

Programmable Variable-rate up/down counter를 사용한信號處理가 가능한 Binary logarithms發生을 위한 回路設計

A circuit design for generating binary logarithms for possible signal processing using programmable variable-rate up/down counter.

* 이 지 영 (Lee, J. Y.)

要 約

本 論文은信號處理가 가능한 2進 보가리즘 函數를發生시키기 위한 programmable variable-rate up/down 계수
기의 設計를 記述한다. 2進數에 對한 밑수가 2 인 로그리즘計算의 適用은 結果的으로 誤差가 發生한다. $\log_2(1+x)$
 $-x$ 에 依해 定義된 것처럼 $\log_2(1+x)$ 에서의 誤差는 直線의 集合으로 간주된다. 계수기 rate는 直線의 기울기에 比
例한다. 그러므로信號處理가 가능한 2進로그리즘 函數는 programmable 계수기를 使用함으로써 쉽게 發生될 수 있다.

ABSTRACT

This paper describes the design of a programmable variable-rate up/down counter for generating
binary logarithmic functions for possible signal processing. An application to the computation of the
base-two logarithm of a binary number can be erroneous in the results.

The error in $\log_2(1+x)$, as defined by $\log_2(1+x) - x$, is considered as a set of straight lines.

* 해군사관학교 조교수

Counter rate is proportional to the slopes of straight lines. So binary logarithmic functions for possible signal processing are generated by using a programmable counter.

I. 序 論

컴퓨터를 이용한 演算方法에는 곱셈, 나눗셈, root, power, exponential 등과 같은 여러가지의 算術演算들이 있는데 結局은 덧셈과 뺄셈에 依해서 實行이 된다. 요즘은 여러 論文들이 2進로가리즘을 사용한 演算方法들을 다루고 있다.

Mitchell은 $N \geq 1$ 일때 2進數 N 의 2진로가리즘 (base가 2인 $\log_2 N$)의 近似值計算을 shifting과 counting을 함으로써 解決하려고 하였다⁽¹⁾. 하지만 이 方法은 相對的으로 느리고 適當한 正確性을 가지고 있다.

Frangakis는 1보다 작은 2進數의 概略的인 2進로가리즘과 anti-logarithm을 計算하는 方法을 提案하였다⁽²⁾. 이것은 piecewise linear approximation이다.

Hall과 Lynch와 Dwyer는 minimum mean square error criteria를 使用하여 近似值係數를 發展시켰다⁽³⁾.

Marino의 近似計算의 결점은 $a \log_2 N$ 의 計算이 지수 K 와 근사가수 x 를 發見하기 위해 counting과 shifting뿐만 아니라 誤差修正을 爲해 너무 많은 演算을 해야 하므로 回路를 複雜하게 하였다⁽⁴⁾. 위의

方法들은 順次쉬프트레지스터, 병렬가산기등 複雑한 하드웨어를 使用한 設計여서 回路가 複雜하고 全體의 計算時間을 增加시킨다. 比較的인 結果가 표 1에 간추려져 있다.

이러한 결점들 때문에 이 論文에서는 前에는 시도되지 않았던 프로그래머블 계수기를 使用하여 2進로가리즘을 發生시키기 爲한 回路設計를 研究한다. 하드웨어가 簡單해지고 誤差가 改善됨을 알 수 있다.

II. Binary-to binary logarithm conversion process

2進 演算을 하는 컴퓨터에 로가리즘을 使用한다면 그것은 밑수가 2인 2進로가리즘일 것이다. 그리고 近似值 로가리즘의 基礎를 다음과 같이 定義한다.

$$\log_2 N; \text{ la } N = \log_2 N$$

N 을 0이 아닌 有限길이의 2進數라면 2進數 N 은 $N = Z_r Z_{r-1} Z_{r-2} \dots Z_{j+1} Z_j$ 로 쓸 수 있다.

이때 $Z_r = 0.1$ 이며 $j = 0, \pm 1, \pm 2, \dots$ 이다. 故로

$$N = \sum_{j=0}^r Z_j 2^j \text{ 이다. (但 } r \geq j)$$

$$\text{또한 } N = 2^k + \sum_{j=0}^{k-1} Z_j 2^j$$

$$= 2^k (1 + \sum_{j=0}^{k-1} Z_j 2^{j-k})$$

$$= 2^k (1 + x)$$

로 쓸 수 있다. (但, $x = \sum_{j=0}^{k-1} Z_j 2^{j-k}$ ($0 \leq x < 1$))

표 1 誤差比較.

Basic equations approached	Number of Sums	Number of Subranges of x	Maximum error in $\log_2(1+x)$
linear approximation	0	0	0.086 (1)
	2	4	0.013 (6)
second order approximation	4	4	0.0099 (3)
	4	2	0.0040 (4)

이때 K 는 N 의 MSB의 2進 거듭제곱(binary power)이며 x 는 K 번째 비트의 右側에 있는 모든 數로서 2進소수(binary fraction)이 된다.

예를 들어 $111.0101 = 2^2 (1 + 0.110101)$ 이며, $0.100101 = 2^{-1} (1 + 0.00101)$ 이 된다.

$N = 2^K (1 + x)$ 에서 $\log_2 N = K + \log_2 (1 + x)$ 이고 2進로가리즘은 $la N = K + la(1 + x)$ 가 된다.

linear approximation은 $la N = K + x$ 가 된다. 여기서 發生한 誤差 E 는 다음과 같다.

$$E = la(1 + x) - x \quad (0 \leq x < 1)$$

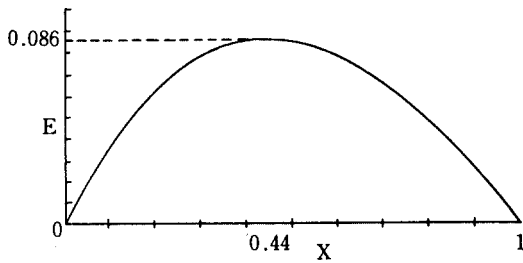


그림 1 X의 函數로서의 誤差曲線.

이때의 最大誤차는 E 를 x 에 關해 微分하여 $\frac{\partial E}{\partial x} = 0$ 으로 함으로써 얻을 수 있다. 卽, 最大絶對誤差 E_{max} 는 $x \approx 0.442695$ 일 때 $E_{max} \approx 0.08639$ 이며, 다음 曲線에서 피이크值가 이에 該當된다.

Mitchell이 提案한 理論의 最大可能 公差誤차는 -11.1%, 最大可能 나눗셈 誤차는 12.5%까지 커질 수 있었다. Combet et al은 더 많은 하드웨어와 진 計算時間을 消費해서 近似誤차의 減少를 通해 Mitchell의 方法을 改善하였다⁽⁶⁾. 이 方法은 piecewise linear approximation을 包含한다.

그림 2에서 네개의 點적한 것에 依해서 0.25, 0.5, 0.75, 1로 나누어지며 擴大된 $\log_2(1+x)$ 曲線과 近似值를 보인다. 여기서 各間隔의 一般의 形態는

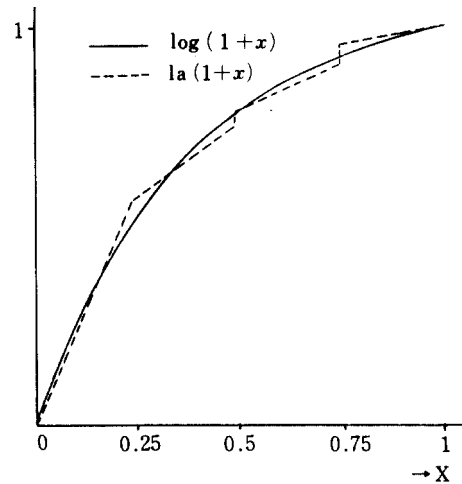


그림 2 $\log_2(1+x)$ 曲線과 近似值線.

$la(1+x) = x + Af(x) + B$ 이다. A 와 B 값은 各各 試行誤차와 또 $\log_2(1+x)$ 로 부터 $la(1+x)$ 의 deviation을 最小化함으로써 얻을 수 있다.

III. Programmable variable-rate up/down counters for generating approximate binary logarithms

이 節에서는 誤차가 最小化되지 않는 勿論, 매우 簡單한 回路를 보여준다.

이 方法은 programmable variable-rate up/down counting을 함으로써 2進 로가리즘을 發生시킨다. $\log_2(1+x) - x$ 인 誤차函數는 正이는 負이는 기울기가 直線의 集合인 것으로 생각할 수 있는데 그 기울기는 2進소수의 積분몹이 되게 선택한다. 여기서는 line segment의 기울기에 比例하는 rate를 가진 프로그래머블 계수기가 사용되었다. 기울기가 正이면 카운트 up하고 기울기가 負이면 카운트 down한다. 이 過程의 結果로서 近似值에는 正 및 負의 誤차가 發生하게 된다.

예를 들어 필승수가 n 비트이고 승수가 m 비트일 때 그積은 $(m+n)$ 비트이다(各各符號를 포함함). 符號비트는 分離해서 取及되므로 ROM을 사용한 직접접합셈에 要求되는 전체 비트수 N 은

$$N = (2^{n+m}) (n+m) \text{이다.}$$

대부분의 경우 積을 P 비트로 반올림하고 ROM 設計에 使用될 때 要求되는 비트數는 $N_1 = (2^{n+m})$

(P)이다. 반올림시의 誤差의 限界는 $-\frac{\rho}{2} < e < \frac{\rho}{2}$ 이

다. (이때 ρ 는 반올림된 積에서 LSB값임) 積이 2^n 으로 썸이 되었을 때 入力量的 범위의 %로서 주어진 最大誤差는 $e_{\max} (\%) = \frac{0.5}{2^n - 1} \times 100$ 이 된다⁽⁷⁾

그러므로 이 過程의 線늘은 曲線의 가장 큰部分까지 쉬프트되는데 이것은 $\frac{0.5}{2^n - 1}$ 값 以下의 誤差를 維持하기 爲한이며 n 은 2進 小數의 語長이며 n 비트의 rounded logarithm이다. 다음 그림은 up/down 2進계수기를 使用한 近似 2進로가리즘函數發生을 爲한 variable-rate 시스템의 블록 다이어그램이다.

여기서 $\overline{P=Q}$ (active low)는 出力이고 3 state

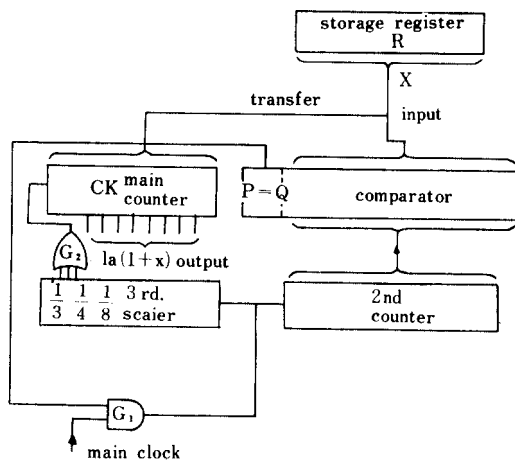


그림 3 $la(1+x)$ 發生을 爲한 variable-rate 시스템의 블록 다이어그램.

이며 Q 入力에는 $20K\Omega$ 의 pull up 抵抗을 使用한다. 이때 8 비트를 使用하면 0.0000000에서 0.11111111까지 $\frac{1}{256}$ 増分스텝이 使用된다. VRC ratio들이 표 2에 있고 표 3의 VR 열에 表示되어 있다. 표 3에 $\log_2(1+x)$ 로 바꾸기 爲해 變換된 x 의 2進값도 計算되었다.

표 2 Count ratio에 對應하는 breakpoint.

segment의 breakpoint	count ratio
$0 \cdot (\frac{1}{256}) \dots\dots 46 \cdot (\frac{1}{256})$	1/3
$46 \cdot (\frac{1}{256}) \dots\dots 94 \cdot (\frac{1}{256})$	1/8
$94 \cdot (\frac{1}{256}) \dots\dots 1136 \cdot (\frac{1}{256})$	0
$136 \cdot (\frac{1}{256}) \dots\dots 200 \cdot (\frac{1}{256})$	1/8
$200 \cdot (\frac{1}{256}) \dots\dots 255 \cdot (\frac{1}{256})$	1/4

여기서 breakpoint를 찾아내기 爲한 方程式은 다음과 같다.

$$B_1 = \overline{X}, \overline{X}, X, \overline{X}, X, X, X, \overline{X}, \dots\dots (46)$$

$$B_2 = \overline{X}, X, \overline{X}, X, X, X, X, \overline{X}, \dots\dots (94)$$

$$B_3 = X, \overline{X}, \overline{X}, \overline{X}, X, \overline{X}, \overline{X}, \overline{X}, \dots\dots (136)$$

$$B_4 = X, X, \overline{X}, \overline{X}, X, \overline{X}, \overline{X}, \overline{X}, \dots\dots (200)$$

$$B_5 = \overline{X}, \overline{X}, \overline{X}, \overline{X}, \overline{X}, \overline{X}, \overline{X}, \overline{X}, \dots\dots (0)$$

breakpoint는 기숫기기 서로 다른 구간적 직선들이 만나는 교점인데 찾아내기가 쉽지 않다. 일반적으로 n 이 8 비트라고 하자. 첫째, differential group의 數(Number of differential Group)를 計算한다. 즉,

표 3 X와 $\log_2(1+X)$ 간의 변환리스트, $0 \leq X < 1$.

Decimal 2^{-n}	X = B	$\log_2(1+X) = A$	A - B	True error 2^{-n}	VR
0	0.00000000	0.00000000	0	0.000	0
1	0.00000001	0.00000001	0	+0.440	↑
2	0.00000010	0.00000011	+1	-0.130	1
3	0.00000011	0.00000100	+1	+0.300	↓
4	0.00000100	0.00000110	+2	-0.270	2
⋮	⋮	⋮	⋮	⋮	⋮
44	0.00101100	0.00111011	+15	-0.422	15
45	0.00101101	0.00111100	+15	-0.193	↓
46	0.00101110	0.00111101	+15	+0.031	↑
47	0.00101111	0.00111110	+15	+0.252	↓
48*	0.00110000	0.00111111	+15	+0.470	↓
49	0.00110001	0.01000001	+16	-0.318	16
50	0.00110010	0.01000010	+16	-0.110	↓
51	0.00110011	0.01000011	+16	+0.096	↓
52	0.00110100	0.01000100	+16	+0.297	↓
53	0.00110101	0.01000101	+16	+0.495	↓
54*	0.00110110	0.01000111	+17	-0.313	↑
55*	0.00110111	0.01001000	+17	-0.123	↓
⋮	⋮	⋮	⋮	⋮	⋮
135	⋮	⋮	⋮	⋮	22
136	0.10001000	0.10011101	+21	+0.365	↓
137	0.10001001	0.10011110	+21	+0.306	↑
138	0.10001010	0.10011111	+21	+0.245	↑
139	0.10001011	0.10100000	+21	+0.181	↑
140	0.10001100	0.10100001	+21	+0.115	↑
141	0.10001101	0.10100010	+21	+0.047	21
142	0.10001110	0.10100011	+21	-0.025	↓
143	0.10001111	0.10100100	+21	-0.100	↓
144	0.10010000	0.10100101	+21	-0.173	↓
145	0.10010001	0.10100110	+21	-0.250	↑
146	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
251	0.11111011	0.11111100	+01	+0.375	↓
252	0.11111100	0.11111101	+01	+0.103	↓
253	0.11111101	0.11111110	+01	-0.170	↑
254	0.11111110	0.11111111	+01	-0.445	0
255	0.11111111	0.11111111	+00	+0.278	↓

$$NDG = ([\log_2(1+x) - x]_{\max} / 1 / (2^n - 1)) + 1$$

$$(0 \leq x < 1)$$

둘째, 方程式에 따른 breakpoint의 位置를 發見한다. 즉,

$$\log_2(1+x_i) - x_i = G_i \quad (0 \leq x_i < 1)$$

G_i 는 各 스텝에 대한 差異값으로 $G_i = K_i \Delta$ 가 되며 K 는 NDG 계산에서 얻은 값이고 Δ 는 LSB 값인 $1/2^8 = \frac{1}{256}$ 이다.

이 方程式은 rate에서 位置變化를 發見하는데 使用되며 scaler와 比較器로 構成한다. 표 4는 8 비트 VRC에 對한 $\log_2(1+x) - x$ 의 breakpoint의 위치를 나타낸다.

count ratio $\frac{1}{3}$ 은 다음과 같이 構成할 수 있다.

$$\begin{matrix} \text{여기서} & (J_a \leftarrow Q_b) & (K_a) \leftarrow Q_b \\ & (J_b \leftarrow Q_a) & (K_b) \leftarrow 1 \end{matrix}$$

이고 다른 ratio들은 4와 8로 나눔으로써 쉽게 얻어질 수 있다. count ratio에 對應하는 breakpoint는 이미 표 2에서 言及하였다. 다음에는 完全한 시스텝 動作을 알아본다.

표 4 8 비트 Programmable Variable-rate up/down 카운터를 사용한 $\log_2(1+X) - X$ 의 break point 위치.

Range of X ΔX_i	Break point occurrence $X_{i+1} = X_i + \Delta X_i$	Binary eqn.	Fraction	VR count ratio
$\Delta X_0 = 0.00887$	$X_1 = 0.00887$	0.00000010	2/256	1/3
$\Delta X_1 = 0.00914$	$X_2 = 0.01802$	0.00000100	4.63/256	1/3
$\Delta X_2 = 0.00942$	$X_3 = 0.02744$	0.00000110	6.0/256	1/3
$\Delta X_3 = 0.00942$	$X_4 = 0.02744$	0.00000110	9.6/256	1/
⋮	⋮	⋮	⋮	⋮
$\Delta X_{12} = 0.01427$	$X_{13} = 0.14483$	0.00100110	37.8/256	1/3
$\Delta X_{13} = 0.01515$	$X_{14} = 0.15990$	0.00101010	40.9/256	1/3
$\Delta X_{14} = 0.01618$	$X_{15} = 0.17600$	0.00101101	45.0/256	1/3
$\Delta X_{15} = 0.01740$	$X_{16} = 0.19340$	0.00110000	49.5/256	1/8
$\Delta X_{16} = 0.01890$	$X_{17} = 0.21230$	0.001110000	56.0/256	1/8
⋮	⋮	⋮	⋮	⋮
$\Delta X_{20} = 0.03194$	$X_{21} = 0.31504$	0.01010000	80.65/256	1/8
$\Delta X_{21} = 0.04110$	$X_{22} = 0.35615$	0.01011011	91.1/256	0
$\Delta X_{22} = 0.06760$	$X_{23} = 0.44260$	0.01110001	113.3/256	1/8
$\Delta X_{23} = 0.06760$	$X_{24} = 0.52420$	0.10000110	134.20/256	1/8
$\Delta X_{24} = 0.06500$	$X_{25} = 0.58920$	0.10010110	150.8/256	1/8
$\Delta X_{25} = 0.03964$	$X_{26} = 0.62884$	0.10100000	160.9/256	1/8
$\Delta X_{26} = 0.03242$	$X_{27} = 0.66126$	0.10101001	169/256	1/8
⋮	⋮	⋮	⋮	⋮
$\Delta X_{42} = 0.01466$	$X_{43} = 0.96743$	0.11111000	248/256	1/4
$\Delta X_{43} = 0.01437$	$X_{44} = 0.98180$	0.11111011	251/256	1/4
$\Delta X_{44} = 0.01409$	$X_{45} = 0.99589$	0.11111111	255/256	1/4

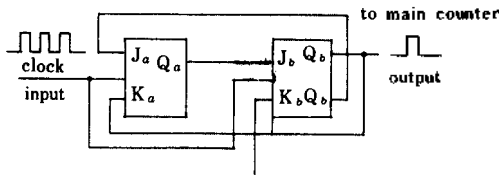


그림 4 $\frac{1}{3}$ scaler의 logic diagram.

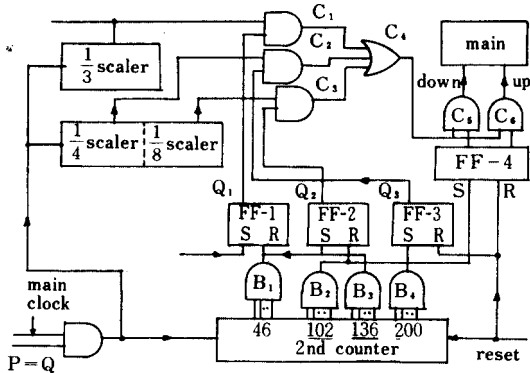


그림 5 2nd계수기controller와 scaler의 細部的 回路.

(1) 초기신호가 FF-1을 initialize 할 때에 $\frac{1}{3}$ scaler는 enable되며 다른 scaler들은 FF-2와 FF-3에 의해서 inhibit된다. 2nd 계수기와 scaler는 세이트G₁을 통해 동시에 clock된다. 이 순간에 比較器의 $\overline{P=Q}$ 핀은 high(즉 P ≠ Q)이다. 主(main) 계수기는 이미 storage 레지스터 R로부터 transfer된 X값을 가지고 있다. 그러므로 main 계수기는 OR게이트 G₂를 통해 count ratio $\frac{1}{3}$ 로 X값으로부터 counting을 시작할 것이다. 이 counting은 다음과 같은 條件들이 發生할 때까지 繼續될 것이다.

- ㉔ breakpoint에 到達해서 count ratio가 $\frac{1}{8}$, 0등으로 바뀔 때
- ㉕ 2nd 계수기의 내용이 0에서 부터 始作하여 x값으로 增加할 때, 比較器의 出力핀 $\overline{P=Q}(x)$ 는 low가 되고 게이트 G₁을 통해 clock을 inhi-

bit한다. 모든 계수기는 스톱되고 그 結果인 近似 2進로가리즘이 main 계수기의 出力으로부터 얻어질 때.

(2) Breakpoint인 46, 94, 136, 200 높은 3개의 Flip Flop과 scalers에 의해서 可能하다. 制御方程式을 使用함으로써, breakpoint들은 4개의 8入力 AND게이트와 1개의 比較器에 의해서 알 수 있다.

(3) FF-1은 첫번째 breakpoint인 $B_1 = \frac{46}{256}$ 에 到達할 때까지 초기신호에 의해서 세트될 것이다. 게이트 B₁의 出力은 FF-1을 리세트시키고 同時に FF-2를 세트시킬 것이다. 게이트 C₁이 enable되고 C₂와 C₃는 disable된다. 그러므로 main 계수기는 두번째 breakpoint $B_2 = \frac{94}{256}$ 에 到達할 때까지 OR게이트 C₁를 통해서 $\frac{1}{8}$ scaler에 의해서 clock될 것이다.

(4) 게이트 B₂의 出力은 FF-2를 리세트시키고 FF-4를 세트시키는데 FF-4는 main 계수기로 하여금 카운트 方向을 轉換하게 한다. 하지만 게이트 C₁, C₂, C₃가 FF-1부터 FF-3까지에 의해서 各各 inhibit되는 것과 같이 main 계수기는 차례로 unlock된다. 반면 2nd 계수기는 세번째 breakpoint인 $B_3 = \frac{136}{256}$ 에 到達할 때까지 main클러에 의해서 아직까지 clock되어 있다.

(5) 類似하게 게이트 B₃의 出力은 FF-2를 세트시킬 것이다. main 계수기는 $B_4 = \frac{200}{256}$ 에 到達할 때까지 $\frac{1}{8}$ scaler에 의해서 다시 clock될 것이다. 하지만 FF-4가 그 前에 세트되어 왔다는 事實에 起因하여 counting down direction이다.

(6) B₄ 出力은 FF-3를 세트하고 main 계수기를 카운트하게 한다. ($\frac{1}{4}$ scaler에 의해 clock된). 出力 B₁과 B₃는 OR wired되었다. 全体 시스템은 어떤 時間에라도 停止될 수 있고, 어떤 breakpoint 사이에 있을 때라도 停止될 수 있다. 만일 2nd 계수기의 내용이 x값으로 增加한다면 이 瞬間에 比較器의핀

出力 $P = \overline{Q}$ 는 low이고 main 클러킹 inhibit 된다.

만일 $X < 0.5$ 라면 시스템은 過去처럼 動作하고, $x \geq 0.5$ 라면 2nd 카운터를 補數化(complement) 하고 count down 한다.

breakpoint B_1 와 B_2 에서 zero 값을 通해서 counting down 한다. 全体 시스템은 두번째 계수기의 내용이 X로 減少되자마자 역시 停止될 수 있다. 계수기의 增加 및 減少는 計算의 實際誤差보다 크거나 또는 작은 값에 到達할 수 있다.

即 表 3에서 最大 正의 誤差는 $VR > (A - B)$ 일 때 發生하고 最大 負의 誤差는 $VR < (A - B)$ 일 때 發生한다. 여기서 最大 正의 誤差는 $\frac{+0.97}{2^8 - 1}$ ($x = 46 \left(\frac{1}{256} \right)$ 에서 發生)이고 最大 負의 誤差는 $\frac{-1.193}{2^8 - 1}$ ($x = 161 \left(\frac{1}{256} \right)$ 에서 發生)이다. 그러므로 全体 誤差 $(E)_{\max} = \frac{2.163}{256} = 0.0848$ 이다. 만일 10비트 레지스터가 使用된다면 全体 最大誤差는 약 $(E)_{\max} = 0.002114$ 가 될 것이다. 따라서 正確性은 factor $1.892 (0.0040 / 0.002114)$ 에 의한 Marino의 2 계近似值 計算보다 더욱 좋다. Marino의 論文에서는 하드웨어기 記述되어 있지 않다. 거기에서는 두 가지 方法에서 使用된 하드웨어의 複雜性을 比較할 方法이 없다. 이 方法은 Marino의 알고리즘과 比較하여 $\frac{1}{1.892}$ 의 比로 誤差를 減少시킬 수 있었다. 하드웨어 역시 산기등이 不必要한 簡單한 方法을 使用하였다. 이 方法의 短點은 스피드가 使用된 main 클러킹의 주파수에 의존한다는 것이다. 構成된 最大 클러킹 주파수는 30MHz이고 LS 소자였다.

IV. 結 論

이 論文은 2進 logarithmic 函數發生에 適合하다. rate가 line segment의 기울기에 比例하는 프로그램

래머블 계수기를 使用함으로써 信號處理가 可能한 2進 로가리즘의 回路를 簡單化하고 誤差를 減少시킬 수 있었다. 로가리즘의 잇점은 곱셈, 나눗셈, 기승제곱, 루트 등을 爲한 빠른 計算에 있다. 그러나 로가리즘을 使用하는 計算은 結果적으로 誤差를 發生한다. 만일 修正되지 않으면 이 方法의 計算에 制限을 받게 된다. 로가리즘은 범용컴퓨터의 기존 연산 方法이나 정수계산에는 適合하지 않다. 이 로가리즘은 패턴인식, 디지털 영상개선, 레이더처리, 음성컴퓨터링과 같이 한성되고 特殊한 應用分野의 特殊 用 프로세서에 適合하다.

참 고 문 헌

1. Mitchell, J.N. Jr.: 'Computer multiplication and division using binary logarithms,' IRE Trans., 1962, EC-11, pp. 512-517.
2. Frangakis, G.P.: 'Fast binary logarithm computing circuit for binary numbers less than one,' Electron. Lett., 1980, Vol 16, No 15, pp. 574-575.
3. Hall, E.L., Lynch, D.D, and Dwyer, S.J.: 'Generation of products and quotients using approximate binary logarithms for digital filtering applications,' IEEE Trans on Comput 1970, C-19, pp. 97-105.
4. Marino, D.: 'New algorithms for the approximate evaluation in hardware of binary logarithms and elementary functions,' IEEE Trans on Comput, 1972, C-21, pp. 1416-1421.
5. Lo, H.Y.: 'Binary logarithms for computing integral and non-integral roots and powers,' Int. J. Electron., 1976, Vol. 40, No. 4, pp. 357-364.
6. Combet, M., Van Zonneveld, H., and Verbeek, L.: 'Computation of the base-two logarithm of a binary numbers,' IEEE Trans., 1965, EC-14, pp. 863-867.
7. Thomas A Brubaker and John C. Becker.: 'Multiplication using Logarithms Implemented with Read-Only-Memory,' IEEE Trans on Comput., 1975, C-24, pp. 761-765.