# *Applications of Distributed Arithmetic to Digital Signal Processing:*

## *A Tutorial Review*

Ref: *Stanley A. White, "Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review," IEEE ASSP Magazine, July, 1989*

# Distributed Arithmetic (DA, 1974)

- The most-often encountered form of computation in DSP:
  - Sum of product
  - Inner-product
  - Executed most efficiently by DA

# Derivation of DA Technique

- Sum of product: $$y = \sum_{k=1}^{K} A_k x_k \qquad (1)$$
  - where $x_k$ is a 2's-complement binary number scaled such that $|x_k|<1$, and $A_k$ is fixed coefficients
- $x_k$: $\{b_{k0}, b_{k1}, b_{k2}, \ldots\ldots, b_{k(N-1)}\}$, wordlength=$N$
  - where $b_{k0}$ is the sign bit
  - Express each $x_k$ as: $$x_k = -b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \qquad (2)$$

Sub (2) into (1) =>

$$y = \sum_{k=1}^{K} A_k \left[ -b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n} \right] \qquad (3)$$

# Derivation of DA Technique
## -continued I

- Critical step

$$y=\sum_{k=1}^{K}A_k\left[-b_{k0}+\sum_{n=1}^{N-1}b_{kn}2^{-n}\right]=\sum_{k=1}^{K}A_k\sum_{n=1}^{N-1}b_{kn}2^{-n}+\sum_{k=1}^{K}A_k(-b_{k0})$$

$$=\sum_{n=1}^{N-1}\left[\sum_{k=1}^{K}A_k b_{kn}\right]2^{-n}+\sum_{k=1}^{K}A_k(-b_{k0}) \qquad (4)$$

- where K is the number of inputs (or taps) and N is the wordlength of Data

# Derivation of DA Technique
## -continued II

- Now consider the equation (4)

$$y = \sum_{n=1}^{N-1}\left[\sum_{k=1}^{K} A_k b_{kn}\right]2^{-n} + \sum_{k=1}^{K} A_k(-b_{k0})$$

  – $\left[\sum_{k=1}^{K} A_k b_{kn}\right]$ has only $2^K$ possible values

  – $\sum_{k=1}^{K} A_k(-b_{k0})$ has only $2^K$ possible values

  – We can store it in a lookup-table(ROM): size=$2\times 2^K$

# Technical Overview of DA

- Advantage of DA: Efficiency of computing mechanization
- A frequently argued:
  - Slowness because of its inherent bit-serial nature (not true)
- Some modifications to increase the speed by employing techniques:
  - Plus more arithmetic operations
  - expense of exponentially increased memory

# Derivation of DA Technique
## -continued III

- Example
  - Let number of inputs K=4
  - The fixed coefficients are $A_1$ =0.72, $A_2$= -0.3, $A_3$ = 0.95, $A_4$ = 0.11

$$y = \sum_{n=1}^{N-1}\left[\sum_{k=1}^{K} A_k b_{kn}\right]2^{-n} + \sum_{k=1}^{K} A_k(-b_{k0})$$

  - We need $2\times2^K$ = 32-word ROM (K=4)

# Example

- Unfolding

$$\left[\sum_{k=1}^{4} A_k b_{kn}\right] = A_1 b_{1n} + A_2 b_{2n} + A_3 b_{3n} + A_4 b_{4n}$$

$$\sum_{k=1}^{4} A_k(-b_{k0}) =$$

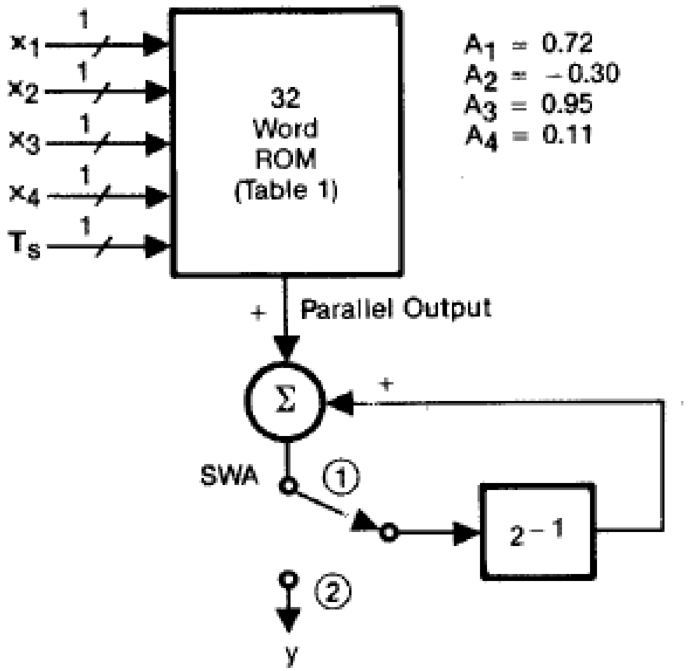
$$A_1(-b_{1,0}) + A_2(-b_{2,0}) + A_3(-b_{3,0}) + A_4(-b_{4,0})$$

**Table 1**

| | Input Code $T_s$ | $b_{1n}$ | $b_{2n}$ | $b_{3n}$ | $b_{4n}$ | 32-Word Memory Contents |
|---|---|---|---|---|---|---|
| $1 \le n \le N-1$ | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 1 | $A_4$ = 0.11 |
| | 0 | 0 | 0 | 1 | 0 | $A_3$ = 0.95 |
| | 0 | 0 | 0 | 1 | 1 | $A_3 + A_4$ = 1.06 |
| | 0 | 0 | 1 | 0 | 0 | $A_2$ = −0.30 |
| | 0 | 0 | 1 | 0 | 1 | $A_2 + A_4$ = −0.19 |
| | 0 | 0 | 1 | 1 | 0 | $A_2 + A_3$ = 0.65 |
| | 0 | 0 | 1 | 1 | 1 | $A_2 + A_3 + A_4$ = 0.75 |
| | 0 | 1 | 0 | 0 | 0 | $A_1$ = 0.72 |
| | 0 | 1 | 0 | 0 | 1 | $A_1 + A_4$ = 0.83 |
| | 0 | 1 | 0 | 1 | 0 | $A_1 + A_3$ = 1.67 |
| | 0 | 1 | 0 | 1 | 1 | $A_1 + A_3 + A_4$ = 1.78 |
| | 0 | 1 | 1 | 0 | 0 | $A_1 + A_2$ = 0.42 |
| | 0 | 1 | 1 | 0 | 1 | $A_1 + A_2 + A_4$ = 0.53 |
| | 0 | 1 | 1 | 1 | 0 | $A_1 + A_2 + A_3$ = 1.37 |
| | 0 | 1 | 1 | 1 | 1 | $A_1 + A_2 + A_3 + A_4$ = 1.48 |
| $n = 0$ | 1 | 0 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 0 | 1 | $-A_4$ = −0.11 |
| | 1 | 0 | 0 | 1 | 0 | $-A_3$ = −0.95 |
| | 1 | 0 | 0 | 1 | 1 | $-(A_3 + A_4)$ = −1.06 |
| | 1 | 0 | 1 | 0 | 0 | $-A_2$ = +0.30 |
| | 1 | 0 | 1 | 0 | 1 | $-(A_2 + A_4)$ = +0.19 |
| | 1 | 0 | 1 | 1 | 0 | $-(A_2 + A_3)$ = −0.65 |
| | 1 | 0 | 1 | 1 | 1 | $-(A_2 + A_3 + A_4)$ = −0.75 |
| | 1 | 1 | 0 | 0 | 0 | $-A_1$ = −0.72 |
| | 1 | 1 | 0 | 0 | 1 | $-(A_1 + A_4)$ = −0.83 |
| | 1 | 1 | 0 | 1 | 0 | $-(A_1 + A_3)$ = −1.67 |
| | 1 | 1 | 0 | 1 | 1 | $-(A_1 + A_3 + A_4)$ = −1.78 |
| | 1 | 1 | 1 | 0 | 0 | $-(A_1 + A_2)$ = −0.42 |
| | 1 | 1 | 1 | 0 | 1 | $-(A_1 + A_2 + A_4)$ = −0.53 |
| | 1 | 1 | 1 | 1 | 0 | $-(A_1 + A_2 + A_3)$ = −1.37 |
| | 1 | 1 | 1 | 1 | 1 | $-(A_1 + A_2 + A_3 + A_4)$ = −1.48 |

# Example -conti

- Hardware architecture




$x_1$ $x_2$ $x_3$ $x_4$

| | Input Code | | | | | 32-Word Memory Contents |
|---|---|---|---|---|---|---|
| | $T_s$ | $b_{1n}$ | $b_{2n}$ | $b_{3n}$ | $b_{4n}$ | |
| $1 \le n \le N-1$ | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 1 | $A_4 = 0.11$ |
| | 0 | 0 | 0 | 1 | 0 | $A_3 = 0.95$ |
| | 0 | 0 | 0 | 1 | 1 | $A_3 + A_4 = 1.06$ |
| | 0 | 0 | 1 | 0 | 0 | $A_2 = -0.30$ |
| | 0 | 0 | 1 | 0 | 1 | $A_2 + A_4 = -0.19$ |
| | 0 | 0 | 1 | 1 | 0 | $A_2 + A_3 = 0.65$ |
| | 0 | 0 | 1 | 1 | 1 | $A_2 + A_3 + A_4 = 0.75$ |
| | 0 | 1 | 0 | 0 | 0 | $A_1 = 0.72$ |
| | 0 | 1 | 0 | 0 | 1 | $A_1 + A_4 = 0.83$ |
| | 0 | 1 | 0 | 1 | 0 | $A_1 + A_3 = 1.67$ |
| | 0 | 1 | 0 | 1 | 1 | $A_1 + A_3 + A_4 = 1.78$ |
| | 0 | 1 | 1 | 0 | 0 | $A_1 + A_2 = 0.42$ |
| | 0 | 1 | 1 | 0 | 1 | $A_1 + A_2 + A_4 = 0.53$ |
| | 0 | 1 | 1 | 1 | 0 | $A_1 + A_2 + A_3 = 1.37$ |
| | 0 | 1 | 1 | 1 | 1 | $A_1 + A_2 + A_3 + A_4 = 1.48$ |
| $n = 0$ | 1 | 0 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 0 | 1 | $-A_4 = -0.11$ |
| | 1 | 0 | 0 | 1 | 0 | $-A_3 = -0.95$ |
| | 1 | 0 | 0 | 1 | 1 | $-(A_3 + A_4) = -1.06$ |
| | 1 | 0 | 1 | 0 | 0 | $-A_2 = +0.30$ |
| | 1 | 0 | 1 | 0 | 1 | $-(A_2 + A_4) = +0.19$ |
| | 1 | 0 | 1 | 1 | 0 | $-(A_2 + A_3) = -0.65$ |
| | 1 | 0 | 1 | 1 | 1 | $-(A_2 + A_3 + A_4) = -0.75$ |
| | 1 | 1 | 0 | 0 | 0 | $-A_1 = -0.72$ |
| | 1 | 1 | 0 | 0 | 1 | $-(A_1 + A_4) = -0.83$ |
| | 1 | 1 | 0 | 1 | 0 | $-(A_1 + A_3) = -1.67$ |
| | 1 | 1 | 0 | 1 | 1 | $-(A_1 + A_3 + A_4) = -1.78$ |
| | 1 | 1 | 1 | 0 | 0 | $-(A_1 + A_2) = -0.42$ |
| | 1 | 1 | 1 | 0 | 1 | $-(A_1 + A_2 + A_4) = -0.53$ |
| | 1 | 1 | 1 | 1 | 0 | $-(A_1 + A_2 + A_3) = -1.37$ |
| | 1 | 1 | 1 | 1 | 1 | $-(A_1 + A_2 + A_3 + A_4) = -1.48$ |

$$y = \sum_{n=1}^{N-1}\left[\sum_{k=1}^{K} A_k b_{kn}\right]2^{-n} + \sum_{k=1}^{K} A_k(-b_{k0})$$

# Example -continued II

- Shorten the table

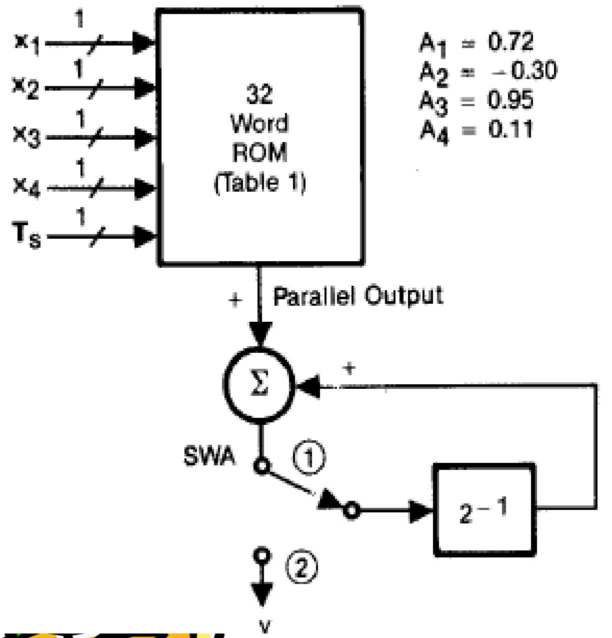
$$\left[\sum_{k=1}^{4} A_k b_{kn}\right] = A_1 b_{1n} + A_2 b_{2n} + A_3 b_{3n} + A_4 b_{4n}$$

$$\sum_{k=1}^{4} A_k(-b_{k0}) = -\left[\sum_{k=1}^{4} A_k(b_{k0})\right]$$

  – Eq. (4)

$$\Rightarrow \quad y = \sum_{n=1}^{N-1}\left[\sum_{k=1}^{K} A_k b_{kn}\right]2^{-n} - \sum_{k=1}^{K} A_k(b_{k0}) \qquad (5)$$

# Example -continued II



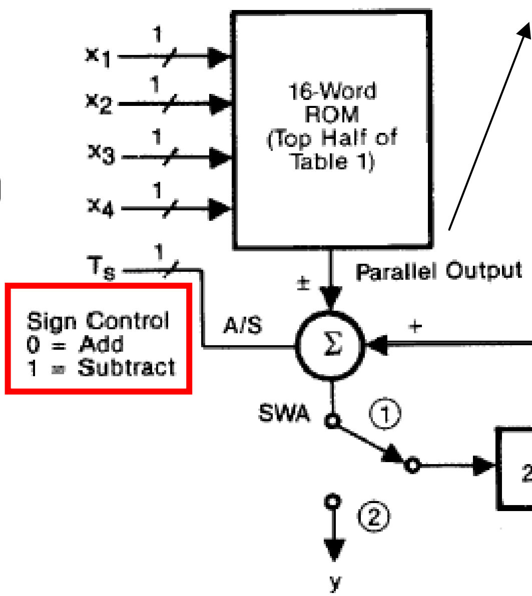




Table 1

| | Input Code | | | | | 32-Word Memory Contents |
|---|---|---|---|---|---|---|
| | $T_s$ | $b_{1n}$ | $b_{2n}$ | $b_{3n}$ | $b_{4n}$ | |
| $1 \le n \le N-1$ | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 1 | $A_4 = 0.11$ |
| | 0 | 0 | 0 | 1 | 0 | $A_3 = 0.95$ |
| | 0 | 0 | 0 | 1 | 1 | $A_3 + A_4 = 1.06$ |
| | 0 | 0 | 1 | 0 | 0 | $A_2 = -0.30$ |
| | 0 | 0 | 1 | 0 | 1 | $A_2 + A_4 = -0.19$ |
| | 0 | 0 | 1 | 1 | 0 | $A_2 + A_3 = 0.65$ |
| | 0 | 0 | 1 | 1 | 1 | $A_2 + A_3 + A_4 = 0.75$ |
| | 0 | 1 | 0 | 0 | 0 | $A_1 = 0.72$ |
| | 0 | 1 | 0 | 0 | 1 | $A_1 + A_4 = 0.83$ |
| | 0 | 1 | 0 | 1 | 0 | $A_1 + A_3 = 1.67$ |
| | 0 | 1 | 0 | 1 | 1 | $A_1 + A_3 + A_4 = 1.78$ |
| | 0 | 1 | 1 | 0 | 0 | $A_1 + A_2 = 0.42$ |
| | 0 | 1 | 1 | 0 | 1 | $A_1 + A_2 + A_4 = 0.53$ |
| | 0 | 1 | 1 | 1 | 0 | $A_1 + A_2 + A_3 = 1.37$ |
| | 0 | 1 | 1 | 1 | 1 | $A_1 + A_2 + A_3 + A_4 = 1.48$ |
| $n = 0$ | 1 | 0 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 0 | 1 | $-A_4 = -0.11$ |
| | 1 | 0 | 0 | 1 | 0 | $-A_3 = -0.95$ |
| | 1 | 0 | 0 | 1 | 1 | $-(A_3 + A_4) = -1.06$ |
| | 1 | 0 | 1 | 0 | 0 | $-A_2 = +0.30$ |
| | 1 | 0 | 1 | 0 | 1 | $-(A_2 + A_4) = +0.19$ |
| | 1 | 0 | 1 | 1 | 0 | $-(A_2 + A_3) = -0.65$ |
| | 1 | 0 | 1 | 1 | 1 | $-(A_2 + A_3 + A_4) = -0.75$ |
| | 1 | 1 | 0 | 0 | 0 | $-A_1 = -0.72$ |
| | 1 | 1 | 0 | 0 | 1 | $-(A_1 + A_4) = -0.83$ |
| | 1 | 1 | 0 | 1 | 0 | $-(A_1 + A_3) = -1.67$ |
| | 1 | 1 | 0 | 1 | 1 | $-(A_1 + A_3 + A_4) = -1.78$ |
| | 1 | 1 | 1 | 0 | 0 | $-(A_1 + A_2) = -0.42$ |
| | 1 | 1 | 1 | 0 | 1 | $-(A_1 + A_2 + A_4) = -0.53$ |
| | 1 | 1 | 1 | 1 | 0 | $-(A_1 + A_2 + A_3) = -1.37$ |
| | 1 | 1 | 1 | 1 | 1 | $-(A_1 + A_2 + A_3 + A_4) = -1.48$ |

Only 16 words of ROM are required now.

# Offset-Binary Coding (OBC)

- Change Input data from "binary" to "signed-digit"

$$x_k = \frac{1}{2}[x_k - (-x_k)] = \{b_{k0}, b_{k1}, \ldots\ldots b_{k(n-1)}\} \qquad (6)$$

$$x_k = -b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n}$$

2's-complement

$$-x_k = -\bar{b}_{k0} + \sum_{n=1}^{N-1} \bar{b}_{kn} 2^{-n} + 2^{-(N-1)}$$

$$x_k = \frac{1}{2}\left[-\left(b_{k0} - \bar{b}_{k0}\right) + \sum_{n=1}^{N-1}\left(b_{kn} - \bar{b}_{kn}\right)2^{-n} - 2^{-(N-1)}\right] \qquad (7)$$

# Offset-Binary Coding (OBC) (cont'd)

$$\Rightarrow \quad x_k = \frac{1}{2}\left[-\left(b_{k0} - \bar{b}_{k0}\right) + \sum_{n=1}^{N-1}\left(b_{kn} - \bar{b}_{kn}\right)2^{-n} - 2^{-(N-1)}\right]$$

- $$c_{kn} = \begin{cases} b_{kn} - \bar{b}_{kn} & ,n \neq 0 \\ -(b_{kn} - \bar{b}_{kn}), & n = 0 \end{cases} \qquad where \quad c_{kn} \in \{-1, 1\}$$

$$\Rightarrow \quad x_k = \frac{1}{2}\left[\sum_{n=0}^{N-1} c_{kn} 2^{-n} - 2^{-(N-1)}\right] \quad \text{代入} \quad y = \sum_{k=1}^{K} A_k x_k$$

$$\Rightarrow \quad y = \frac{1}{2}\sum_{k=1}^{K} A_k \left[\sum_{n=0}^{N-1} c_{kn} 2^{-n} - 2^{-(N-1)}\right] = \sum_{n=0}^{N-1} Q(b_n)2^{-n} + 2^{-(N-1)} Q(0)$$

Where $Q(b_n) = \sum_{k=1}^{K} \frac{A_k}{2} c_{kn}$ and $\boxed{Q(0) = -\sum_{k=1}^{K} \frac{A_k}{2}}$

Constant

# Offset-Binary Coding (OBC) (cont'd)

- Hardware architecture

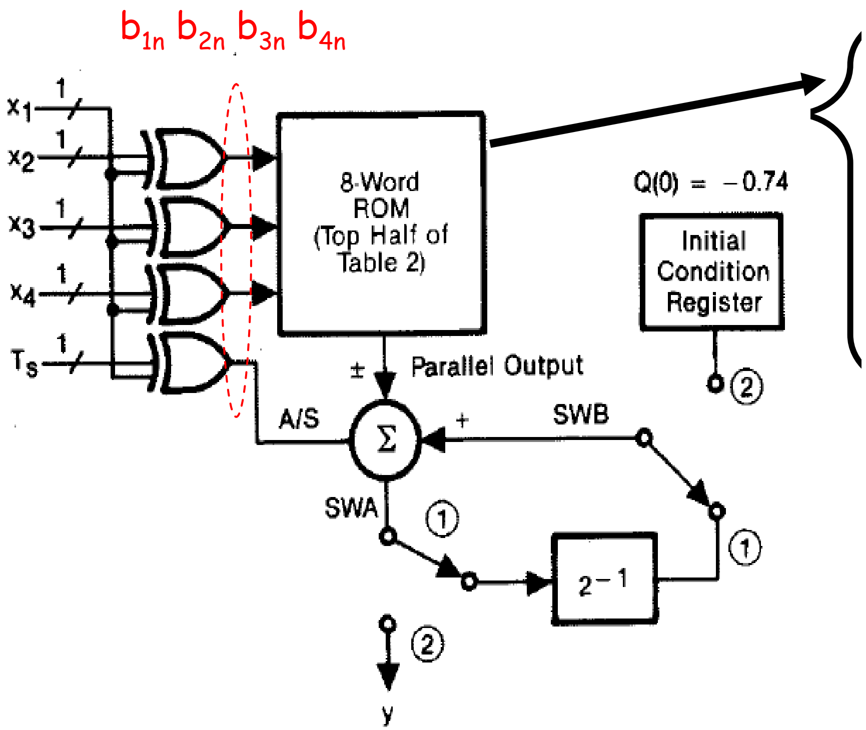



Table 2

| Input Code $b_{1n}$ | $b_{2n}$ | $b_{3n}$ | $b_{4n}$ | 8-Word Memory Contents, Q |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $-1/2\,(A_1+A_2+A_3+A_4) = -0.74$ |
| 0 | 0 | 0 | 1 | $-1/2\,(A_1+A_2+A_3-A_4) = -0.63$ |
| 0 | 0 | 1 | 0 | $-1/2\,(A_1+A_2-A_3+A_4) = 0.21$ |
| 0 | 0 | 1 | 1 | $-1/2\,(A_1+A_2-A_3-A_4) = 0.32$ |
| 0 | 1 | 0 | 0 | $-1/2\,(A_1-A_2+A_3+A_4) = -1.04$ |
| 0 | 1 | 0 | 1 | $-1/2\,(A_1-A_2+A_3-A_4) = -0.93$ |
| 0 | 1 | 1 | 0 | $-1/2\,(A_1-A_2-A_3+A_4) = -0.09$ |
| 0 | 1 | 1 | 1 | $-1/2\,(A_1-A_2-A_3-A_4) = 0.02$ |
| 1 | 0 | 0 | 0 | $1/2\,(A_1-A_2-A_3-A_4) = -0.02$ |
| 1 | 0 | 0 | 1 | $1/2\,(A_1-A_2-A_3+A_4) = 0.09$ |
| 1 | 0 | 1 | 0 | $1/2\,(A_1-A_2+A_3-A_4) = 0.93$ |
| 1 | 0 | 1 | 1 | $1/2\,(A_1-A_2+A_3+A_4) = 1.04$ |
| 1 | 1 | 0 | 0 | $1/2\,(A_1+A_2-A_3-A_4) = -0.32$ |
| 1 | 1 | 0 | 1 | $1/2\,(A_1+A_2-A_3+A_4) = -0.21$ |
| 1 | 1 | 1 | 0 | $1/2\,(A_1+A_2+A_3-A_4) = 0.63$ |
| 1 | 1 | 1 | 1 | $1/2\,(A_1+A_2+A_3+A_4) = 0.74$ |

$\sum_{n=0}^{N-1} Q(b_n)2^{-n} - 2^{-(N-1)}Q(0)$

# Speed up of DA multiplication

- Way I: Plus more arithmetic operations

$$y = \sum_{n=0}^{N-1} Q(b_n)2^{-n} + 2^{-(N-1)}Q(0)$$

Initial condition

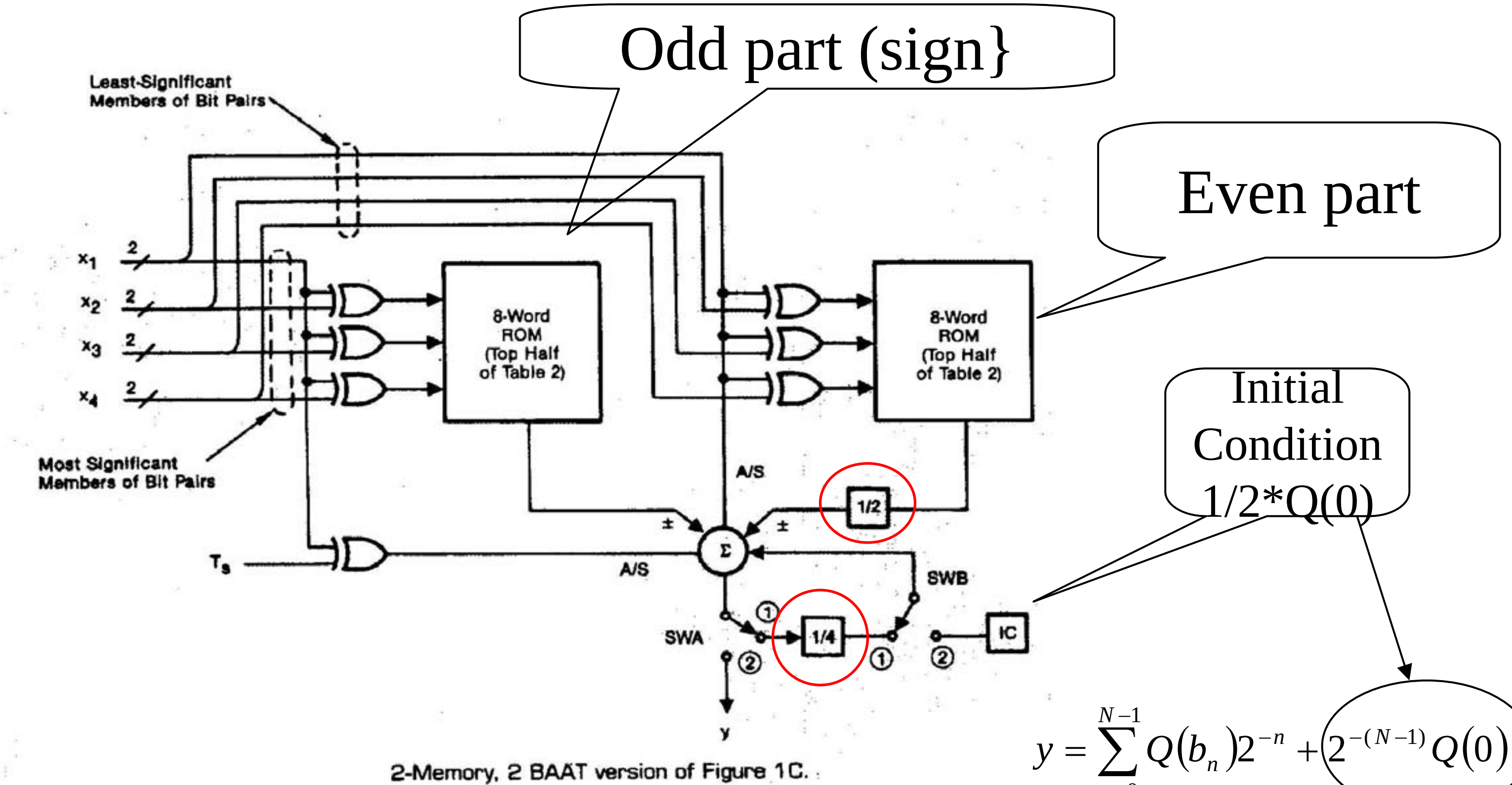
$$\sum_{n=0}^{N-1} Q(b_n)2^{-n} = Q(b_0)2^{-0} + Q(b_1)2^{-1} + \ldots + Q(b_{N-2})2^{-(N-2)} + Q(b_{N-1})2^{-(N-1)}$$

Even part

Odd part

# Speedup of DA multiplication

- Way I: at the expense of linearly increased memory & arithmetic operation




2-Memory, 2 BAAT version of Figure 1C.

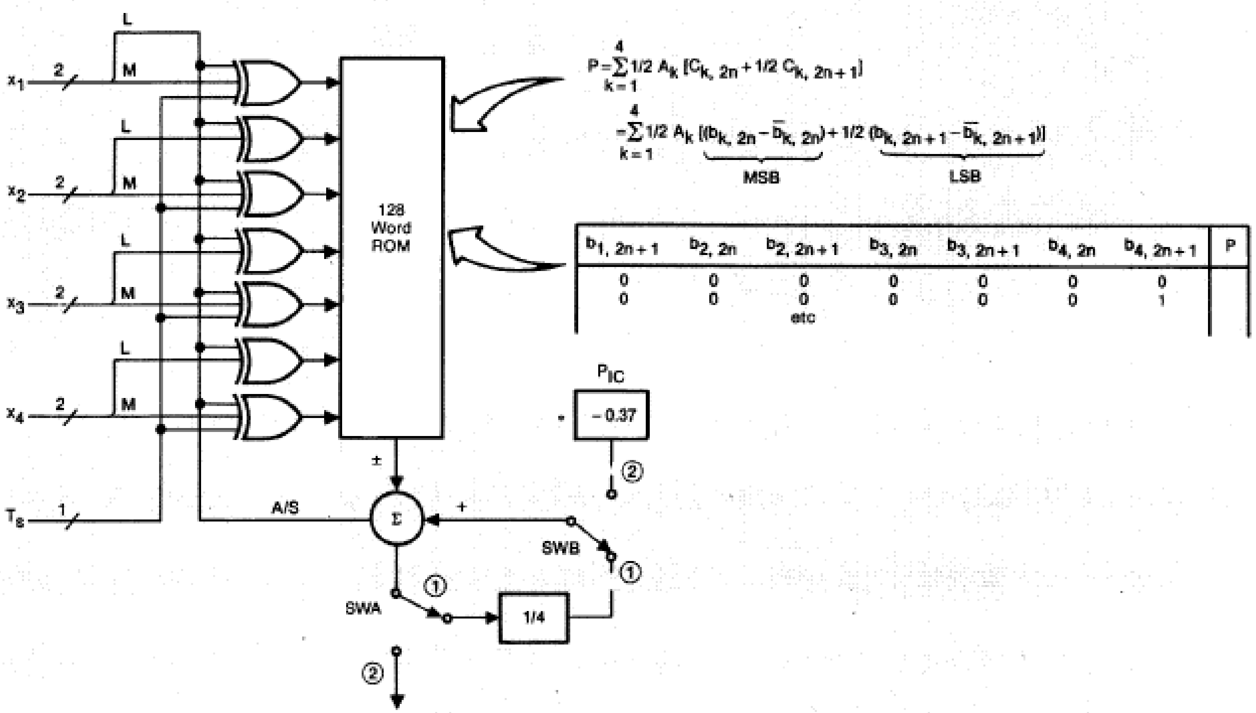
$$y = \sum_{n=0}^{N-1} Q(b_n)2^{-n} + 2^{-(N-1)}Q(0)$$



(cwliu@twins.ee.nctu.edu.tw)

# Speed up of DA Multiplication

- Way II: at the expense of exponentially increased memory
- ROM : 2*7 words ⇨ 1*128 words



# Conclusions

- DA is a very efficient mechanism for computations that are dominated by inner products (convolution)
- A good way to trade combinational logic with memory for high-performance computation.
- When a many computing methods are compared, DA should be considered. It is not always (but often) best, and never poorly: save gate count around 50% to 80%.
- Application: "**VLSI implementation of a 16*16 discrete cosine transform,**" **by** *M.-T. Sun, T.-C. Chen, A. M. Gottlieb, IEEE Transactions on Circuits and Systems,* Volume: 36 Issue: 4 , April 1989, Page(s): 610 –617, and many other transforms and DSP kernels.