

1-비트 기호치환 가산기의 광학적인 구현

(Optical Implementation for 1-bit Symbolic Substitution Adder)

曹 雄 鎬*, 金 秀 重**

(Woong Ho Cho and Soo Joong Kim)

要 約

MSD 수체계를 사용한 광가산기는 전통적인 2진 가산에서 발생하는 연속적인 올림수 전달을 좌측이동 2번으로 제한하도록 제안되었다. 그러나 MSD 수체계는 3디지트들을 표현하기 위하여 3가지 다른 상태로 부호화해야 한다. 본 논문에서는 기호치환 방법에 의한 1-비트 가산규칙에 근거한 광가산기의 구성을 제안한다. 이 가산기는 디지털컴퓨터에 사용한 2진 입력을 그대로 사용할 수 있고, 또한 MSD가산기보다 시스템의 크기도 더 작게 구현될 수 있음을 보인다.

Abstract

Optical adders using a modified signed-digit(MSD) number system have been proposed to restrict the carry propagation chain encountered in a conventional binary adder to two positions to the left. But, MSD number system must encode three different states to represent the three possible digits of MSD. In this paper, we propose the design of an optical adder based on 1-bit addition rules by using the method of symbolic substitution (SS). We show that this adder can use binary input which is used by a digital computer, as it is, and be implemented by smaller system in size than MSD adder.

1. 서론

디지털 컴퓨터는 일반적으로 2진수를 사용하고, 이를 전통적인 가산규칙으로 연산할때 올림수가 발생한

다. 이 가산규칙을 광컴퓨터에 그대로 이용하면 가감산을 수행할 때 올림수 지연시간이 발생하므로 광의 병렬성을 최대한으로 이용할 수가 없다. 올림수 지연시간은 올림수의 연속적인 좌측이동에 의한 것이므로 올림수 발생을 제한하도록 다른 수체계를 사용하여 구현한 광가산기들에는 유수(residue number)^{1,3}나 MSD(modified signed-digit)^{4,6} 등을 사용한 방법들이 제안되었다.

특히 2진수의 잉여표현(redundant representation)인 MSD표현은 디지털 컴퓨터에서 가감산을

*正會員, 大邱工業專門大學 電子計算科
(Dept. of Com. Sci., Taegu Tech. Coll.)

**正會員, 慶北大學校 電子工學科
(Dept. of Elec. Eng., Kyungpook Nat'l Univ.)
接受日字 : 1993年 11月 19日

수행하는 동안에 올림수의 좌측이동을 2번으로 제한한다.⁹⁾ 즉, MSD로 표현된 두 수의 가산은 3디지트(digit)인 '1', '0', '1'로 구성된 모든 가능한 조합에서의 가산규칙에 의해 비트수에 관계없이 3단계에서 합을 구한다.

MSD표현을 광 컴퓨터에 사용하기 위한 강력한 수단은 Huang에 의해 제안된 기호치환(symbolic substitution : SS)방법이다.¹⁰⁾ 기호치환은 영상안에 있는 특별한 패턴의 모든 위치를 인식하여 그 위치에 다른 패턴을 치환하는 것이다. 부울연산자는 비트들의 결합을 인식하여 1비트를 출력하지만 기호치환은 비트들의 결합과 이러한 비트들의 상대적인 위치를 인식하여 비트들의 결합을 출력한다는 점에서 부울논리와 다르다. 즉 기호치환 논리(symbolic substitution logic)는 디지털 논리를 광학적으로 구현하기 위하여 패턴을 변형시켜 바꾸어 넣는 기술이다. 이런 기호치환 논리는 산술과 논리연산, 신호 및 영상처리, 광병렬 컴퓨팅, prolog와 같은 인공지능언어의 구현, 그리고 광랜덤 액세스 기억장치(optical random access memory: ORAM)와 같은 여러 응용분야에 적용될 수 있다.

기호치환 방법을 사용하여 MSD로 표현된 값들을 연산하는 MSD가산기의 구현방법이 제안되었지만⁸⁾⁹⁾, MSD표현은 2진수의 잉여표현이므로 각 값에 대한 MSD표현은 여러가지가 될 수 있다. 또한 기호치환 규칙을 사용하기 위해서 MSD를 부호화할 때 2진값의 부호화와 달리 서로 다른 3가지 형태로 부호화해야 하는 단점이 있다.

본 연구에서는 기호치환 규칙을 도입하여 전통적인 2진 가산에서 올림수가 연속적으로 발생하는 것을 제한하고, 입력데이터의 비트수에 관계없이 2단계에서 합을 구하는 1-비트 가산방법을 제안한다. 또한 LCTV를 사용하여 입력데이터를 편광부호화해서 기호치환의 기준패턴에 따른 데이터 처리를 용이하게 하고, 가산규칙에 의한 1-비트 기호치환 가산기를 구현할 수 있음을 제시한다.

II. 기호치환의 원리

기호치환 방법은 패턴을 인식 및 치환하는 과정으로 나눌 수 있다. 특히 패턴의 어두운 화소('0') 위치를 찾는 인식과정은 입력패턴의 공간 쉬프트연산, 중첩연산, NOR 문턱치(thresholding value)연산, 그리고 마스크연산으로 구성된다.

이들을 광학적으로 구현하기 위해서 쉬프트와 중첩연산은 간섭계를 사용하고 NOR 연산은 병렬 비선형

광문턱치 NOR 게이트를 사용하여 가능하다. 패턴의 어두운 화소를 기준으로 한 기호치환의 인식과정은 그림 1과 2에 나타내고 치환과정은 그림 3에 나타냈다. 그림 1(a)에서와 같이 2 X 2 셀로 구성된 패턴을 기준패턴이라 하고, 기준패턴의 원점을 좌측하단 화소로 가정한다. 기준패턴의 좌측상단과 우측하단 화소가 어두운 값('0')으로, 우측상단과 좌측하단 화소가 밝은 값('1')으로 구성되어 있다고 가정하면 인식과정은 입력영상에서 기준패턴과 같은 패턴이 발생하는 위치를 찾는 것이다. 그림 1(b)에는 인식방법을 나타내고, 인식을 수행하기 위해서는 입력영상의 두 복사본 I₁과 I₂를 만든다.

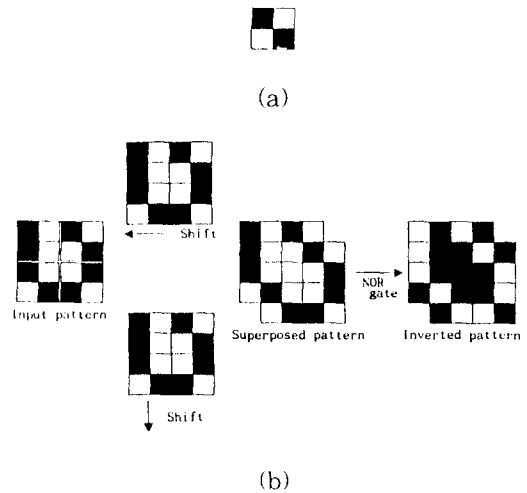


그림 1. 기준패턴의 인식. (a) 기준패턴, (b) 인식 방법

Fig. 1. Recognition of the reference pattern: (a) Reference pattern: (b) method of the recognition.

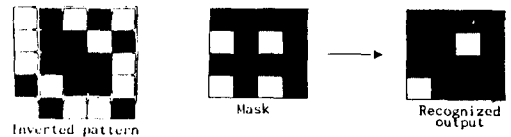


그림 2. 마스크 후의 인식된 영상

Fig. 2. Recognized image after masking.

기준패턴의 좌측상단 화소('0')가 기준패턴의 원점으로 이동하도록 입력영상의 복사본 I₁을 하단으로 1 화소 쉬프트하고, 기준패턴의 다른 '0'인 우측하단 화소가 기준패턴의 원점으로 이동하도록 입력영상의

다른 복사본 I_2 를 좌측으로 1화소 쉬프트한다. 쉬프트된 두 영상은 중첩되어 새로운 영상을 만들고, 이 영상은 NOR 게이트 배열을 통과해서 모두 어두운 화소들로 중첩된 것만 밝게 되고, 나머지 화소들은 모두 어둡게 된 영상으로 된다. 그림 2에는 반전된 영상을 기준패턴의 원점화소들만 통과할 수 있는 마스크를 통과시켜서 최종 인식된 영상이 얻어지는 것을 나타낸다. 최종 인식된 영상의 모든 밝은 화소들은 기준패턴과 같은 패턴이 있는 위치를 나타낸다.

그림 3(a)에는 치환될 패턴을 나타내고, 이의 좌측상단과 우측하단은 밝은 화소('1')로 우측상단과 좌측하단은 어두운 화소('0')로 구성되어 있다. 그림 3(b)에는 치환방법을 나타내고, 치환을 수행하기 위해서는 마스크된 영상의 두 복사본 I_3 와 I_4 를 만든다. 영상 I_3 는 상단으로 1화소 쉬프트하고 영상 I_4 는 우측으로 1화소 쉬프트하여, 각 영상을 중첩시키면 인식된 위치에 치환패턴으로 구성된 새로운 치환영상을 얻을 수 있다.

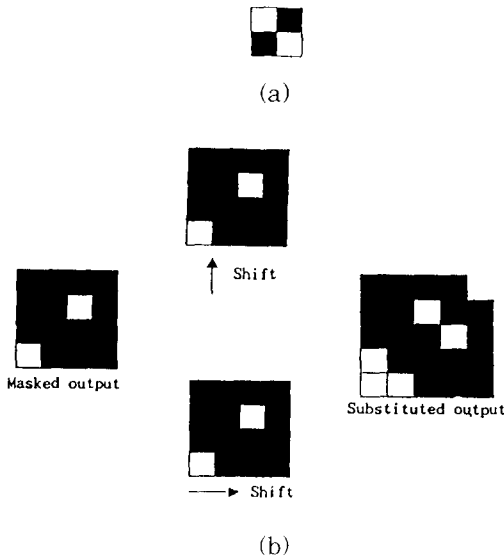


그림 3. 치환된 패턴의 출력, (a) 치환패턴, (b) 치환방법

Fig. 3. Output of the substituted pattern: (a) Substituted pattern: (b) method of the substitution.

그림 1에서 그림 3까지의 인식과 치환과정은 입력 패턴들의 부호화를 세기 부호화 방법을 사용했지만 또 다른 방법은 편광 부호화 방법을 사용하는 것이다. 이는 세기 부호화로 부호화된 영상의 인식 및 치환과정과 수행과정은 같지만 어두운 화소에 대해서는

수평편광빛('↔')으로, 밝은 화소에 대해서는 수직편광빛('↑')으로 나타낸 것이다. 이를 이용하여 그림 1에서의 세기 부호화 패턴을 편광 부호화 패턴으로 바꾸면 그림 4에서와 같다.

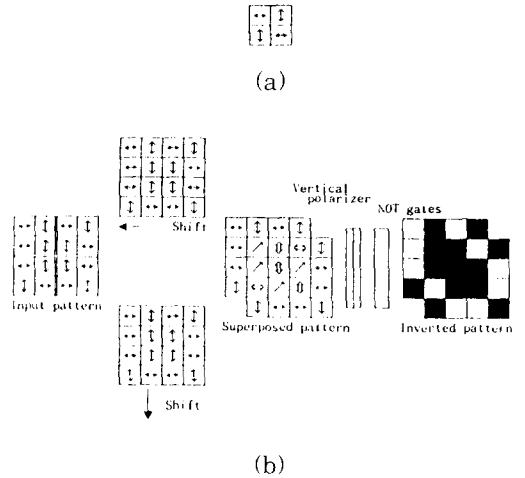


그림 4. 기준패턴의 인식, (a) 기준패턴, (b) 인식 방법

Fig. 4. Recognition of the reference pattern: (a) Reference pattern: (b) method of the recognition.

기준패턴은 그림 4(a), 입력영상에 대한 중첩패턴은 그림 4(b)에서와 같이 나타내며 이 중첩패턴은 두 개의 수평 편광빛을 갖는 화소('↔')들, 두 개의 수직 편광빛을 갖는 화소('↑')들, 그리고 수직 및 수평방향의 편광빛 모두를 갖는 화소('↗')들로 구성된다. 두 개의 수평 편광빛만을 포함하고 있는 화소가 입력 영상에서 기준패턴과 같은 패턴이 있는 위치를 나타낸다. 따라서 이를 제외한 원하지 않는 모든 화소들을 제거하기 위해 다음 단계 수직 편광판을 둔다. 두 수평 편광빛으로 구성된 모든 화소들은 어둡게 되고, 그외 다른 화소들은 밝게 된다. 이 영상이 NOT 게이트 배열을 통과하게 되면 기준패턴이 존재하는 화소들의 위치가 밝게 된 영상으로 된다. 본 연구에서는 기호치환 방법을 광병렬 가산기 구현에 적용할 때 입력의 부호화는 편광 부호화를 이용한다.

그림 1에서 3까지의 과정이 하나의 기호치환 규칙을 수행한 것이고, 하나의 규칙이상을 수행하려면 입력의 여러 복사본을 만들어서 복사본들을 다른 인식 및 치환 구성단위에 적용하면 된다. 광을 이용하여 기호치환 규칙들을 병렬로 수행하면 여러 패턴들의

인식과 치환을 병렬로 동시에 할 수 있으므로 광논리와 광4칙연산들은 SIMD(single instruction multiple data)구조로 구현할 수 있다.

Ⅲ. 1-비트 기호치환 가산기

전통적인 2진 가산에 기호치환 규칙을 적용하려면 두 입력데이터의 각 비트들을 연산하는 반가산기를 사용해야 하고, 이 출력을 편광부호화한 4개 패턴 각각에 대한 치환규칙은 그림 5와 같다.

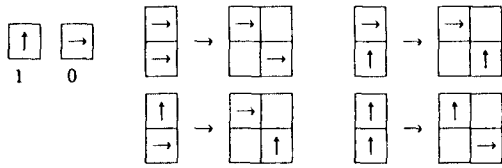


그림 5. 반가산기의 치환규칙
Fig. 5. Substitution rules of half adder.

그림 5의 모든 치환규칙은 병렬로 수행할 수 있지만 결과로 발생한 올림수는 상단의 가산에 영향을 준다. 즉, 가산에서 발생하는 연속적인 올림수 전달은 광의 특성인 병렬성의 최대 이용에 제한적인 요소가 된다.

기호치환 규칙을 사용하여 올림수 전달을 제한하고 연산할 수 있는 제안한 1비트 기호치환 가산기는 1×1 비트들로 구성된 각 셀을 병렬로 동시에 가산하는 방법으로 5비트로 구성된 입력 데이터를 사용하는 이의 블록도는 그림 6에 나타났다.

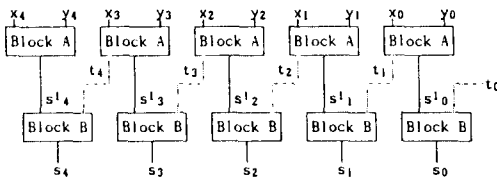


그림 6. 1-비트 기호치환 가산기의 블록도
Fig. 6. Block diagram of 1-bit SS adder.

이의 블록도에서 블록 A에는 두 개의 1비트 값을 동시에 가산해서 중간 합(s'_i)과 상위 올림수(t_i)를 출력한다. 즉, 블록 A에서의 가산에 기호치환 규칙을 도입하면 중간 합은 입력데이터의 4개 기준패턴에 대한 치환패턴을 나타내고, 상위 올림수는 기준패턴이 인식될 때 기준패턴에 따라 미리 결정되는 값으로 병렬

로 동시에 구할 수 있다. 또 블록 B에서는 발생한 상위 올림수의 값을 제어값(t_i)으로 사용하여 중간 합(s'_i)을 변화시켜서 최종 합(s_i)을 출력한다.

그림 6에서의 t_i 값이 '1'이면 중간 합을 인버팅시켜서 최종 출력값으로, '0'이면 그대로 최종 출력값으로 한다. 즉 그림 6에서 블록 B는 좌측 쉬프트된 t_i 값과 s'_i 값을 XOR 논리연산하는 기능이다. 이 블록도와 같은 가산에 기호치환 규칙을 적용하기 위한 가산규칙은 그림 6의 블록 A에 대해서 그림 7에서와 같다. 그림 7(a)에는 1×1셀의 일반적인 가산규칙을 나타내고, 그림 7(b)에는 가산의 각 가능한 기준패턴을 나타냈다. 그림 7(b)에서와 같이 구성된 셀의 가산규칙에서 집단(group) A는 중간 합과 상위 올림수 모두가 '0'인 경우를, 집단 C는 중간 합은 '0'이고 상위 올림수는 '1'인 경우를, 집단 B는 중간 합은 '1'이고 상위 올림수가 '0'인 경우를 나타낸다. 그림 6의 블록 A에서 기준패턴을 인식할 때 상위 올림수는 기준패턴이 집단 A로 인식될 때는 '0'을, 집단 B로 인식될 때는 'X'를, 집단 C로 인식될 때는 '1'을 갖도록 한다. 이 규칙을 그림 6의 블록도에 적용했을 때 그림 7(b)의 집단 A와 C인 경우는 각각 올림수 '0'과 '1'이 발생하지만 하단의 1-비트 기호치환 가산기로부터 발생한 올림수 '1'이 상단에 전달되면 가산결과에는 영향을 주지만 다음 상단의 1-비트 기호치환 가산기의 연산에는 영향을 주지 않는다.

$$\begin{array}{r} x_i \\ y_i \\ \hline t_i \ s'_i \end{array} \quad (a)$$

group A	group B		group C
0	0	1	1
0	1	0	1
0 0	0 1	0 1	1 0
	x	x	

(b)

그림 7. 1-비트 기호치환 가산기 치환규칙의 유도.
(a) 일반적인 경우, (b) 4가지 기준패턴
Fig. 7. Derivation of the substitution rules of the 1-bit SS adder: (a) General case; (b) the 4 reference patterns.

그러나 그림 7(b)의 집단 B인 경우는 올림수 '0'이 발생하지만 하단의 1-비트 기호치환 가산기로부터 올림수 '1'이 상단에 전달되면 가산결과에도 영향을 주고, 올림수 '0'이 '1'로 바뀌게 되어서 다음 상단의 1-비트 기호치환 가산기의 연산에 영향을 준다. 따라서 집단 B의 올림수를 'X'로 표현하였으며 이 값은 하단 올림수와 같은 값을 갖는다. 또한 모든 집단의 경우에 하단으로부터 올림수 '0'이 상단에 전달되면 상단의 올림수가 변하지 않으므로 다음 상단에는 상단의 올림수가 그대로 전달되어서 하단 올림수에 의한 다음 상단의 가산에는 영향을 주지 않는다. 기준패턴을 인식할 때 집단 B의 'X'는 하단의 올림수와 같도록 하여야 하므로 'X'가 연속해서 발생할 경우 이로 인한 지연시간이 발생하고 이는 입력데이터 구성에 따라 의존적이다. 즉, (1,0)나 (0,1)이 연속해서 발생하는 경우 이 올림수를 하단의 올림수와 같도록 하는 데 지연시간이 걸린다. 하지만 이 가산규칙은 기존의 MSD가산규칙에 비해서 기준패턴의 수와 치환단계가 적어서 더 작은 시스템으로 구현할 수 있다.^[11]

1-비트 기호치환 가산기의 가산규칙을 기호치환 규칙에 적용하기 위하여 편광 부호화된 치환규칙은 그림 8에서와 같다. 여기서 패턴의 화소들은 '0'과 '1'로 구성되고 '0'은 수평 편광빛으로, '1'은 수직 편광빛으로 표현하였다.

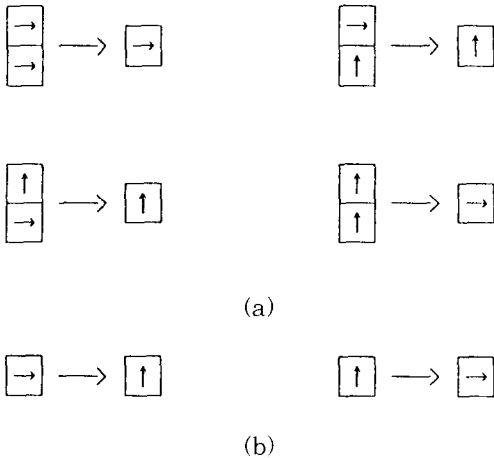


그림 8. 1-비트 기호치환 가산기의 치환규칙. (a) 그림 6의 블록 A에 대한 치환규칙. (b) 그림 6의 블록 B에 대한 치환규칙

Fig. 8. Substitution rules of 1-bit SS adder: (a) Substitution rules for block A in Fig. 6; (b) substitution rules for block B in Fig. 6.

그림 8의 치환규칙에 의한 1-비트 기호치환 가산기의 블록 A 연산을 위한 기호치환 논리에 기본을 둔 구조는 그림 8(a)에서 왼쪽의 패턴을 인식하여 오른쪽 패턴으로 치환하는 것이다. 이때 기준패턴의 인식 과정은 4단계로 나눌 수 있다.

- (1) 왼쪽의 기준패턴들에 대해서 각 패턴에 따라 입력 영상의 복사본 2 개를 만든다.
- (2) 기준패턴의 각 화소에 대한 입력영상의 복사본을 '0'에 대해서는 45°에서 $\lambda/2$ 위상지연판을 통과시키고, '1'은 그대로 둔다.
- (3) 하나의 복사본은 그대로 두고, 다른 복사본은 한 화소 만큼 아래로 쉬프트한다.
- (4) 쉬프트된 복사본을 중첩시킨다.

중첩된 셀 중에서 2개의 수직편광된 빔만을 갖는 셀이 기준패턴의 존재를 가르킨다. 따라서 이를 제외한 다른 결합의 셀들을 제거하기 위하여 중첩된 영상 뒤에 수평 편광판과 인버터 배열을 사용한다. 치환단계는 위의 과정으로 인식된 위치에 그림 8(a)에서의 오른쪽 패턴을 치환하면 된다.

또한 기준패턴을 인식할 때 집단구분에 의해 얻는 올림수는 광쉬프트 레지스터에 저장하고, 이 값을 좌측 쉬프트해서 그림 8(b)의 기호치환 규칙을 수행하는데 제어신호로 사용한다. 그림 8(b)에는 좌측 쉬프트된 올림수 값이 '1'일 때의 기호치환 규칙을 나타낸다. 블록 B에 대한 기호치환 과정은 그림 8(b)의 왼쪽 패턴을 인식하여 그 자리에 오른쪽 패턴으로 치환하면 된다.

위의 인식과정을 광학적으로 구현하기 위한 시스템 구성은 그림 9에서와 같다.

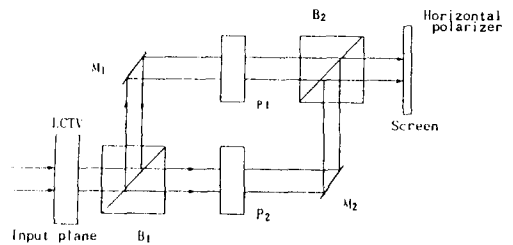


그림 9. 1-비트 기호치환 가산기의 인식과정을 위한 광시스템

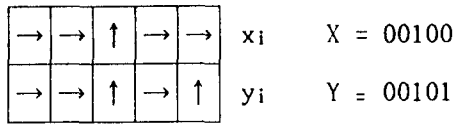
Fig. 9. Optical system for recognition process of 1-bit SS adder.

여기서 P_1 는 $\lambda/2$ 위상지연판을, M_1 는 거울을 나타낸다. 이의 시스템에서 입력면으로는 LCTV를 사용했고, LCTV의 두개의 편광 부호화된 입력값은 마이크로 컴퓨터를 사용하여 만든 패턴을 LCTV의 비디오

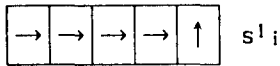
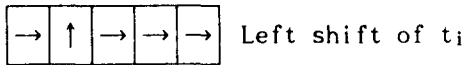
신호 단자를 통해서 입력시켰다.

BS1 (beam splitter)과 BS2는 각각 입력패턴의 복사본 두 개와 두 복사본의 중첩패턴을 만들기 위한 것이고, 거울은 복사본의 화소들을 각 패턴의 원점 화소 위치로 옮기도록 쉬프트시키는 역할을 한다. 또 $\lambda/2$ 위상 지연판은 복사본의 화소들을 기준패턴의 기준 화소에 따라 바꾸는 역할을 하는 것으로 '0'에 대해서는 45° 로 맞추고 '1'에 대해서는 그대로 둔다.

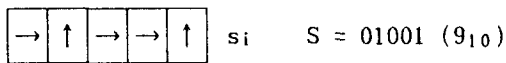
두 데이터 $X=00100(4_{10})$ 와 $Y=00101(5_{10})$ 를 1-비트 기호치환 가산기를 이용하여 연산한 예는 그림 10에서와 같다. 그림 10(a)에는 두 입력데이터를 편광 부호화로 나타낸 것이고, 그림 10(b)에는 그림 8(a)의 기준패턴들에 대한 기호치환 규칙을 그림 10(a)의 입력값에 적용하여 얻은 s_i^1 와 기준패턴들에 따라 정해진 t_i 값을 좌측 쉬프트한 결과를 나타낸다.



(a)



(b)

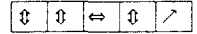
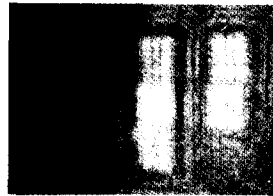


(c)

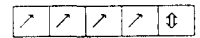
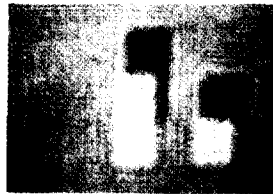
그림 10. 편광부호화된 기호치환 규칙을 사용한 1-비트 기호치환 가산의 예. (a) x_i 와 y_i 를 나타내는 입력데이터. (b) 그림 6에서 블록 A의 출력데이터. (c) 그림 6에서 블록 B의 출력데이터

Fig. 10. Example of the 1-bit SS addition using polarization-coded symbolic substitution: (a) Input data representing x_i and y_i ; (b) output data of block A in fig.6; (c) output data of block B in fig. 6.

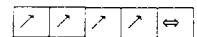
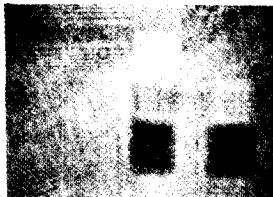
그림 10(c)에는 좌측 쉬프트된 레지스터의 값을 제어 값으로 해서 그림 10(b)의 기호치환 규칙을 수행한 후의 최종 결과 값인 s_i 를 나타내고, 이는 입력값 x_i 와 y_i 의 합을 나타낸다.



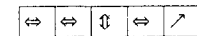
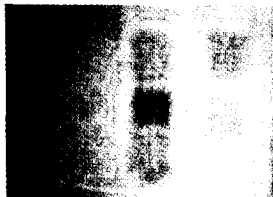
(a)



(b)



(c)



(d)

그림 11. 1-비트 기호치환 가산기의 인식을 실험한 결과. (a)기준패턴 (0, 0). (b)기준패턴(0, 1). (c)기준패턴(1, 0). (d)기준패턴(1, 1)

Fig. 11. Experimental results about recognition of 1-bit SS adder: (a) Reference pattern (0,0); (b)reference pattern (0,1); (c)reference pattern (1,0); (d) reference pattern (1,1).

IV. 실험결과 및 고찰

그림 11은 그림 10에 주어진 두 데이터에 대해서 1-비트 기호치환 가산규칙의 인식과정을 그림 9의 시스템을 사용하여 얻은 실험결과를 나타낸다.

즉 그림 9에서 스크린 면의 수평 편광기 다음에 카메라를 놓고 찍은 영상이다. 그림 9의 시스템에서 입력 데이터는 그림 10(a)의 데이터를 사용했고, 이는 마이크로 컴퓨터를 사용하여 부호화한 후 LCTV에 입력시켰다. 그림 11에서 (a)는 기준패턴이 (0, 0)인 경우를, (b)는 (0, 1)인 경우를, (c)는 (1, 0)인 경우를, 그리고 (d)는 (1, 1)인 경우를 각각 나타낸다. 특히 그림 11(c)의 결과로 입력영상에 기준패턴 (1, 0)이 존재하지 않음을 알 수 있다. 그림 11의 오른쪽은 그림 9의 시스템에서 기준패턴에 따른 스크린 면의 중첩영상을 편광 부호화로 나타낸 것이다. 여기서 '↔'는 수평 편광빛이 중첩된 경우를, '⊥'는 수직 편광빛이 중첩된 경우를, 그리고 '/'는 수직 및 수평 편광 빛이 같이 있는 경우를 나타낸다. 치환과정은 그림 11에서와 같이 인식된 각각의 셀 위치에 그림 8(a)의 기준패턴에 따라 오른쪽의 패턴으로 치환하면 된다.

V. 결론

1-비트 기호치환 가산기는 연산에서 올림수 발생을 2 단계로 제한하였고, 입력데이터의 비트 수에 관계없이 병렬가산을 수행할 수 있게 하였다. 편광 부호화에 의한 1-비트 기호치환 가산기는 전자식 컴퓨터와 같이 2진 표현을 사용하지만 광의 병렬성을 충분히 이용한 광 병렬 시스템으로 구현할 수 있음을 확인하였다. '0'과 '1'의 두 디지털에 대한 부호화는 0° 및 90°로 편광 부호화해서 기준패턴을 인식하는데 편광기를 사용하였다.

1-비트 기호치환 가산기는 MSD 가산기보다 기호치환 수행단계가 더 적고, 각 단계에서 동시에 수행해야 하는 기호치환 규칙수도 더 적어서 더 작은 시스템으로 구현 가능하였다. 또한 1-비트 기호치환 가산기는 인식시스템에서 문턱치 대신에 편광기를 사용할 수 있어서 인식을 쉽게 할 수 있었다.

參 考 文 獻

[1] A. Huang, "The implementation of a residue arithmetic unit via optical and other physical phenomena," in *proceedings of the international optical*

computing conference, IEEE Cat. no. 75, CHO941-5C, 1975.

- [2] A. Huang, Y. Tsunoda, J. W. Goodman, and S. Ischihara, "Optical computation using residue arithmetic," *Appl. Opt.*, vol. 18, pp. 149-162, 1979.
- [3] C. D. Capps, R. A. Falk, and T. L. Houk, "Optical arithmetic and logic unit based on residue arithmetic and symbolic substitution," *Appl. Opt.*, vol. 27, pp. 1682-1686, 1988.
- [4] K-H. Brenner, A. Huang, and N. Streibl, "Digital optical computing with symbolic substitution," *Appl. Opt.*, vol. 25, pp. 3054-3060, 1986.
- [5] P. A. Ramamoorthy and S. Antony, "Optical modified signed digit adder using polarization-coded symbolic substitution," *Opt. Eng.*, vol. 26, pp. 821-825, 1987.
- [6] S. Kawai, and Y. Kohga, "Modified signed-digit optical processors using computer-generated holograms," *Appl. Opt.*, vol. 31, pp. 6193-6199, 1992.
- [7] K-H. Brenner, "New implementation of symbolic substitution logic," *Appl. Opt.*, vol. 25, pp. 3061-3064, 1986.
- [8] K. Hwang and A. Louri, "Optical multiplication and division using modified signed-digit symbolic substitution," *Opt. Eng.*, vol. 28, pp. 364-373, 1989.
- [9] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," *IRE Trans. Electron. Comput. EC-10*, pp. 389-400, 1961.
- [10] A. Huang, "Parallel algorithms for optical digital computers," in *Technical digest, IEEE, Tenth International Optical Computing Conference*, pp. 13-17, 1983.
- [11] 조 옹호, 배 장근, 김 정우, 노 덕수, 김 수중, "부호치환 규칙을 이용한 광2-비트 가산기," 한국 통신학회 논문지, vol. 18, pp. 871-880, 1993

著 者 紹 介

曹 雄 鎬(正會員) 第 29卷 B編 第 12號 參照
현재 대구공업전문대학 전자계산과
조교수

金 秀 重(正會員) 第 29卷 B編 第 12號 參照
현재 경북대학교 전자공학과 교수