論 文

부호치환 규칙을 이용한 광2-비트가산기

正會員 曺 雄 鎬* 正會員 裵 長 根** 正會員 金 正 雨**
正會員 盧 德 樹*** 正會員 金 秀 重**

Optical 2-bit Adder Using the Rule of Symbolic Substitution

Woong Ho Cho,* Jang Keun Bae,** Jeong Woo Kim,** Duck Soo Noh,***
Soo Joong Kim** *Regular Members*

要 約

전통적인 2진 가산규칙은 올림수를 발생시키고 MSB까지 올림수 전달이 발생하므로 직렬가산을 수행한다. 따라서 2진 가산에서 올림수 전달은 광의 병렬성을 최대한으로 이용할 수가 없다. MSD 수체계를 사용한 광가산기는 전통적인 2진 가산에서 발생하는 연속적인 올림수 전달을 제한하도록 제안되었다. 그러나 MSD 수체계는 MSD의 3가지 다지트를 표현하기 위하여 3가지 다른 상태로 부호화해야 한다. 본 논문에서는 SS방법을 사용하여 2-비트 가산규칙에 근거한 광병렬 가산기의 구성을 제안한다.

ABSTRACT

Conventional binary addition rules require a carry formation and propagation to the most significant bit, and lead to serial addition. Thus, the carry propagation in a binary addition stands as a hindrance to the full utilization of parallelism optics offers. Optical adders using a modified signed-digit(MSD) number system have been proposed to elimnate the carry propagation chain encountered in a conventional binary adder. But, MSD number system must encode three different states to represent the three possible digits of MSD. In the paper, we propose the design of a parallel optical adder based on 2-bit addition rules using the method of symbolic substitution(SS).

I. 서 론

디지탈 컴퓨터는 일반적으로 2진수를 사용하고, 이를 전통적인 가산규칙으로 연산할때 올림수가 발생하는 직렬가산을 수행한다. 이 가산규칙을 광컴퓨터에 이용하면 가감산을 수행할때 올림수지연이 발생하므로 광의 병렬성을 최대한으로 이용할 수가 없다. 올림수지연은 올림수의 연속적인 좌측이동에 의한 것이므로 올림수 발생을 제거할 수 있는 제안된 한 방법은 피연산자를 잉여표현(redundant representation)으로 나타내는 것이다.

2진수의 잉여표현인 MSD(modified signed-digit) 표현은 디지탈 컴퓨터에서 가감산을 수행하는 동안에 올림수의 좌측이동을 2번으로 제한한다.[1] 즉, MSD로 표현된 두 수의 가산은 비트수에 관계없이 3

* 大邱工業專門大學 電子計算科
Dept. of Com. Sci., Taegu Tech. Coll.
** 慶北大學校 電子工學科
Dept. of Elec. Eng., Kyungpook Nat'l Univ.
*** 慶北産業大學校 電子工學科
Dept. Elec. Eng., Kyungpook Sanup Univ.
論文番號 : 93-88

단계에서 합을 구할 수 있다. MSD는 3디지트(digit)인 '1', '0', '$\bar{1}$'로 표현되고 이들의 가산은 완전히 병렬로 수행할 수 있다. MSD표현과 관련된 알고리즘의 제안된 구현방법들은 LSI(large scaled integration)기술을 이용한 전기적인 방법과[2] 광의 병렬성을 이용하기 위하여 MSD의 디지트를 3상태 크기부호화[3]나 3상태 편광부호화[4] 같은 방법으로 부호화하여 광학적인 시스템에 적용시킨 광학적인 방법으로 나눌 수 있다. MSD표현을 광병렬 컴퓨터에 사용하기 위한 강력한 수단은 Huang에 의해 제안된 SS(symbolic substitution)방법이다.[5-7] SS방법을 사용하여 MSD로 표현된 값들을 연산하는 MSD가산기의 구현방법이 제안되었으며,[8-9] 이렇게 구현된 MSD가산기는 MSD로 표현된 피연산자의 비트수에 관계없이 3단계에서 합을 구할 수 있음도 제시되었다.

그러나 MSD표현은 2진수의 잉여(redundancy)표현이므로 각 값에 대한 MSD표현은 여러가지가 될 수 있다. 또, SS규칙을 사용하기 위해서 MSD를 부호화할 때 2진 값의 부호화와 달리 서로 다른 3가지 형태로 부호화해야 하는 단점이 있다.

본 연구는 2진데이타를 사용하는 연산방법에 SS규칙을 도입하여 전통적인 2진 가산에서 올림수가 연속적으로 발생하는 것을 제안하고, 입력데이타의 비트수에 관계없이 2단계에서 합을 구하는 2-비트 가산방법을 제안한다. 또, 입력데이타를 편광부호화하고 SS규칙을 사용하여 2-비트가산기를 구현할 수 있음을 제시한다.

Ⅱ. SS의 원리

SS원리는 영상안에 있는 특별한 패턴의 모든 위치를 인식하여 그 위치에 다른 패턴을 치환하는 것이다. SS는 인식과 치환과정으로 나눌 수 있는데, 인식과정은 그림1과 그림2에 나타내고 치환과정은 그림3에 나타내었다.[6] 그림 1(a)와 같이 2*2 셀로 구성된 패턴을 기준패턴이라 하고, 기준패턴의 원점을 좌측하단 화소로 가정한다. 기준패턴의 좌측상단과 우측하단 화소가 어두운값('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 I_1, I_2를 만든다. 기준패턴의 좌측상단 화소('0')가 기준패턴의 원점으로 이동하도록 입력영상의 복사본 I_1을 하단으로 1화소쉬프트 하고, 기준패턴의 다른 '0'인 우측하단 화소가 기준패턴의 원점으로 이동하도록 입력영상의 다른 복사본 I_2를 좌측으로 1화소 쉬프트 한다. 쉬프트된 두 영상은 중첩되어 새로운 영상을 만들고, 이 영상은 NOR 게이트배열을 통과해서 둘다 어두운 화소들만 밝게 되고 나머지 화소들은 모두 어둡게 된 영상으로 된다.

그림 2는 반전된 영상을 기준패턴의 원점화소들만 통과할 수 있는 마스크를 통과시켜서 최종인식된 영상이 얻어지는 것을 나타낸다. 최종인식된 영상의 모든 밝은 화소들은 기준패턴과 같은패턴이 있는 위치를 나타낸다.

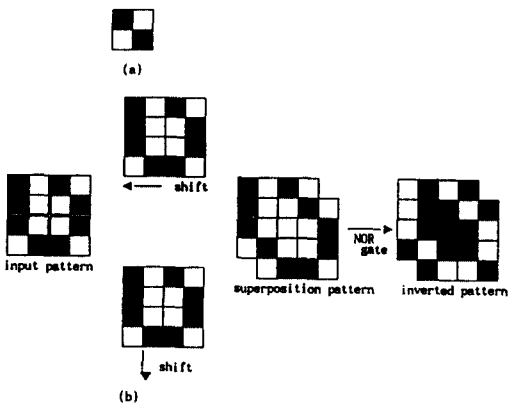


그림 1. 기준패턴의 인식
(a) 기준패턴
(b) 인식방법

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

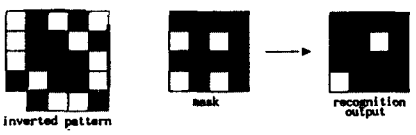


그림 2. 마스크 후의 인식된 영상

Fig. 2. Recognized image after masking.

그림 3(a)는 치환될 패턴을 나타내고, 이의 좌측상단과 우측하단은 밝은 화소('1')로 우측상단과 좌측하단은 어두운 화소('0')로 구성되어 있다. 그림 3(b)

는 치환방법을 나타내고, 치환을 수행하기 위해서는 마스크된 영상의 두 복사본 I_3, I_4를 만든다. 영상I_3는 상단으로 1화소 쉬프트 하고 영상I_4는 우측으로 1화소 쉬프트 하여, 각 영상을 중첩시키면 인식된 위치에 치환패턴으로 구성된 새로운 치환영상을 얻는다.

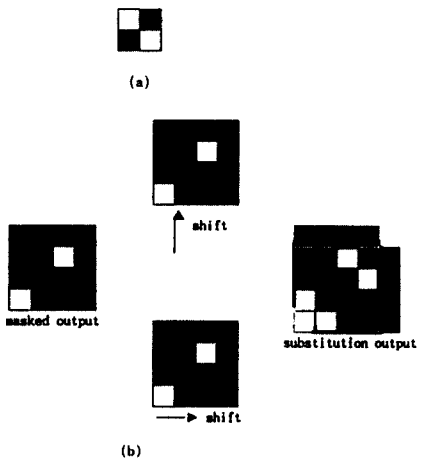


그림 3. 치환된 패턴의 출력
(a) 치환패턴
(b) 치환방법

Fig. 3. Output of the substituted pattern.
(a) substitution pattern,
(b) method of the substition.

그림 1-3까지의 과정이 하나의 SS규칙을 수행한 것이고, 하나의 규칙 이상을 수행하려면 입력의 여러 복사본을 만들어서 복사본들을 다른 인식및 치환 구성단위에 적용하면 된다.

광을 이용하여 SS규칙들을 병렬로 수행하면 여러 패턴들의 인식과 치환을 병렬로 동시에 할 수 있으므로 광논리, 광4칙연산들은 SIMD(Single instruction multiple data)구조로 구현할 수 있다.

Ⅲ. SS를 이용한 MSD가산기

MSD표현은 가감산을 수행할 때 올림수 이동을 좌측 2자리로 해서 밑수(r)를 2로 사용할 수 있는 SD (signed digit)표현이다. 이러한 2자리이동 가산방법은 합을 구하는데 단지 r+1의 값만으로 충분하다. MSD로 표현된 수는 3 디지트('1', '0', '$\bar{1}$')로 구성되고, 임의의 10진수를 n비트 MSD표현으로 나타내면 다음과 같이 된다.[9]

$$X=[1,0,\bar{1}]\,2^{n-1}+\cdots+[1,0,\bar{1}]\,2^{1}+[1,0,\bar{1}]\,2^{0} \quad (1)$$

각 항에 적당한 표현을 위해서는 [1, 0, $\bar{1}$]중에 한 디지트가 선택되어야 하므로 어떤값 X는 여러 MSD 표현 방법을 가질 수 있다. 두 MSD수의 가산은 중간올림수(t_i)와 중간합(w_i)을 구하는 것에 의해 연속적인 3단계로 수행될 수 있다.

$$\begin{aligned}&(1)\ x_i+y_i=rt_i+w_i\\&(2)\ w_i+t_i=rt^1{}_{i+1}+w^1{}_i\\&(3)\ w^1{}_i+t^1{}_i=s_i\end{aligned} \quad (2)$$

SS는 공간적으로 분포된 배열에 적합한 방법이고 SS규칙은 MSD가산방법에 의해 정의된다. 그림 4는 MSD가산방법에 SS규칙을 적용하기 위한 4비트 MSD가산기의 블록도를 나타내고, 이는 식(2)를 수행하는 3단계 블럭으로 구성된다.

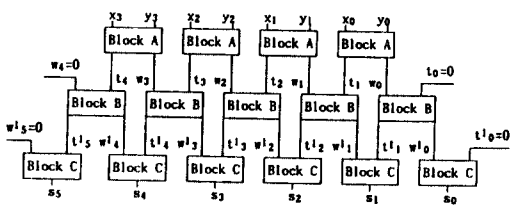


그림 4. SS규칙을 이용하기 위한 MSD가산기의 블럭도
Fig 4. Block diagram of the MSD adder to use SS rules.

그림 5는 그림 4의 블럭도에서 모든 입력디지트에 대한 각 블럭A, B, C에서의 입력과 출력의 관계를 나타낸다.

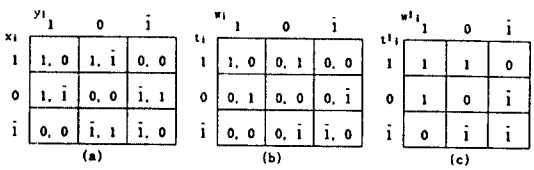


그림 5. 두 디지트에 대한 각 블럭의 입·출력 관계
(a) 블럭 A에 대한 입·출력
(b) 블럭 B에 대한 입·출력
(c) 블럭 C에 대한 입·출력

Fig. 5. Input and output relation of each block for two digits.
(a) input and output for block A,
(b) input and output for block B,
(c) input and output for block C.

이 MSD가산기의 광학적인 구현을 위해서는 MSD의 3가지 디지트의 가능한 값에 대한 표현방법이 필요하다. 이것은 빛의 편광특성을 이용하여 '1'은 수직편광빛으로, '0'은 수평편광빛으로, '$\bar{1}$'는 45°로 편광된 빛으로 표현할 수 있고, 이를 그림 6에 나타냈다.

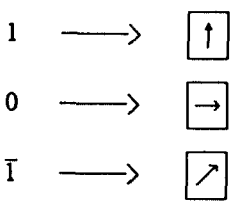


그림 6. 편광에 의한 MSD 디지트의 표현방법
Fig. 6. Representation of an MSD digit by polarized light.

그림 7은 입력과 출력을 편광부호화 했을때 블럭 A, B, C에 대한 입·출력 관계를 나타낸다.
MSD가산을 수행하는 SSL(SS logic)에 기본을 둔 구조는 그림 7의 왼쪽에 있는 패턴을 인식하고, 인식된 패턴은 수행되고 있는 단계에 일치하는 패턴으로 치환한다. 여기서, 패턴의 화소들은 '1', '0', '$\bar{1}$'의 편광표현으로 구성된다. 패턴은 다음의 4단계로 인식할 수 있다.

(1) 왼쪽에 있는 기준패턴들에 대해서 각 패턴에 따라 입력데이타 영상의 복사본 2개를 만든다.
(2) 기준패턴의 각 화소에 대한 입력영상의 복사본은 '0'에 대해서는 45°에서 $\lambda/2$ 위상지연판을 통과시키고, '$\bar{1}$'에 대해서는 45°에서 $\lambda/4$ 위상지연판을 통과시키고 '1'은 그대로 둔다.
(3) 복사본의 화소들은 각 패턴의 원점화소 위치로 옮기도록 쉬프트 한다.
(4) 쉬프트된 복사본들을 중첩시킨다.

중첩된 셀 중에서 두 개의 수직편광된 빔만을 갖는 셀이 기준패턴의 존재를 가르키므로 이를 제외한 다른 결합의 셀들은 제거되어야 한다. 따라서, 기준패턴을 포함하고 있는 셀만이 통과하도록 문턱치(thresholding value)를 사용한다. 위의 인식과정 단계는 그림 8과 같다.

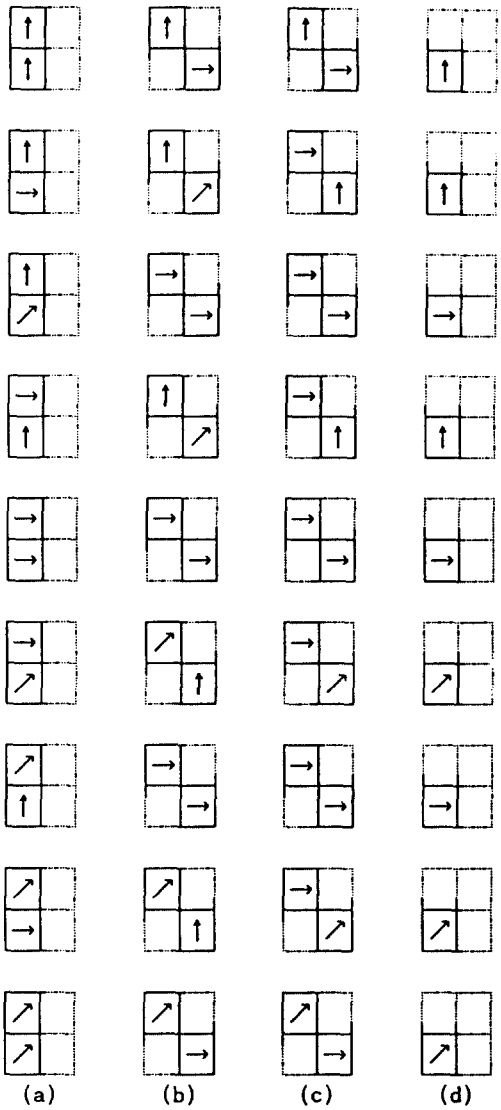


그림 7. 각 블럭에 대한 치환규칙
(a) 각 단계의 입력패턴(xi, yi)
(b) 블럭 A에 대한 치환규칙
(c) 블럭 B에 대한 치환규칙
(d) 블럭 C에 대한 치환규칙
Fig. 7. Substitution rules for the each block.
(a) input pattern(xi, yi) of each stage,
(b) substitution rules for block A,
(c) substitution rules for block B,
(d) substitution rules for block C.

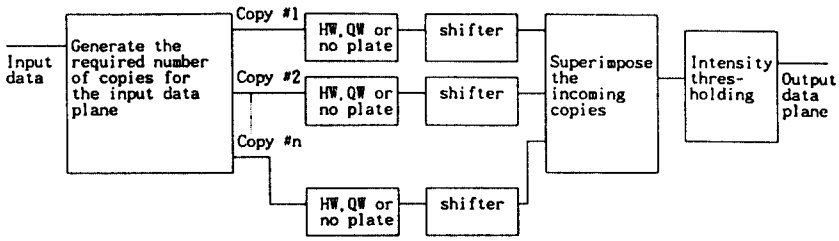
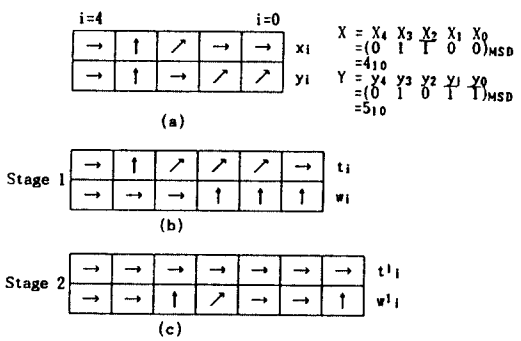


그림 8. 기준패턴에 대한 일반화 된 인식시스템의 블럭도
Fig 8. Block diagram of the generalized recognition systm for reference patterns.

인식된 패턴을 수행되고 있는 블럭의 출력패턴으로 치환하는 규칙은 위의 인식과정과 같은 4단계로 수행될 수 있다. MSD 가산기의 각 단계에서 블럭은 9개의 가능한 기준패턴을 가지므로 9개의 가능한 치환규칙이 필요하다. 각 패턴에서 인식해야 할 화소는 두 화소뿐이므로 각 블럭의 SSL구현은 9개의 치환규칙이 병렬로 수행되고 각 규칙에서 2개의 복사본이 만들어진다.

그림9는 두 데이타 X=01$\bar{1}$00, Y=010$\bar{1}\bar{1}$에 대한 SS를 사용한 MSD가산기의 연산 방법을 나타낸다.

그림 9(a)는 MSD로 표현된 두 데이타를 편광부호화로 나타낸 것이다. 블럭 A의 규칙을 적용하여 그림 9(b)에 주어진 t_i와 w_i의 값을 얻고, 이 값을 기준패턴으로 하여 블럭 B에 정의된 규칙을 적용하여 그림 9(c)에 주어진 $t^1{}_i$와 $w^1{}_i$의 값을 얻는다. 이의 중간 출력값에 블럭 C의 규칙을 적용하여 그림 9(d)에 주어진 최종 출력값인 001$\bar{1}$001을 얻는데 이 값은 사용된 두 x_i, y_i값의 합을 나타낸다.



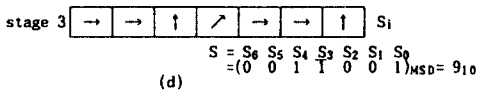


그림 9. 편광부호화된 SS를 사용한 MSD가산의 예
(a) xi와 yi를 나타내는 입력 데이타
(b) 단계 1의 출력 데이타
(c) 단계 2의 출력 데이타
(d) 단계 3의 출력 데이타(최종출력)

Fig. 9. Example of MSD addition using polarization-coded symbolic substitution.
(a) input data representing xi and yi,
(b) output data corresponding to stage 1,
(c) output data corresponding to stage 2,
(d) output data corresponding to stage 3(final output).

Ⅳ. 제안한 2-비트 가산기

전통적인 2진 가산에 SS규칙을 적용하려면 두 입력데이타의 각 비트들을 연산하는 반가산기를 사용해야 하고, 이 출력을 편광부호화한 4개 패턴 각각에 대한 치환규칙은 그림 10과 같다.

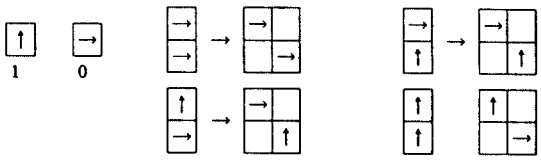


그림 10. 반가산기의 치환규칙
Fig. 10. Substitution rules of half adder.

그림 10의 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.

SS규칙을 사용하여 올림수 전달을 제한하고 연산할 수 있는 제안한 2진 2-비트가산기는 2*2 비트들로 구성된 각 셀을 병렬로 동시에 가산하는 방법으로 4비트 데이타에 대한 이의 블럭도는 그림 11과 같고, 이를 SS규칙에 적용하기 위한 가산결과의 생성과정을 블럭A에 대해서는 그림 12, 블럭B에 대해서는 그림 13과 같다. 그림 11의 시스템 블럭도는 두 2비트 값을 입력하여 동시에 가산해서 두 비트의 합과 상위 올림수를 출력한다. 그림 12(a)는 2*2 셀의 일반적인 가산규칙을 나타내고, 그림 12(b)는 각 가능한 경우를 나타낸다.

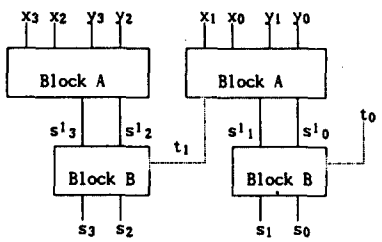


그림 11. 2-비트가산기의 블럭도
Fig. 11. Block diagram of 2-bit adder.

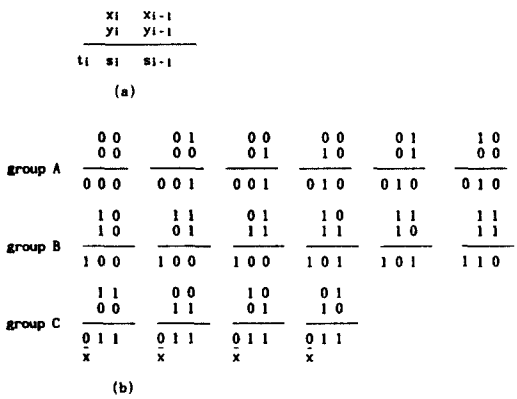


그림 12. 2-비트가산기 치환규칙의 유도
(a) 일반적인 경우
(b) 16가지 가능한 경우

Fig. 12. Derivation of the substitution rules of the 2-bit adder.
(a) general case,
(b) the 16 possible cases.

그림 12(b)에서와 같이 구성된 셀의 가산규칙에서 집단A는 올림수 '0'만이 진단B는 올림수 '1'만이 상단에 전달된다. 또 집단A와 B는 하단의 2-비트가산기로부터 발생한 올림수 '1'이 상단에 전달되면 다음 상단의 2-비트가산기의 연산결과에는 영향을 주지않는다. 집단C는 올림수 '0'이 발생하지만 하단으로부터 올림수 '1'이 상단에 전달되면 다음상단에 '1'의 올림수가 전달되므로 가산결과에 영향을 주고, 하단으로부터 올림수 '0'이 상단에 전달되면 다음상단에 올림수 '0'이 전달되어서 다음상단의 가산결과에는 영향을 주지 않는다. 즉, 집단A와 B의 셀구조는 상단에 영향을 주는 올림수가 '0'이나 '1'로 고정되지만, 집단 C의 셀구조는 하단의 셀구조에 영향을 받으므로 하단에서 발생한 올림수가 '1'이면 상단의 올림수가 '1'이 되고 하단에서 발생한 올림수가 '0'이면 상단의 올림수가 '0'이 된다. 그래서, 집단C의 올림수는 X로 표현한다. 그러나 그림 11의 블럭도는 집단C 때문에 올림수 X가 연속해서 발생하는 경우 이들을 하단의 올림수와 같도록 하는데 필요한 지연시간을 갖는다. 그림 12의 가산규칙에서 하단에서 발생한 올림수가 '1'이면 합이 바뀌게 되므로 위의 SS적용을 위한 가산규칙은 다시 한번 더 연산을 수행해야 하고, 이때 적용되는 SS규칙은 그림 13에 나타냈다. 즉, 그림 13은 s^1_i s^1_{i-1}의 값에 '1'을 증가시킨 것을 나타낸다.

s^1_i	s^1_{i-1}	->	s_i	s_{i-1}
0	0	->	0	1
0	1	->	1	0
1	0	->	1	1
1	1	->	0	0

그림 13. 증가치 치환규칙의 유도
Fig. 13. Derivation of the substitution rules for increment.

그림 12의 SS규칙은 16가지가 되고, 이의 셀값들에 AND, XOR 논리연산을 전처리하여 얻은 결과를 각각 입력값으로 해서 SS규칙을 적용하면 SS규칙은 9가지로 줄어든다. 따라서 병렬로 수행해야 하는 SS규칙이 43.75% 줄어든 시스템을 구현할 수 있다. AND와 XOR 전처리를 한 후의 SS규칙을 적용하기 위한 가산방법은 그림 14에 나타냈다. 여기서, C_i는

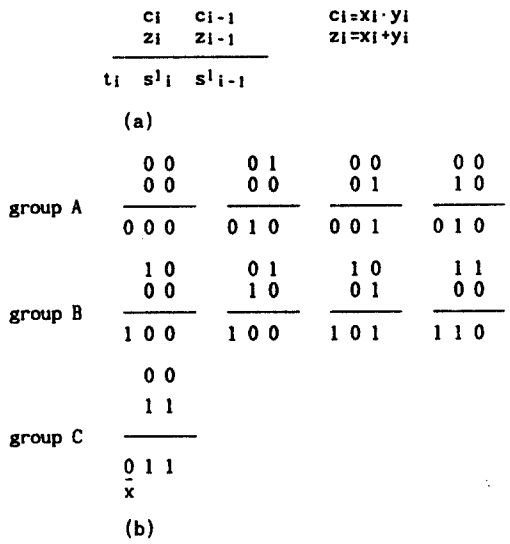


그림 14. 전처리된 2-비트가산기의 치환규칙의 유도
(a) 일반적인 경우
(b) 9가지 가능한 경우

Fig. 14. Derivation of the substitution rules of the preprocessed 2-bit adder.
(a) general case,
(b) the 9 possible cases.

AND연산한 값이고, Z_i는 XOR연산한 값이다.

표 1은 2−1비트 가산기에 대한 SS규칙의 가능한 모든 경우를 나타낸 것이다.

표1의 가산규칙을 광의 병렬성을 이용하기 위한 SS규칙으로 바꾸기 위해서 각 셀에 대한 s_i와 s_{i-1}는 그 셀에 대한 치환값으로 한다. 그리고 t_i는 광쉬프트 레지스터에 저장하고, 이 레지스터값을 좌측쉬프트 해서 그림 12의 SS규칙을 수행하는데 제어신호로 사용한다. 이때 t_i의 값은 표 1의 SS규칙에서 집단 A, B, C에 따라 결정되고, '0'인 경우는 그림 13의 단계를 수행하지 않고, '1'인 경우는 그림 13의 단계를 수

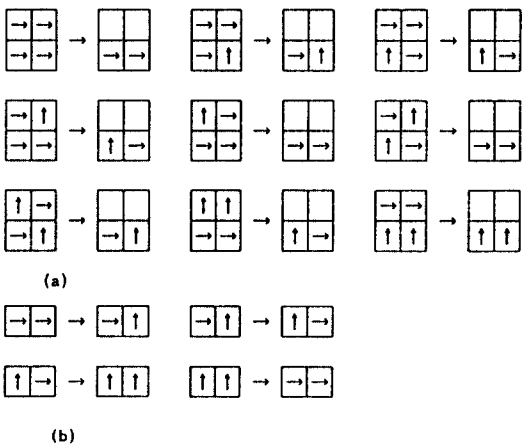


그림 15. 전처리된 입출력에 대한 치환규칙
(a) 그림 14의 치환규칙
(b) 그림 13의 치환규칙

Fig. 15. Substitution rules for the preprocessed input and output.
(a) substitution rules of Fig. 14.
(b) substitution rules of Fig. 13.

표 1. 2-비트가산기의 치환규칙

Talble 1. Substitution rules of 2-bit adder.

group	input digit				preprocessed input digit						
	x_i	x_{i-1}	y_i	y_{i-1}	c_i	c_{i-1}	z_i	z_{i-1}	t_i	s_i	s_{i-1}
A	0	0	0	0	0	0	0	0	0	0	0
	0	1	0	0	0	0	0	1	0	0	1
	0	0	0	1							
	0	0	1	0	0	0	1	0	0	1	0
	1	0	0	0							
	0	1	0	1	0	1	0	0	0	1	0
B	1	0	1	0	1	0	0	0	1	0	0
	1	1	0	1	0	1	1	0	1	0	0
	0	1	1	1							
	1	0	1	1	1	0	0	1	1	0	1
	1	1	1	0							
	1	1	1	1	1	1	0	0	1	1	0
C	1	1	0	0	0	0	1	1	x	1	1
	0	0	1	1							
	1	0	0	1							
	0	1	1	0							

행한 후 최종출력값을 얻는다. 그래서 제안한 2-비트 가산기는 연속적인 2단계 SS규칙의 수행으로 연산을 끝낸다. 전처리(XOR과 AND연산) 과정과 쉬프트 연산은 광ALU(arithmetic and logical unit)에 있는 논리 연산기와 쉬프트 연산기에서 수행될 수 있고, 여기서는 이 결과를 이용하는 것으로 가정했다. 그림 15는 전처리된 각 셀들의 값을 편광부호화했을때 그림 13과 14의 SS규칙을 나타낸다. 여기서, 패턴의 화소들은 '0', '1'로 구성되고 '1'은 수직편광빛으로 '0'은 수평편광빛으로 표현했다.

전처리된 데이타를 사용하여 2-비트 가산기의 연산을 위한 SSL에 기본을 둔 구조는 그림 15(a)의 왼쪽에 있는 패턴을 인식하고, 이를 오른쪽의 패턴으로 치환하는 것이다. 패턴의 인식과정은 5단계로 구분된다.

(1) 왼쪽에 있는 기준패턴들에 대해서 각 패턴에 따라 입력데이타 영상의 복사본 4개를 만든다.
(2) 기준패턴의 각 화소에 대한 입력영상의 복사본들은 '0'에 대해서는 45°에서 $\lambda/2$위상지연판을 통과시키고, '1'은 그대로 둔다.
(3) 복사본의 화소들은 각 패턴의 원점화소 위치로 옮기도록 쉬프트 한다.
(4) 쉬프트된 복사본들은 중첩시킨다.
(5) 중첩된 영상뒤에 마스크를 두어 기준영상의 원점화소들만 통과하도록 한다.

마스크된 셀 중에서 4개의 수직편광된 빔만을 갖는 셀이 기준패턴의 존재를 가르킨다. 따라서 이를 제외한 다른 결합의 셀들을 제거하기 위하여 마스크된 영상뒤에 수평 편광판과 인버터(invert)배열을 사용한다. 인식된 패턴을 치환패턴으로 치환하는 규칙은 위의 인식단계 중 (1)-(4)까지이고, (1)에서 복사본은 2개만 만들면 된다.

그림 16은 두 데이타 X=00100110(38_{10})와 Y=0011 0011(51_{10})를 AND와 XOR 논리연산을 한 후 2-비트 가산기에 입력하여 연산하는 예를 나타낸다.
그림 16(a)는 두 입력데이타 값을 AND와 XOR 논리연산을 한 후 이를 편광부호화로 나타낸 것이고, 그림 16(b)는 그림 15(a)의 기준패턴들에 대한 SS규칙을 그림 16(a)의 입력값에 적용하여 얻은 s^1_i와 기준패턴들에 따라 정해진 t_i의 값을 좌측 쉬프트한 결과를 나타낸다. 그림 16(c)는 좌측쉬프트된 레지스터의 값을 제어값으로 해서 그림 15(b)의 SS규칙을 수행한 후의 최종결과 값을 나타내고, 이는 두 입력값 x_i, y_i의 합을 나타낸다.

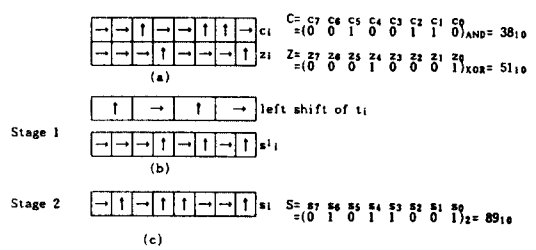


그림 16. 편광부호화된 SS를 사용한 전처리 된 2-비트가산의 예
(a) ci, zi를 나타내는 입력 데이타
(b) 단계1의 출력 데이타
(c) 단계2의 출력 데이타

Fig. 16. Example of the preprocessed 2-bit addition using polarization-coded symbolic substitution.
(a) input data representing ci and zi,
(b) output data corresponding to stage 1,
(c) output data corresponding to stage 2.

V. 결 론

제안한 2-비트가산기는 연산에서 올림수 발생을 2단계로 제한하므로 피연산자의 비트수에 관계없이 완전히 병렬로 수행한다. 또, 이는 입력으로 이진수를 사용하므로 전자식 컴퓨터의 수표현을 그대로 사용할 수 있고, MSD보다 부호화가 용이하다.

전처리된 2-비트가산기는 각 블럭에서 동시에 처리해야 하는 SS규칙이 적어서 2-비트가산기보다 더 작은 시스템으로 구현 가능하다. 편광부호화 SS규칙을 이용한 2-비트 가산기는 전자식컴퓨터와 같이 2진 표현을 사용하지만 광의 병렬성을 충분히 이용할 수 있다.

參 考 文 獻

1. A. Avizienis, "Signed-digit number represntations for fast parallel arithmetic," IRE Trans. Electron. Comput. Ec-10, 389-400(1961).
2. N. Takagi, H. Yasuura, and S. Yajima, "High-Speed VLSI multiplication algorithm with a redundant binary addition tree," IEEE Trans. Comput. COM-34, 789(1985).
3. B. L. Deake, R. P. Bocker, M. E. Lasher, R. H. Patterson, and W. J. Miceli, "Photonic co-

mputing using the modified signed-digit number representation," Opt. Eng. 25, 38(1986).
4. M. E. Lasher, T. B. Henderson, B. L. Drake, and R. P. Bocker, "Encoding Schemes for a digital optical multiplier using the modified signed-digit numer representation," Proc. Soc. photo-Opt. Instrum. Eng. 639(1986).
5. A. Huang, "Parallel algorithms for optical digital computers," in Technical digest, IEEE. Tenth International Optical Computing Conference, 13-17(1983).
6. K-H. Brenner, A. Huang, and N. Streibl, "Digital optical computing with symbolic substitution," Appl. Opt. 25, 3054-3060(1986).
7. K-H Brenner, "New implementation of symbolic substitution logic," Appl. Opt. 25, 3061-3064(1986).
8. P. A. Ramamoorthy and S. Antony, "Optical modified signed digit adder using polarization-coded symbolic substitution," Opt. Eng. 26(8),

曺 雄 鎬(Woong Ho Cho) 正會員
1959年 10月 22日生
1982年 2月 : 경북대학교 전자공학과 졸업(공학사)
1984年 2月 : 영남대학교 대학원 전자공학과 졸업(공학석사)
1988年~현재 : 경북대학교 대학원 박사과정 수료
1985年~현재 : 대구공업전문대학 전산과 조교수
※주관심분야 : 광컴퓨팅, 광신호처리 및 광패턴인식등임

裵 長 根(Jang Keun Bae) 正會員
1965年 12月 1日生
1987年 2月 : 경북대학교 전자공학과 졸업(공학사)
1991年 2月 : 경북대학교 대학원 전자공학과 졸업(공학석사)
1991年 3月~현재 : 경북대학교 대학원 전자공학과 박사과정 재학중
※주관심분야 : 광컴퓨팅, 광패턴인식 및 광신호처리등임.

金 正 雨(Cheong Woo Kim) 正會員
1964年 5月 10日生
1987年 2月 : 경북대학교 전자공학과 졸업(공학사)
1989年 2月 : 경북대학교 대학원 전자공학과 졸업(공학석사)
1989年 3月~現在 : 경북대학교 대학원 전자공학과 박사과정 재학중
※주관심분야 : 광신호처리, 광형태인식 및 광컴퓨팅 등임.

盧 德 樹(Duck Soo Noh) 正會員
1954年 1月 14日生
1977年 2月 : 경북대학교 전자공학과 졸업(공학사)
1983年 2月 : 경북대학교 대학원 전자공학과 졸업(공학석사)
1987年 3月~현재 : 경북대학교 대학원 전자공학과 박사과정 수료
1983年 3月~현재 : 경북산업대학교 전자공학과 부교수
※주관심분야 : 광컴퓨팅, 광패턴인식 및 광신호처리등임.

金 秀 重(Soo Joong Kim) 正會員

1941年 6月 25日生

1962年 12月 : 인하대학교 전기공학과 졸업

1966年 2月 : 인하대학교 대학원 공학석사 학위취득

1979年 2月 : 인하대학교 대학원 자공학과 공학박사 학위취득

1966年 3月~1970年 12月 : 삼척공업전문대학 조교수

1976年 9月~1977年 1月 : 미국 SUNY at Bufflo교환 조교수

1980年 8月~1981年 8月 : 미국 University of Texas at Austin 연구교수

1970年 12月~현재 : 경북대학교 전자공학과 교수

1992年 3月~현재 : 경북대학교 공과대학 전자기술연구소 소장

※주관심분야 : 광신호처리, 광컴퓨팅 및 광패턴인식등임.