i-2-1} + m2^{-2i-4-1} + m2^{-2i-6-1}$$

Modified Booth Encoding

$\lambda(t) = 1$ for $|t| = 0$

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

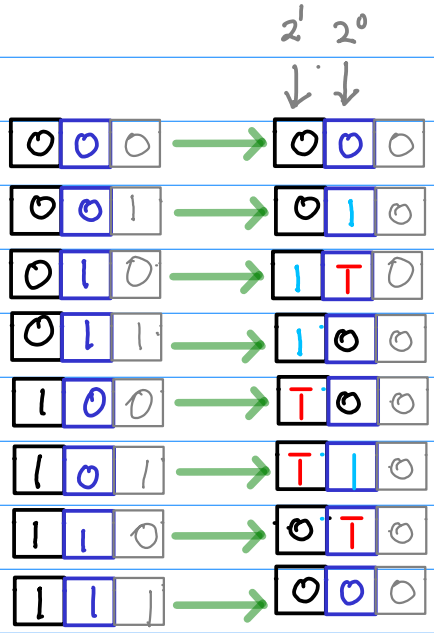
$$x_{i+2} = (x_i - m\,\sigma_i\,2^{-i}\,y_i - m\,\sigma_{i+1}\,2^{-i-1}\,y_i)$$

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

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

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

2-bit encoding



scale factor

$0 \cdot 2 + 0 = +0$ → $+0$

$0 \cdot 2 + 1 = +1$ → $+X$

$1 \cdot 2 + T = +1$ → $+X$

$1 \cdot 2 + 0 = +2$ → $+2X$

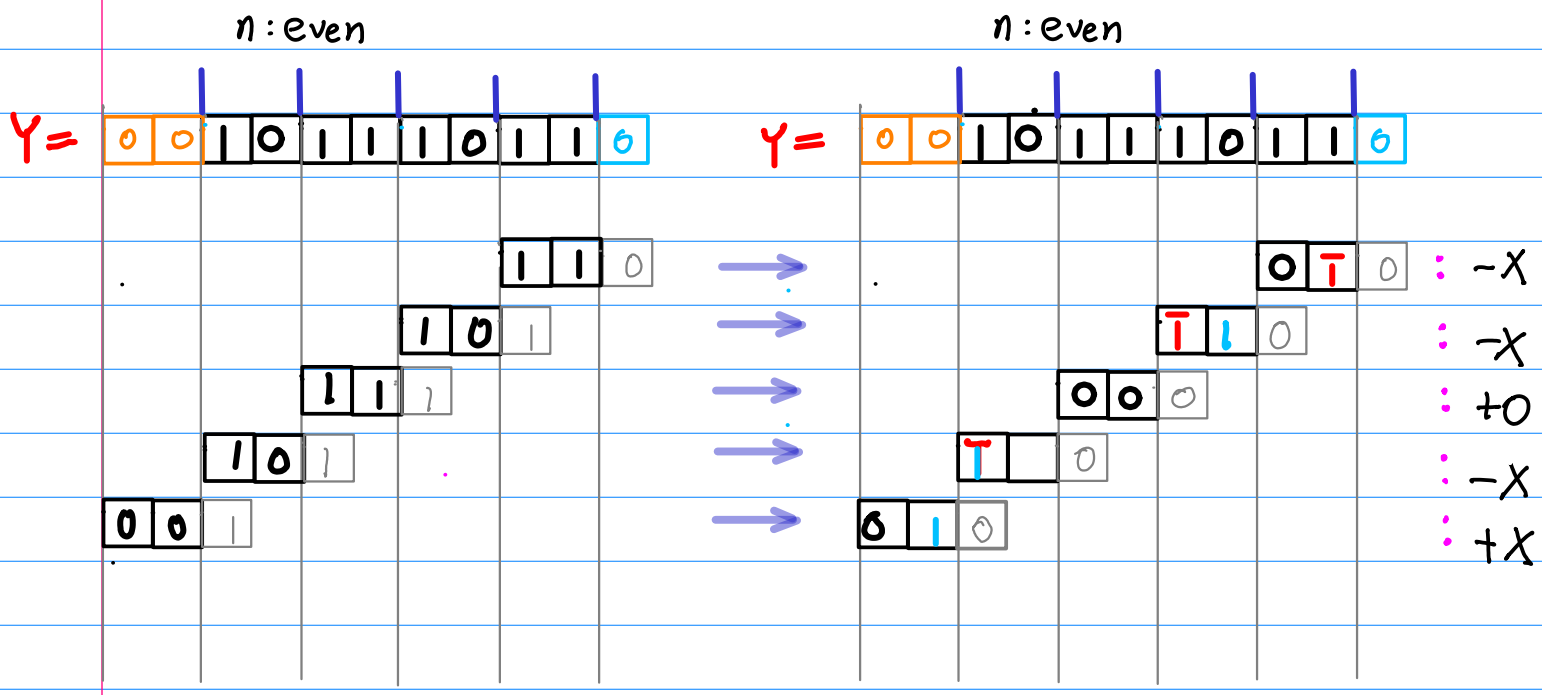
$T \cdot 2 + 0 = -2$ → $-2X$

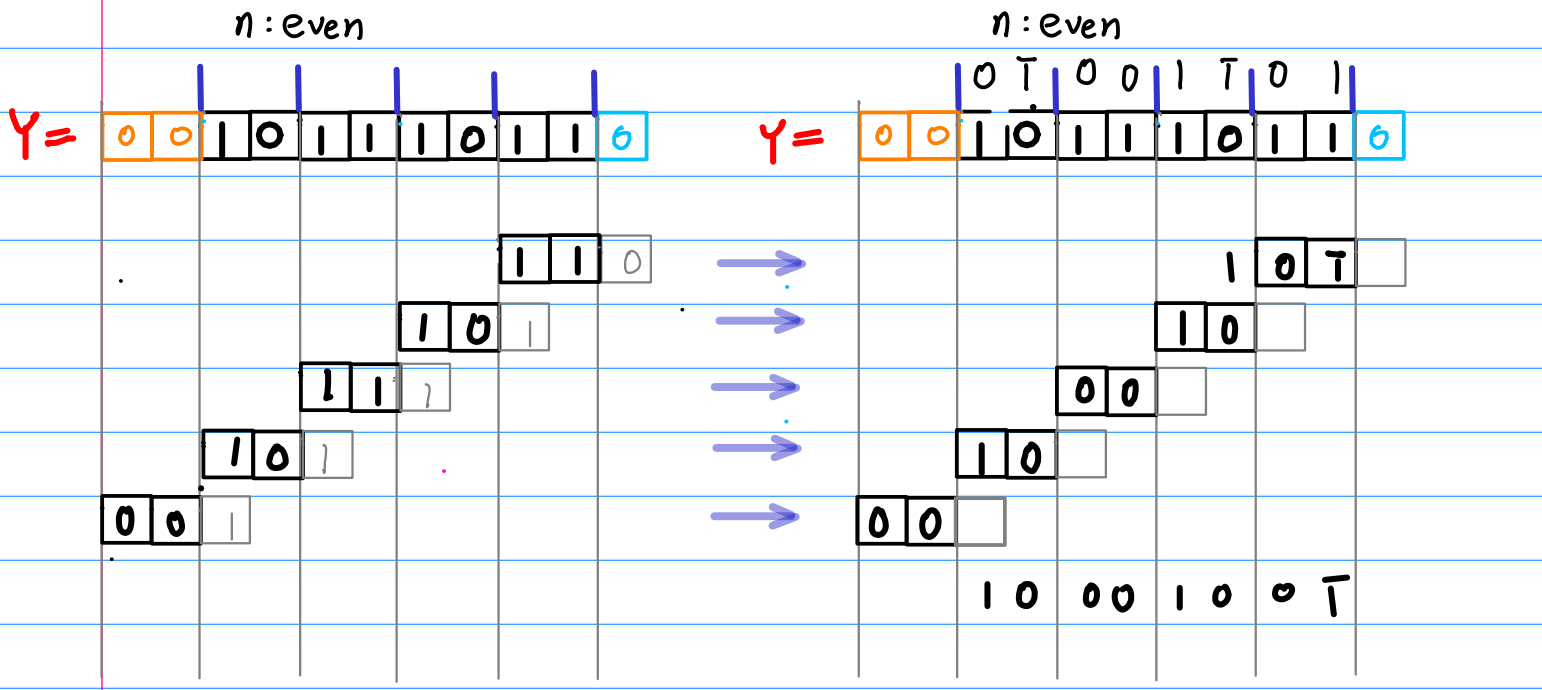
$T \cdot 2 + 1 = -1$ → $-X$

$0 \cdot 2 + T = -1$ → $-X$

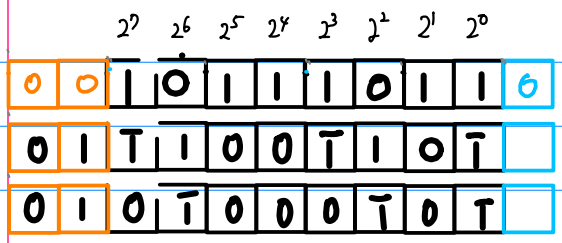
$0 \cdot 2 + 0 = 0$ → $+0$

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





0 0	0 0
0 1	T 0
1 0	1 0
1 1	0 0



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

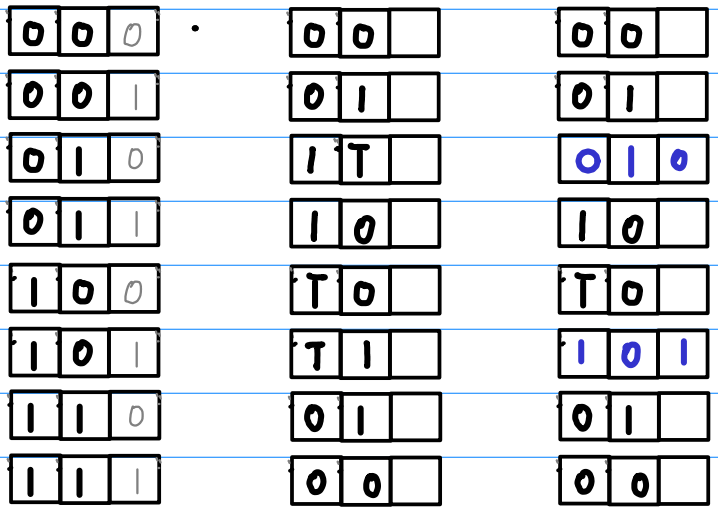
$2^8 - 2^7 + 2^6 - 2^3 + 2^2 - 2^0 = 256 - 128 + 64 - 8 + 4 - 1 = 187$

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

1 1	
1 0	1 0
1 T	0 1
0 1	0 1
0 0	0 0
0 T	0 T
T 1	0 T
T 0	T 0
T T	

0 | 1 | 0

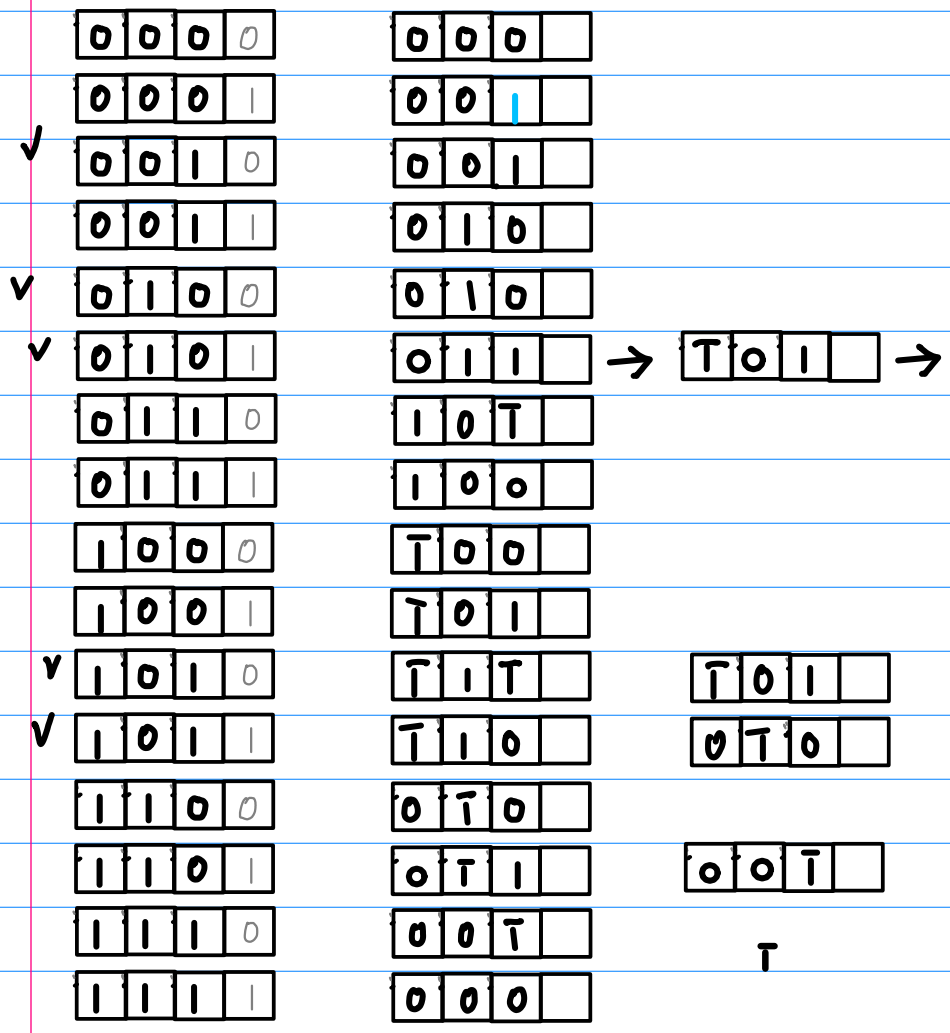
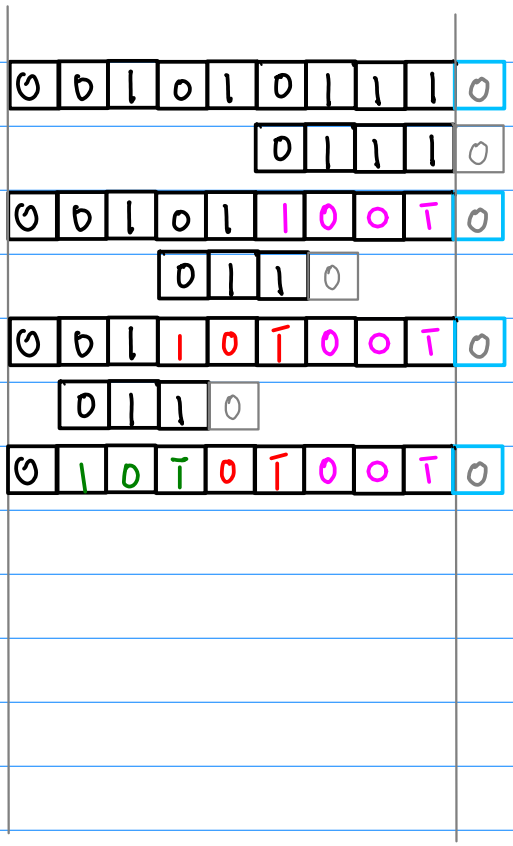
1 0 0 T 0



0 | 0

T

1

1 0 1

T 1