

論 文

음성인식용 DTW PE의 IC화를 위한
ADD 및 ABS 회로의 설계

準會員 鄭 光 載* 正會員 文 洪 眞** 正會員 崔 圭 勳*** 正會員 金 鐘 玟*

An Integrated ADD and ABS Circuits Design
of DTW PE for Speech Recognition

Kwang Jae JEONG*, Hong Chin MOON**, Kyu Hoon CHOI***

Chong Kyo KIM* *Regular Members*

要 約 음성 인식에 있어서 계산 속도를 보다 빠르게 하기 위한 여러 방법중에 systolic array를 이용하여 multiple processing 하는 방법이 있다. 이러한 systolic array의 각 연산은 pipeline 방식으로 수행되며, 이 연산은 Processing Element (PE)로 multiprocessing되어 계산 효율을 증대시키게 된다.

이 DTW용 PE cell은 크게 세가지의 블록으로 대별된다. 즉 MIN block, ADD block 그리고 ABS block인데, 본 연구에서는 이들 세가지 블록중 "ADD"와 "ABS" 대한 회로설계및 검증을 행하였으며, 3 μ m CMOS N-well 설계 규칙에 따라 두 블록에 대한 각각의 레이아웃과 전체 레이아웃을 설계한 후 설계 규칙 검사(DRC)를 마쳤다.

ABSTRACT There are many methods for speed up counting in speech recognition. A multiple processing method is the one way to achieve the aim using systolic array. This arithmetic operation by the array is achieved pipelining skill. And the operation is multiprocessed by processing element(PE) that is increasing counting efficiencies

The DTW PE cell is seperated into three large blocks. "MIN" is the one block for counting accumulated minimum distance, "ADD"block calculated these minimum distances, and "ABS" seeks for the absolut values to the total sum of local distances.

We have accomplished circuit design and verification about the "ADD" and "ABS" blocks, and performed total layout & DRC(design rule check) using 3 μ m CMOS N-Well rule base.

I. 서 론

DTW(dynamic time warp)를 이용한 음성

인식이란 미지음성 신호에서 그것이 다른 발음들과 구별될 수 있는 중요한 특징을 추출하고 이러한 방법으로 미리 만들어서 기억시켜 놓은 기준 특징들의 패턴들과 비교하여 그 차이가 가장 적은 것으로 해당 음성을 인식하는 것이다.

이러한 음성 인식 과정에서의 문제점으로는 기억시켜둔 기준 패턴이 많을수록 인식할 수 있는 어휘는 늘어나지만 인식하려는 입력 패턴을

*全北大學校 電子工學科
Dept. of Electronic Engineering Chonbuk National University
**群山實業專門大學 電子計算科
***全州工業專門大學 電子科
論文番號 : 90-65 (接受 1990. 5. 21)

기준 패턴 전체와 비교하여야 하므로 계산 시간이 길어지는 점이다. 또한 많은 어휘를 인식하기 위하여 기준 패턴 수를 증가시키려고 할 때에는 이들을 저장할 메모리의 양이 많아진다는 점이다^{(2)(X10)}.

음성 인식에 있어서 계산 속도를 보다 빠르게 하기 위한 여러 방법중에는 systolic array를 이용하여 multiple processing하는 방법이 있다. 이러한 systolic array의 각 연산은 pipeline 방식으로 수행되며, 이 연산은 Processing Element (PE)로 multiprocessing 되어 계산 효율을 증대시키게 된다⁽²⁾⁽⁴⁾⁽⁸⁾.

음성 인식에서의 DTW는 반복적 계산을 필요로 하며, 이 계산을 수행하기에 합당한 PE cell의 설계는 매우 중요하다. 따라서 이 연구에서는 실제에 가까운 실시간 어휘 인식을 가능하게 하는 large dictionary의 DTW 알고리즘을 하드웨어로 구현하기 위한 PE cell의 설계에 주안점을 두었다.

이 DTW용 PE cell은 크게 세가지의 블록으로 대별된다. 즉, MIN block, ADD block 그리고 ABS block인데, "MIN"은 축적된 최소거리를 계산하기 위한 블록이고 "ADD"는 이들 최소거리들의 합을 계산하는 블록, 그리고 "ABS"는 이러한 합에 의한 local distance의 절대값을 구하기 위한 블록이다.

이들 각 PE들은 종래의 systolic DTW 방식과는 다르게 digit pipelined 방식으로 array됨으로써 1) 모든 데이터들의 base4 format signed-digit로 표현되고 2) 이러한 데이터들은 digit serial 방식으로 PE 사이를 통과하게 되어서 3) 패턴 매칭이 디지털레벨에서 pipeline 방식으로 이루어지게 된다. 이러한 방식이 도입됨에 따라 아주 높은 동시성과 데이터율을 유지하게 한다⁽¹⁾⁽⁶⁾.

본 연구에서는 이들 세가지 블록중 "ADD"와 "ABS" 블록에 대한 회로설계 및 검증을 행하였으며, 3 μ m CMOS N-well 설계 규칙에 따라 두가지 블록에 대한 각각의 레이아웃과 전체 레이아웃을 설계한 후 설계 규칙 검사(DRC)

를 마쳤다.

II. 디지털 pipelined DTW PE

DTW PE는 대수적 성분 블록으로 형성되어 있다. DTW 프로세서에서의 각 PE는 (1)식에 의하여 계산되는데 이와 같은 하나의 PE에 대한 내부 블록다이아그램은 그림 1과 같다.

$$\begin{aligned} S_{i,j} &= \text{MIN}(S_{i-1,j} + D_{i,j}, S_{i-1,j-1} + 2D_{i,j}, S_{i,j-1} + D_{i,j}) \\ &= \text{MIN}(\text{MIN}(\text{ADD}(S_{i-1,j}, \text{ABS}(u_i, r_j)), \text{ADD}(S_{i,j-1}, \text{ABS}(u_i, r_j))), \text{ADD}(S_{i-1,j-1}, \text{ADD}(\text{ABS}(u_i, r_j), \text{ABS}(u_i, r_j)))) \end{aligned} \quad (1)$$

각 PE는 두 개의 MIN 성분, 네개의 ADD 성분, 그리고 한 개의 ABS 성분을 포함한다. 그리고 이러한 PE는 s_{out} 디지털 값을 래칭시키기 위한 한 개의 3비트 레지스터와 r_{out} 과 u_{out} 의 디지털 값을 래칭시키기 위한 두 개의 2비트 레지스터도 포함한다.

파형 제어 신호 W_{in} 은 모든 PE들 사이를 역 대각선 방향으로 동일하게 전파되지만 PE를 통하여 래칭되지 아니한다.

각 PE에 대한 clocking scheme은 그림2에서 제시하고 있다. PE를 통하여 한 개의 디지털가 완전하게 처리되는데는 two-phase clock으로 네 개의 펄스가 필요로 된다.

파형 제어 펄스 W_k 는 다음 번째 계수의 msd가 도착될 때 다시금 발생할 것이다. 한 개의 디지털는 네 개의 펄스를 필요로 하므로 serial하게 8개의 디지털들을 처리하려면 32개의 클럭 펄스가 요구된다. 즉, 8디지털 오퍼랜드는 32개의 펄스로 처리되며, 이 경우 W_k 는 32개의 펄스마다 다시 발생된다고 할 수도 있다.

한 디지털다 소요되는 펄스의 수(:4 클럭)는 처리 속도의 개선과 관계되는데, 이러한 개선은 단일 PE내의 원래 동작을 'pipelining' 함으로

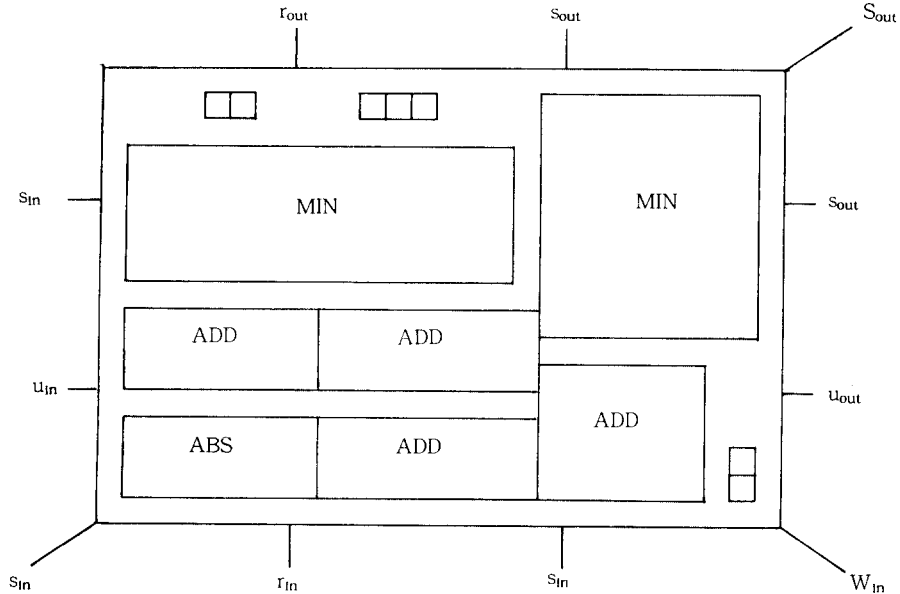


그림 1. DTW PE의 블록다이아그램

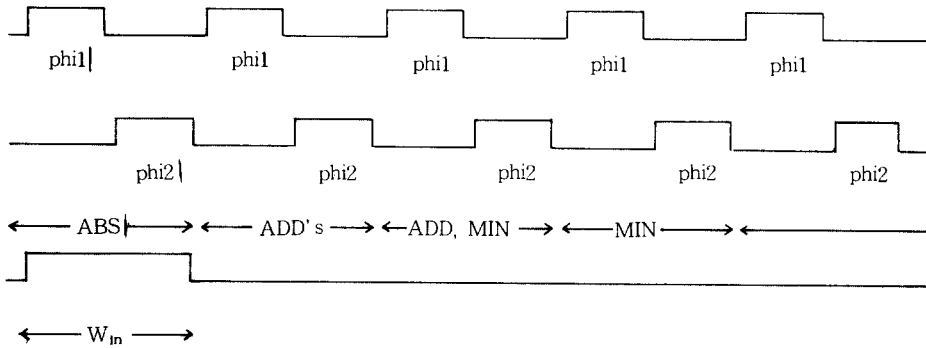


그림 2. PE clocking scheme

써 가능하다.

즉, 개선율의 증가는 각 PE 내부의 네 단계 스텝($\Leftarrow \text{ABS} \Rightarrow \Leftarrow \text{ADD's} \Rightarrow \Leftarrow \text{ADD, MIN} \Rightarrow \Leftarrow \text{MIN} \Rightarrow$)과 디지털 레지스터들을 하드웨어적으로 분리시킴으로써 가능하다고 할 수 있다.

그림 2에서 한 개의 게이트 지연 시간이 2ns 인 경우, 각 phase($\Leftarrow \text{ph1}$ 또는 ph2)는 네 개의

게이트 로직을 가지므로 8ns의 시간이 걸린다.

예를 들어서 그림8의 ADD 로직 블록다이아그램에서 첫 번째 디지털 동작(1c, is)은 ph1 에서, 두 번째 디지털 동작 (2s)은 ph2 중심의 적절한 클럭 사이클동안에 각각 수행된다.

두 phase 구분에 사이마다 1ns의 시간이 허용된다면 전체 phase 구분에 허용되는 시간이 8

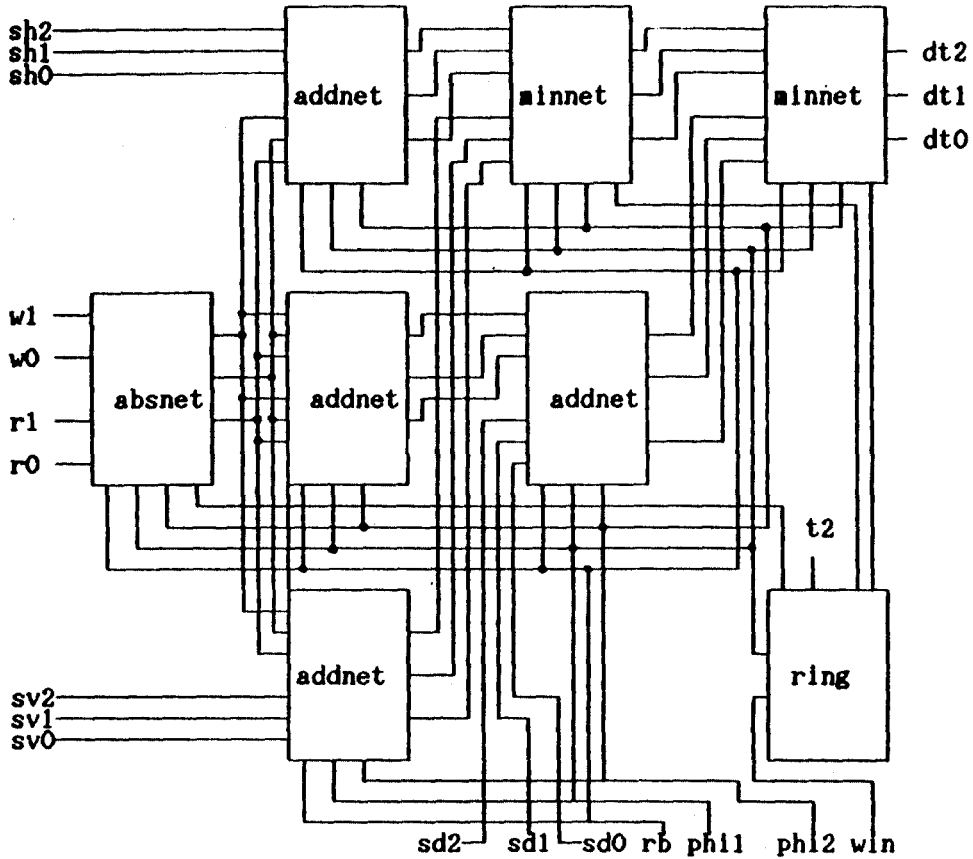


그림 3. 완성된 전체 DTW PE 블록다이아그램

ns이 되고, PE 하나가 8개의 디지털 오퍼랜드인 경우 $8\text{digits} \times 8\text{ns} = 64\text{ns}$ 이므로 결국 PE 사이클 시간은 $8 + 64 = 72\text{ns}$ 인 셈이 된다⁽¹⁾⁽⁶⁾⁽⁷⁾.

이 연구에서의 대수적 성분 블록들은 모두 digit pipeline되기 때문에 현재의 PE 사이클 수행 결과가 이웃한 PE로 전달될 동안 이어지는 다음의 PE 사이클 시간 중의 다음 디지털 오퍼랜드가 현재의 PE 사이클시간에 의하여 처리된다.

그림 3은 실제 회로 설계를 마친 후 완성된 전체 DTW PE 블록다이아그램 회로이다.

III. PE 블록 설계

1. ADD 블록 설계

그림 4에 나타낸 바와 같이 디지털 직렬 ADD 성분은 축적된 최소 거리 $S_{i,j}$ 를 계산하기 위하여 사용된다.

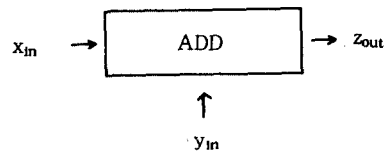


그림 4. 디지털 직렬 가산기

그림의 입력 x_{in} 은 두 패턴 성분 차의 절대값에 의해 계산된 국부거리 $D_{i,j}$ 의 디지털 값이

고, y_{in} 은 인접 PE에 의해 계산된 최소 축적 거리의 디지털 값이다. 그림 4의 현재 중간 sum 디지털 s_j 는 (2)식으로, 현재의 중간 carry 디지털 c_j 는 (3)식으로 각각 표현된다.

$$s_j = x_{in} + y_{in} - 4c_j \quad (2)$$

여기서, c_j 는 $-2 \leq s_j \leq 2$ 되도록 하기 위하여 $-1 \leq c_j \leq 1$

$$z_{out} = s_{j-1} + c_j \quad (3)$$

(2)식의 첫 디지털 동작에 의하여 결정된 s_j 는 다음 주기에서 이용되기 위하여 내부 래치에 일시 저장된다. (3)식의 두 번째 디지털 동작에서 현재의 c_j 와 래치에 저장되어 있던 직전의 sum s_{j-1} 과 합해진다. 이러한 가산은 원래 제한된 디지털 set이 중간 carry와 sum을 허용하였기 때문에 'carry free'되도록 설계된다.

표1은 (2)식에 의한 첫 디지털 동작에 대한 ADD mapping table이다.

표 1. ADD operation mapping

$x_{in} + y_{in}$	c_j	s_j
-6	-1	-2
-5	-1	-1
-4	-1	0
-3	-1	1
-2	0	-2
-1	0	-1
0	0	0
1	0	1
2	0	2
3	1	-1
4	1	0
5	1	1
6	1	2

(3)식의 두 번째 디지털 가산 성분은 두 개의 디지털 가산기 셀을 필요로 한다. 그림 5는 ADD

블록내의 두 디지털 가산기 셀을 보여주고 있다.

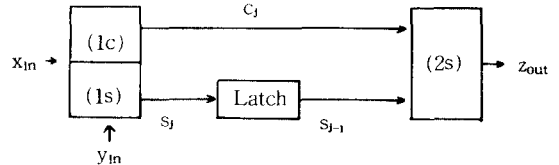


그림 5. ADD logic 블록다이아그램

첫 디지털 가산기 셀은 1s와 1c로 구성되어 있으며, 두 번째 디지털 가산기 셀은 2s만으로 이루어진 이유는 두 번째 디지털 carry 셀 2c는 z_{out} 상에서 'carry free'하기 때문에 2c가 생략된 셈이다.

표2는 base 4 직렬 가산의 예를 보여준다. 표에서 $x_{in} = 1223_4, 1232_4 (= 1112_4)$ 의 순서로, $y_{in} = 1213_4, 1121_4 (= 1113_4)$ 의 순서로 각각 입력된 경우 앞에서의 설명에 따라 $z_{out} = 3102_4, 3111_4 (= 2231_4)$ 로 출력된다. 표의 "*"는 계산에서 사용되지 않는 값을 의미한다.

표 2. 디지털 직렬 가산의 예시

clock	1	2	3	4	5	6	7	8	9
x_{in}	1	2	2	3	1	2	-3	2	*
y_{in}	1	2	1	3	1	1	2	-1	*
c_j	0	1	1	1	0	1	0	0	0
s_j	2	0	-1	2	2	-1	-1	1	*
z_{out}	*	3	1	0	2	3	-1	-1	1

가산 순차를 위하여 다음 가산을 위한 오퍼랜드의 msd(most significant digit)값은 현재의 가산을 위한 오퍼랜드의 lsd(least significant digit) 값이 입력된 후 다음 클럭에서 입력으로 된다. Overflow가 허용되지 않는 한 다음 가산을 위한 (2)식의 동작에서 형성된 중간적인 carry는 'zero' 되도록 조정된다. 이러한 중간 carry는 현재의 가산을 완성시키기 위해 (3)식의 동작

에서 이용된다. 따라서 만일 이러한 carry가 zero가 아니라면 현재의 가산에서 부정확한 1sd를 출력시킬 수도 있다.

표 3. Round down ADD operation mapping

$x_{in} + y_{in}$	c_j	s_j
-6	0	-2
-5	0	-2
-4	0	-2
-3	0	-2
-2	0	-2
-1	0	-1
0	0	0
1	0	1
2	0	2
3	0	2
4	0	2
5	0	2
6	0	2

원래의 DTW 프로세서 연산 성분은 연산중개 결과 overflow가 전혀 발생되지 않도록 가상하여 설계되어 있다. Overflow가 발생되면 앞의 연산중개 결과의 1sd는 혼란에 빠진다. 이러한 상황은 $S_{i,j}$ 계산에서 쓰이는 ADD 성분 동작에서 msd overflow를 'corrupt' 시킴으로써 피할 수

있다. DTW 프로세서는 가장 근접한 매치(; the smallest match factor)를 찾기 때문에 이러한 결과는 어떠한 경우에도 적용되는 것은 아니므로 언제나 정확한 결과가 요구되지는 않는다.

표3에서 보여주는 디지털 매핑을 이용함으로써 overflow를 동시에 제한할 수 있다. 즉, (2)식의 디지털 동작에서 msd가 생성될 경우에만 모든 carry는 zero로 조정된다.

그림 6은 지금까지 설명한 알고리즘을 실제 회로로 구현한 전체 ADD 회로에 대한 블록다이아그램이다. 그림의 MPWADD 블록은 그림 5의 두 입력 x_{in} , y_{in} 의 3비트 가산기이고, ADDMAP 블록은 그림 5의 s_j , c_j 를 얻는 회로이며, 2DLA 블록은 s_{j-1} 을 얻기 위한 s_j 의 래치, 그리고 MPWADD1은 overflow가 생략된(; 'carry free') c_j , s_{j-1} 의 합을 구하기 위한 3비트 가산기로서 3비트 z_{out} 출력을 얻도록 설계되어 있다. 입력 측에는 D-FF 래치 회로를 부가하여 회로의 안정을 도모하였으며, 전체의 동작은 4단계로 이루어진다⁽⁴⁾⁽¹²⁾.

2. ABS 블록 설계

그림 7에 소개된 디지털 직렬방식의 절대값 성분 블록(; ABS block)은 국부 거리 $D_{i,j}$ 를 계산하는데 이용된다. 그림의 입력 $x_{in}=u_{in}$ 과 $y_{in}=r_{in}$ 은 전형적인 디지털 set{0, 1, 2, 3}으로

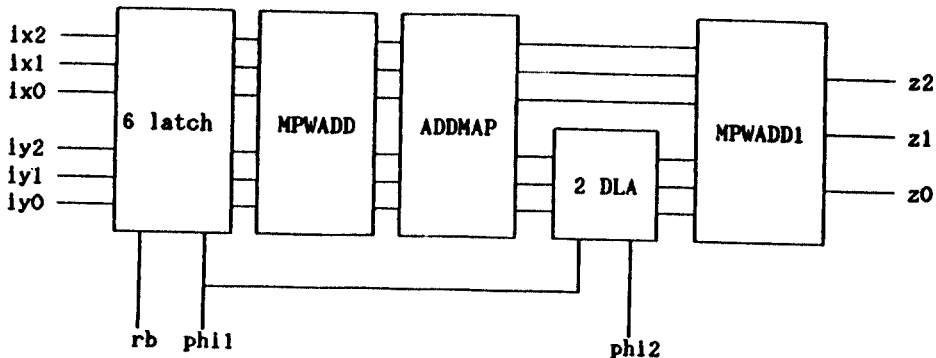


그림 6. ADD 회로 전체의 블록다이아그램

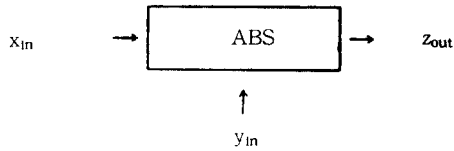


그림 7. 1비트 직렬 절대치 감산기

형성되어 있다.

ABS 성분의 기능은 앞서 설명한 ADD 성분과 아주 비슷하다. 그림에서의 첫번째 디지털 동작은 (4)식과 같고, 두번째 디지털 동작은 (5)식과 같다.

$$S_j = x_{in} - y_{in} - 4c_j \quad (4)$$

여기서 c_j 의 범위는 $-2 \leq s_j \leq 2$ 가 되도록 $-1 \leq c_j \leq 1$ 이어야 한다.

$$z_{out} = |s_{j-1} + c_j| \quad (5)$$

표 4는 식(4)의 디지털 동작을 매핑한 표이다. (5)식의 디지털 동작은 msd의 부호에 의한다. 따라서 절대값의 계산은 msd 우선(msd first) 입력을 필요로 하는 동작이다.

만일 msd가 양(positive)이면 두번째 디지털 동작은 ADD에서와 같이 단순한 'carry free' 디지털 가산이 되고, msd가 음(negative)이면 이 디지털 동작의 결과는 해당 디지털과 모든 일련의 디지털에 대하여 2의 보수를 취하여야 할 것이다.

표 4. ABS operation mapping

$x_{in} - y_{in}$	c_j	s_j
-3	-1	1
-2	0	-2
-1	0	-1
0	0	0
1	0	1
2	0	2
3	1	-1

그림 8은 전체적인 ABS 동작 상태를 흐름선도로 제시한다. 그림의 ㉠, ㉢ 및 ㉣ 세가지 상태 각각에 대해 괄호안의 내용은 다음 출력 디지털 값을 보여준다. 파형 제어 신호 W_{in} 은 최종 상태를 상태 ㉠로 초기화 시키는데 이용된다. ABS 동작 상태는 그림 5의 ADD logic 블록 다이어그램과 견주어 파악하면 ADD와의 유사성을 발견하게 된다.

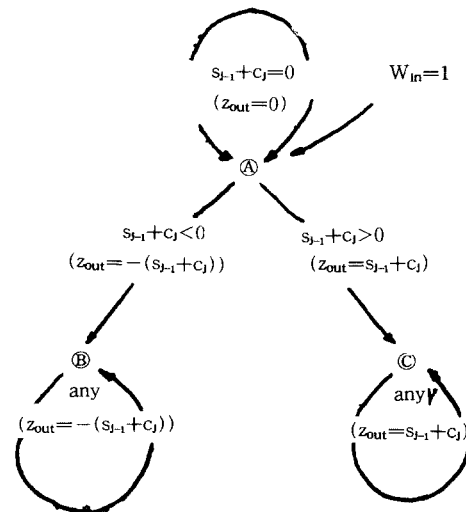


그림 8. 전체적인 ABS 동작 상태

예를 들어서 $s_{j-1} + c_j = 1232_4 (= -0102_4) < 0$ 일 때 절대값의 출력은 2의 보수를 취하여 $z_{out} = 1232_4 (= 0102_4)$ 가 될 것이다.

그림 9는 전체 ABS회로의 블록다이어그램이다. 전체의 구성은 (4)식의 $x_{in} - y_{in}$ 을 수행하기 위한 borrow looked 방식의 subtractor인 BLB 블록, BLB 출력에 의한 mapper에 따라 s_j 와 c_j 가 생성되는 ABSMAP 블록, s_j 의 s_{j-1} 을 위한 래치용 1DLA 블록, $z_{out} = s_{j-1} + c_j$ 를 구하기 위한 MPWADD2 블록, z_{out} 이 negative일 경우 2의 보수를 취하기 위한 NEGATOR 블록, 그리고 INSEL 블록의 제어조건에 따라 최종 출력을 선택하는 2MUX 블록으로 구성되어 있다⁽²⁾.

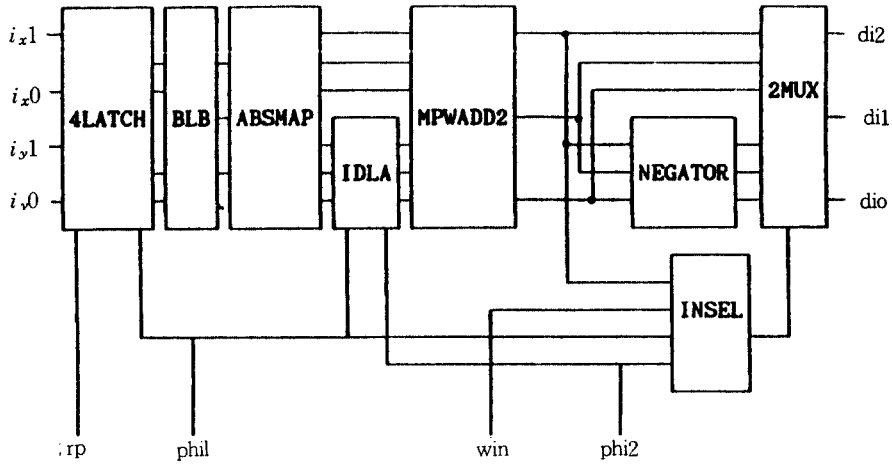


그림 9. 전체 ABS회로 블록다이아그램

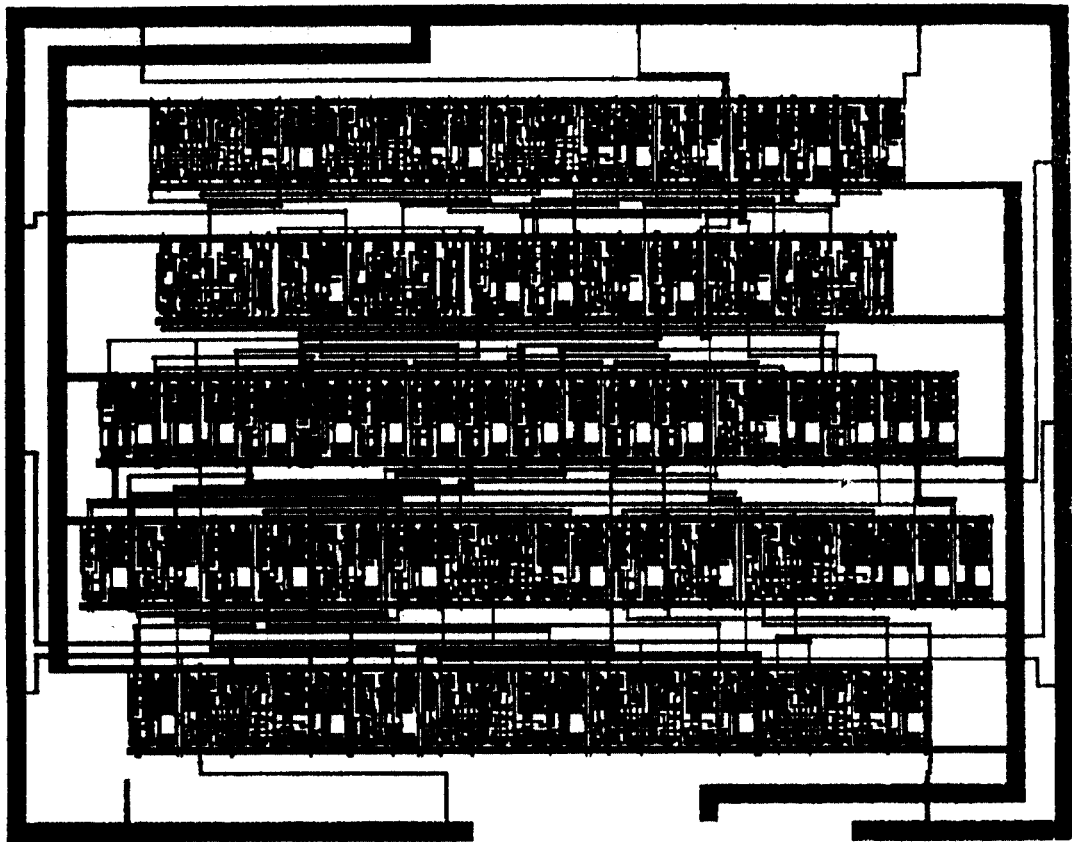


그림 10. ADD 회로의 레이아웃

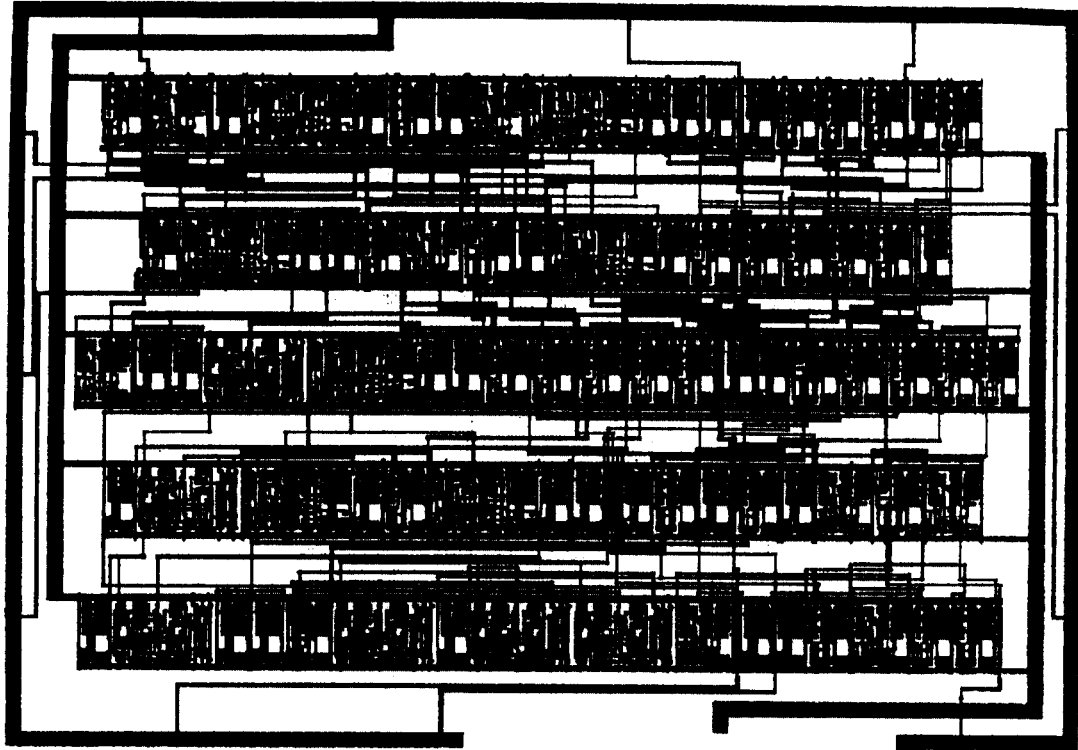


그림 11. ABS 회로의 레이아웃

IV. 레이아웃

설계된 ADD 및 ABS 회로에 대해 $3\mu\text{m}$ CMOS N-well 표준셀 설계규칙에 따라 설계한 레이아웃의 결과는 각각 그림 10, 그림 11과 같다. 레이아웃 설계 및 설계규칙검사 (design rule check)는 한국전자통신연구소의 자동설계 tool을 이용하였으며 SUN workstation을 활용하여 도면을 plot하였다.

V. 결 론

본 연구에서는 DTW PE의 블록다이어그램을 중심으로 하여 격리단어인식 문제에 관하여 디지털로 'pipeline'되는 방법에 대해 고찰하였다.

인식에서의 중요한 관점은 실시간 처리를 위한 신속성, 활용성 제고를 위한 경제성, 그리고 편리한 이용을 위한 기능성 등을 들 수 있다. 이러한 현실적 문제의 구현을 위해서는 DTW processor의 개발이 필요한데, 이 프로세서의 Processing Element(PE) 설계 및 one chip layout은 문제해결의 가장 중요한 부분이라고 할 수 있다.

PE회로는 크게 세가지 블록으로 대별되어 ADD, ABS 그리고 MIN 블록으로 나뉜다. 본 연구에서는 이들 세가지 블록중 ADD와 ABS 블록에 대한 회로설계 및 검증을 행하였으며, $3\mu\text{m}$ CMOS N-well 표준셀 설계규칙에 따라 두 블록에 대한 각각의 레이아웃과 전체 레이아웃을 설계한 후 설계 규칙 검사(DRC)를 마쳤다.

MIN 블록 회로는 본 논문의 ADD-ABS 블록

회로와 회로설계를 위한 알고리즘이 전혀 다르고 회로 규모에서도 서로 상이하지만, MIN도 PE 셀의 한 블록으로써 회로설계 및 검증을 위하여 동일한 룰을 이용하였다.⁽¹²⁾

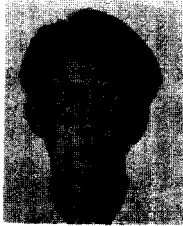
이들 각 PE는 종래의 시스톨릭 DTW 방식과는 달리 디지털 파이프라인 방식으로 배열됨으로써 모든 데이터는 베이스-4 형태 부호와 디지털 (base4 signed digit)로 표현하였으며, 이러한 데이터들은 직렬 디지털-형태로 PE사이를 통과하게 되어서 패턴 매칭이 디지털 레벨에서 파이프라인 방식으로 이루어지도록 설계하였다.

베이스-4 형태 부호화 디지털 방식을 이용함으로써 종래의 병렬처리 방식과는 달리 하드웨어가 간단하여지며, 이 결과 VLSI화가 용이하게 되어 고속처리가 가능하게 됨으로써 이를 이용한 DTW 프로세서가 완성될 경우 실시간 음성 인식이 기대된다.

앞서 발표한 MIN블록의 설계와 함께 완성된 전체 PE cell은 앞으로 wafer process가 진행될 예정이며, 이의 IC화 제작이 끝나면 chip test를 행할 계획이다.

參 考 文 獻

1. M. J. Irwin, "A Digit Pipelined Dynamic Time Warp Processor", IEEE Trans. Acoust., Speech, Signal Processing, vol. 36, No. 9, pp. 1,412~1,422, Sept. 1 988.
2. N. Weste, D. J. Burr, and B. D. Ackland, "Dynamic Time Warp Pattern Matching Using an Integrated Multiprocessing Array", IEEE Trans. on Computers, vol. C-32, No. 8, pp. 731~744, Aug. 1983.
3. C. Myers and L. R. Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition", IEEE Trans., Acoust., Speech, Signal Processing, vol. ASSP-29, No. 2, pp. 284~2 97, April 1981.
4. D. J. Burr, B. D. Ackland, and N. Weste, "Array Configurations for Dynamic Time Warping", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-32, No. 1, pp. 119~127, Feb. 1984.
5. L. R. Rabiner, A. E. Rosenberg, and S. E. Levinson, "Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-26, No. 6, pp. 575~582, Dec. 1978.
6. H. Murveit and R. W. Brodrsen, "An Integrated Circuit Based Speech Recognition System", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-34, No. 6, pp. 1,465~1,472. Dec. 1986.
7. C. Myers, L. R. Rabiner, and A. E. Rosenberg, "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-28, No. 6, pp. 623~635, Dec. 1980.
8. F. Charot, P. Frison, and P. Quinton, "Systolic Architectures for Connected Speech Recognition", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-34, No. 4, Aug. 1986.
9. R. Bellman, "Dynamic Programming", Princeton University Press, Princeton, New Jersey, 1957.
10. H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", IEEE Trans. Acoust., Speech, Singal Processing, vol. ASSP-26, No. 1, pp. 194~200, Feb. 197 8.
11. N. Weste and K. Eshraghian, "Principles of CMOS VLSI Design", Addison-Wesley Publishing Company, 1985.
12. 정광재, 문홍진, 최규훈, 김종교, "음성인식용 DTW PE의 IC화를 위한 MIN회로의 설계", 한국통신학회 논문지, 제13권 제2호, pp. ~ , 1990.



鄭光載(Kwang Jae Jeong) 準會員
1968年4月18日生
1989年2月：全北大學校 電子工學科 卒業
1989年3月～1990年 現在：全北大學校 電子工學科 碩士課程
※主關心分野：디지털 신호처리 및 CAD



文洪眞(Hong Chin Moon) 正會員
1960年7月3日生
1983年2月：全北大學校 電子工學科 卒業
1986年8月：全北大學校 大學院 電子工學科 工學碩士
1990年2月：全北大學校 大學院 電子工學科 博士課程 修了
1988年10月～現在：郡山實業專門大學 電子計算課 全任講師
※主關心分野：디지털信號處理 및 컴퓨터그래픽스



崔圭勳(Kyu Hoon Choi) 正會員
1950年7月11日生
1974年2月：光云工科大学 電子工學課程 卒業
1976年2月：延世大學校 產業大學院 電子工學科 工學碩士
1989年8月：全北大學校 電子工學科 博士課程 修了
1981年3月～現在：全州工業專門大學 電子科 助教授
※主關心分野：디지털信號處理 및 컴퓨팅



金鐘玟 (Chong Kyo Kim) 正會員
1944年8月22日生
1966年2月：全北大學校 電氣工學科 卒業
1977年8月：全北大學校 大學院 電氣工學科 工學碩士
1983年8月：全北大學校 大學院 電氣工學科 工學博士
1979年4月～現在：全北大學校 電子工學科 副教授
1966年7月～1972年5月：遞信部 勤務 (장거리 通信擔當)
1976年3月～1979年4月：全州工業專門大學 電子科
1983年8月～1984年8月：美國 일리노이 工科大学 (IIT) 客員教授
※主關心分野：디지털信號處理, CAD 및 컴퓨터그래픽스