論文98-35D-10-14

Booth 알고리즘의 승수 비트-쌍 재코딩을 이용한 광곱셈기의 구현에 관한 연구

(A study on implementation of optical high-speed multiplier using multiplier bit-pair recoding derived from Booth algorithm)

曺 雄 鎬 *, 金 鍾 允 **, 盧 德 樹 ***, 金 秀 重 **

(Woong Ho Cho, Jong Yun Kim, Duck Soo Noh, and Soo Joong Kim)

요 약

피승수와 승수의 부호에 상관없이 빠른 이진곱셈을 수행할 수 있는 효과적인 방법으로서 Booth 알고리즘의 승수 비트-쌍 재코딩 알고리즘을 사용한다. 본 연구에서는 승수 비트-쌍 재코딩 알고리즘을 광특성에 적합하도록 변형 발전시킨 광곱셈 알고리즘과 기호치환 가산기로 구성된 고속의 광곱셈기의 구현을 제안한다. 특히, 기호치환 가산규칙을 듀얼-레일 논리로 부호화해서 이 논리의 보수가 언제나 존재하기때문에 기호치환 가산기에서 이 논리의 보수가 시프트연산에 의해 쉽게 구할 수 있게 했다. 또한 시프트된 두 영상을 직렬 연결하여 중첩시키므로서 중첩영상을 얻고, 이 중첩영상을 마스크로 보내 기준영상을 인식하는 기호치환 시스템을 구성한다. 따라서 광곱셈기의 수동광소자의 수와 시스템의 크기를 줄여서 일반적인 광시스템과 비교하여 작은 시스템으로 구현한다.

Abstract

A multiplier bit-pair recoding technique derived from Booth algorithm is used as an effective method that can carry out a fast binary multiplication regardless of a sign of both multiplicand and multiplier. In this paper, we propose an implementation of an optical high-speed multiplier which consists of a symbolic substitution adder and an optical multiplication algorithm, which transforms and enhances the multiplier bit-pair recoding algorithm to be fit for optical characteristics. Specially, a symbolic substitution addition rules are coded with a dual-rail logic, and so the complement of the logic of the symbolic substitution adder is easily obtained with a shift operation because it is always present. We also construct the symbolic substitution system which makes superposition image by superimposing two shifted images in a serial connection and recognizes a reference image by feeding this superimposed image to a mask. Thus, the optical multiplier, which is compared with a typical system, is implemented to the smaller system by reducing the number of optical passive elements and the size of this system.

* 正會員, 大邱工業大學 電子計算科
(Dept. of Computer Science, Taegu technical college)
** 正會員, 慶北大學校 電子電氣工學部
(School of Electronics & electrical Eng., Kyungbook national university)
*** 正會員, 慶一大學校 電子情報工學科
(Dept. of Electronic & Information Eng., Kyungil university)

※ 이 논문은 1997년 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음.
接受日字:1998年6月2日, 수정완료일:1998年9月8日

Ⅰ. 서 론

많은 신호처리 응용에 사용되는 디지털 필터링, 퓨리에변환 등과 같은 것은 주로 곱셈 및 덧셈 등의 연산들로 이루어지므로 이러한 신호처리가 실시간에 처리되기 위해서는 고속의 연산장치가 필요하다. 특히 곱셈은 많은 신호처리 응용에서 중요한 역할을 하므로 VLSI 구현에 적합한 여러 가지 고속의 곱셈알고리즘들이 제안되어 왔으며 전자식 응용분야에 이용되어 왔다[1-3]. 하지만 최근에 광기술의 급속한 발달로 많은 연구자들이 이러한 고속의 알고리즘들을 광의 빠른 처리속도, 높은 병렬성, 비간섭 상호연결성 등의 특징과 결합할 수 있도록 하는 광연산장치 연구에 많은 노력을 기울여 왔다.

디지털광곱셈기를 구현하기 위하여 제안된 한 기술은 광상승적(optical convolution)장치를 사용하여 곱셈과 덧셈을 수행하는 DMAC(digital multiplication by analog convolution)으로 알려진 알고리즘을 사용하는 것이다[4-5]. DMAC알고리즘을 사용하여 이진수 X와 Y를 곱하는 광장치는 AO(acousto-optic) 셀의 곱셈영역을 X와 Y가 반대방향으로 통과하도록 하여 발생한 회절된 빛이 렌즈를 통해 검출기에 집중되도록 한다. 이 검출기의 출력이 두 입력신호의 상승적을 나타낸다. 이러한 DMAC알고리즘의 단점은 고속의 A/D변환기가 필요하다는 것과 두 n비트 수의 곱셈에 대해 처리시간이 2n-1까지 증가된다는 것이다. 다른 방법으로는 이진수, MSD(modified signed digit), RBN(redundant binary number)와 같은 수체계에 기존의 빠른 곱셈알고리즘을 변형 발전시켜 광곱셈기를 구현하는 것이다[6-9]. 이러한 곱셈알고리즘 중에는 곱셈을 하는데 널리 사용되는 방법으로 음수를 2의 보수로 표현하여 피승수와 승수의 부호에 관계없이 곱셈을 수행할 수 있는 A. D. Booth에 의해 제안된 Booth알고리즘이 있다[10]. 하지만 이 곱셈알고리즘의 평균적인 속도는 전형적인 곱셈알고리즘의 속도와 같게 된다. 그래서 이 Booth알고리즘에서 부분적을 선택하는데 승수의 비트-쌍(bit-pair)을 사용하여 승수의 부호에 상관없는 연산을 보장하고 승수가 n비트로 구성될 때 최대한 n/2개의 부분합만을 생성하게 해서 곱셈의 속도를 훨씬 더 증가시킨 기법을 유도할 수 있다[11-12]. 이 기법을 승수 비트-쌍 재코딩(multiplier bit-pair recoding) 방법이라고 한다.

따라서 본 연구에서는 광곱셈 알고리즘으로 Booth 알고리즘의 변형인 승수 비트-쌍 재코딩을 채택하여 광특성에 적합하도록 발전시켜서 제안하고, 이 알고리즘에서의 부분합들은 광의 빠른 처리속도 및 병렬성을 충분히 이용할 수 있는 기호치환(symbolic substitution)가산기로 병렬 덧셈되도록 한 광곱셈기의 구성을 제안한다. 또한 기호치환 가산기에서 덧셈규칙의 부호화에 듀얼-레일(dual-rail)논리를 적용하고 입력면을 직렬로 연결하여 광시스템의 크기를 작게 구현한다.

Ⅱ. 기호치환 가산기의 구성

기호치환(SS : symbolic substitution)규칙을 사용하여 올림수 전달을 제한하고 연산할 수 있는 기호치환 가산기는 1×1 비트들로 구성된 각 셀을 병렬로 동시에 덧셈하는 방법으로 5비트로 구성된 입력데이터를 사용하는 이의 블럭도는 그림 1에 나타냈다[13].

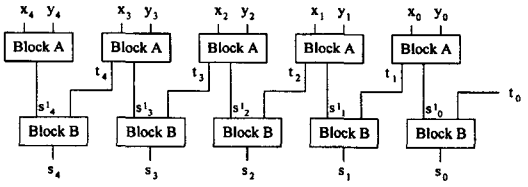


그림 1. 기호치환 가산기의 블럭도
Fig. 1. Block diagram of SS adder.

이의 블럭도에서 블럭 A에는 두 개의 1비트 값을 동시에 덧셈해서 중간 합(s^1_i)과 상위 올림수(t_i)를 출력한다. 즉, 블럭 A에서의 덧셈에 기호치환 규칙을 도입하면 중간 합은 입력데이타의 4개 기준패턴에 대한 치환패턴을 나타내고, 상위 올림수는 기준패턴이 인식될 때 기준패턴에 따라 미리 결정되는 값으로 병렬로 동시에 구할 수 있다. 또 블럭 B에서는 발생한 상위 올림수의 값을 제어값(t_i)으로 사용하여 중간 합(s^1_i)을 변화시켜서 최종 합(s_i)을 출력한다. 그림 1에서의 t_i값이 '1'이면 중간 합을 인버팅시켜서 최종 출력값으로, '0'이면 그대로 최종 출력값으로 한다.

이 블럭도와 같은 덧셈에 기호치환 규칙을 적용하기 위한 덧셈규칙은 그림 1의 블럭 A에 대해서 그림 2에서와 같다. 그림 2(a)에는 1×1셀의 일반적인 덧셈규칙을 나타내고, 그림 2(b)에는 덧셈의 각 가능한 4가지 기준패턴을 나타냈다. 그림 2(b)에서와 같이 구성

된 셀의 덧셈규칙에서 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 1의 블럭 A에서 기준패턴을 인식할 때 상위 올림수는 기준패턴이 집단 A로 인식될때는 '0'을, 집단 B로 인식될때는 'X'를, 집단 C로 인식될때는 '1'을 갖도록 한다. 이 규칙을 그림 1의 블럭도에 적용했을때 그림 2(b)의 집단 A와 C인 경우는 각각 올림수 '0'과 '1'이 발생하지만 하단의 블럭으로부터 발생한 올림수 '1'이 상단에 전달되면 덧셈결과에는 영향을 주지만 다음 상단의 연산에는 영향을 주지 않는다. 그러나 그림 2(b)의 집단 B인 경우는 올림수 '0'이 발생하지만 하단의 블럭으로부터 올림수 '1'이 상단에 전달되면 덧셈결과에도 영향을 주고, 올림수 '0'이 '1'로 바뀌게 되어서 다음 상단의 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.

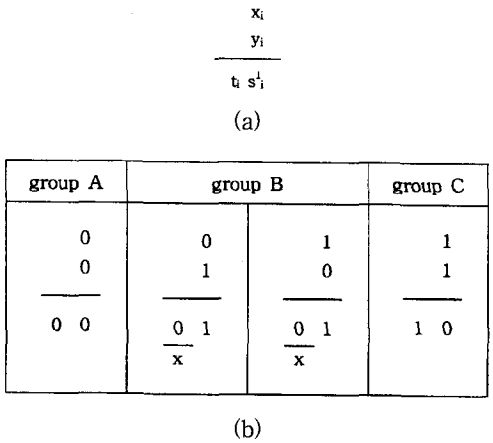


그림 2. 기호치환 가산기 치환규칙의 유도
(a) 일반적인 경우, (b) 4가지 가능한 경우
Fig. 2. Derivation of the substitution rules of the SS adder.
(a) General case, (b) the 4 possible cases

Ⅲ. Booth 알고리즘의 승수 비트-쌍 재코딩

곱셈을 하는데 널리 사용되는 방법으로 음수를 2의 보수로 표현하여 피승수와 승수의 부호에 관계없이 연산할 수 있는 Booth 알고리즘이 있다. Booth 기법으로부터 적어도 Booth 알고리즘의 곱셈 속도보다 2배 이상 빠른 승수 비트-쌍(multiplier bit-pair) 규칙을 유도할 수 있다[10]. Booth 기법에서 승수비트(g_i)는 오른쪽 비트(g_{i-1})의 함수로서 피가수(summand)를 선택한다. 선택된 피가수는 피가수가 덧셈되기 직전의 곱의 LSB(least significant bit) 위치로부터 i이진위치만큼 좌측 시프트된다. 곱셈속도를 증가시킨 기법의 기본 개념은 하나의 피가수를 선택하는데 비트 i-1의 함수로서 비트-쌍 i+1과 i를 선택하여 피가수위치 i(SPi)에서 더해지도록 하는 것이다. 이 피가수는 승수로부터 쉽게 구해질 수 있어야 하고 n/2개의 피가수는 비트-쌍인 (x_1, x_0), (x_3, x_2), (x_5, x_4) 등에 의해 구해진다. Booth 알고리듬에 의해 승수를 Booth 재코딩을 하고 이로부터 비트-쌍 재코딩을 하는 예는 표 1에 나타냈다. 이때 승수의 LSB 우측비트는 항상 '0'인 것으로 한다.

표 1. Booth 재코딩에서 유도된 비트-쌍 재코딩의 예
Table 1. Example of bit-pair recoding derived from Booth recoding.

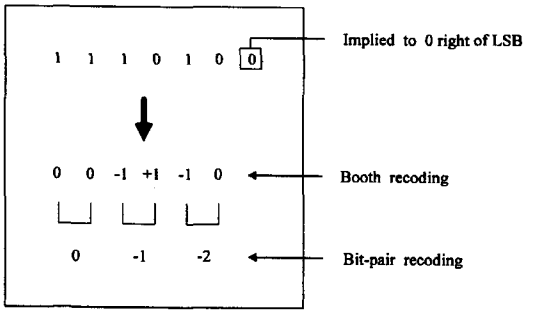


표 1에서 보여진 바와 같이 Booth 재코드화된 인접비트를 비트-쌍으로 하여 각 쌍에 대해서 적당하게 시프트된 하나의 피가수를 얻는다. 가장 우측의 Booth 쌍 (-1, 0)는 SP0에서 -2 × M(피승수)과 같고, 다음의 쌍 (-1, +1)은 SP2에서 -1 × M과 같고, 마지막의 쌍 (0, 0)은 SP4에서 0 × M과 같다. 이러한 선택을 원래의 승수쌍에 고려하면 우측의 '0'과 같이 (1,0)인 경우는 -2 × M을, 우측에 '1'과 같이 (1,0)인 경우는 -1 × M을, 우측에 '1'과 같이 (1,1)인 경우는 0 × M를 선택한다. 이와같은 방법으로 모든 비트쌍에 대해서 선택되는 피승수는 표 2에 나타냈다.

표 2. 승수 비트-쌍 재코딩에 의한 피승수 선택 결정표

Table 2. Table of multiplicand selection decisions by multiplier bit-pair recoding.

Multiplier bit-pair		Previous bit	Multiplicand selected at SPi
i+1	i	i-1	
0	0	0	0 × M
0	0	1	+1 × M
0	1	0	+1 × M
0	1	1	+2 × M
1	0	0	-2 × M
1	0	1	-1 × M
1	1	0	-1 × M
1	1	1	0 × M

Ⅳ. 제안된 광곱셈기 구현

1. 변형된 비트-쌍 재코딩 알고리즘

광의 빠른 속도와 병렬성을 이용할 수 있도록 제안한 광곱셈기는 곱셈의 기본 연산인 덧셈을 병렬로 할 수 있는 기호치환 가산기와 승수 비트-쌍 재코딩 알고리즘을 사용할 수 있도록 몇개의 시프트 레지스터로 구성되고 이의 블럭도는 그림 3에 나타냈다.

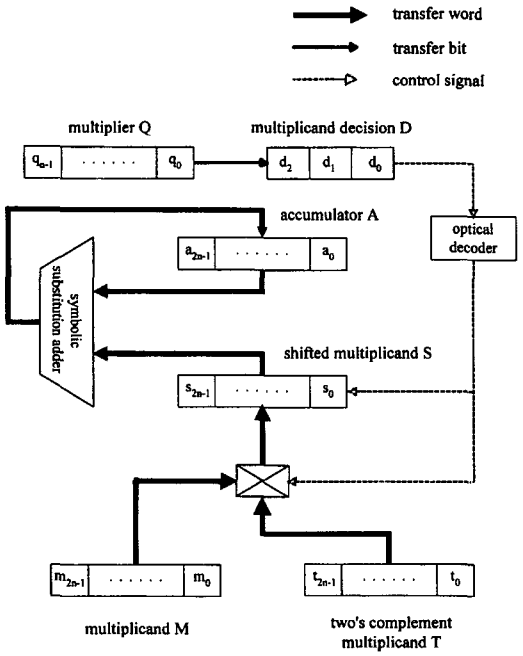


그림 3. 광곱셈기의 블럭도

Fig. 3. Block diagram of the optical multiplier.

그림 3의 블럭도에서 각각 n비트로 구성된 두 수를 곱할때, A는 부분적이 누적되는 2n비트 레지스터이고, M과 T는 각각 피승수 및 이의 2의 보수가 저장되는 2n비트 레지스터이다. 또한 S는 D레지스터 값에 따라 M이나 T의 값이 이동된 후에 우측 시프트되어 저장된다. Q는 승수가 저장되는 n비트 레지스터이고 D는 우측 시프트된 승수 2비트값이 저장되는 3비트 레지스터이다. D레지스터 값은 광디코더에 입력되어 피승수 선택 및 좌측 시프트 횟수를 결정하는 제어신호를 발생한다.

그림 3의 블럭도에 적용된 제안한 광곱셈 알고리즘은 다음과 같다.

(입력과 출력)

- 입력 : 각각 n비트의 피승수 및 승수이고, 음수인 경우는 2의 보수를 사용.
- 출력 : 2n비트의 곱셈결과.

(알고리즘)

(1) 승수는 Q레지스터에, 피승수 및 이의 2의보수는 각각 M레지스터 및 T레지스터에 입력되고, A, S, 그리고 D레지스터는 '0'으로 초기화된다.

(2) Q레지스터를 2비트 우측 시프트함과 동시에 M과 T레지스터를 2비트 좌측시프트한다.

(3) D레지스터 값에 따라 M 혹은 T, 및 S 레지스터에 제어신호를 발생한다.
- 제어신호가 '0'이면 (5)로 간다.
- 제어신호가 '+1'이면 M을 S에 전송한 후에 2비트 우측 시프트한다.
- 제어신호가 '+2'이면 M을 S에 전송한 후에 1비트 우측 시프트한다.
- 제어신호가 '-2'이면 T를 S에 전송한 후에 1비트 우측 시프트한다.
- 제어신호가 '-1'이면 T를 S에 전송한 후에 2비트 우측 시프트한다.

(4) A와 S레지스터 값을 기호치환가산기에서 덧셈한 후에 A레지스터에 저장한다.

(5) Q레지스터의 우측 시프트 횟수가 [n]/2가 될때까지 (2)에서 (4)를 반복한다.

최종출력은 A레지스터의 결과이다.

위의 알고리즘에서 좌우측 시프트는 산술시프트를 의미한다. 알고리즘 (3)에서 D레지스터의 값에 따른 제어신호는 표 2의 승수 비트-쌍 재코딩을 이용하여 피승수를 선택 결정해 주는 값인 '0', '±1', '±2'를 나타낸다. 표 2의 3비트 승수 비트-쌍에 따른 제어값들

은 3비트 승수 비트-쌍을 입력으로 한 3×8 광디코더로부터 쉽게 얻을 수 있고, 이 제어신호들의 기능은 표 3에서와 같다. 즉 광디코더의 출력에 따라 S레지스터로 전송되는 레지스터를 제어하고, S레지스터의 우측시프트 횟수를 제어하게 한다.

표 3. 승수 비트-쌍 재코딩에 의한 제어신호의 기능

Table 3. Functions of the control signals by multiplier bit-pair recoding.

multiplier bit-pair		previous bit	control values	제어신호의 기능	
				S에 전송되는 레지스터	S레지스터의 우측시프트 횟수
i+1	i	i-1			
0	0	0	0	없 음	
0	0	1	1	M	2
0	1	0	1		
0	1	1	2	M	1
1	0	0	-2	T	1
1	0	1	-1	T	2
1	1	0	-1		
1	1	1	0	없 음	

2. 제안한 기호치환 가산기

앞에서 제안된 기호치환 가산기의 덧셈규칙을 부호화하는 방법에는 편광(polarization)부호화, 세기(intensity)부호화 등이 있는데 본 연구에서는 세기부호화의 한 형태인 듀얼-레일(dual-rail) 논리를 이용하여 부호화한다. 왜냐하면 이렇게 부호화된 두 입력면을 직렬 연결하므로써 기호치환 가산기의 구성에 광소자들을 적게 사용하고 시스템의 크기를 작게 구현할 수 있기 때문이다. 덧셈규칙을 부호화한 기준패턴에서 사용되는 듀얼-레일 논리는 논리 '1'과 '0'을 밝고 어두운 세기를 모두 포함한 패턴으로 나타내며 '1'과 '0'에 대한 패턴정의는 그림 4에서와 같다.

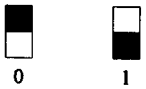


그림 4. 듀얼-레일 논리의 패턴정의

Fig. 4. Pattern definitions for dual-rail logic.

그림 1의 블럭도의 블럭 A 및 블럭 B에 대한 덧셈규칙에 그림 4의 듀얼-레일 부호화를 적용한 기호치환 가산기의 치환규칙은 그림 5에서와 같다.

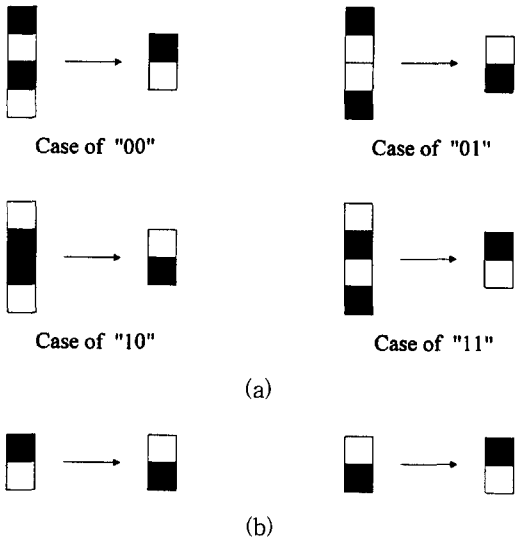


그림 5. 기호치환 가산기의 치환규칙, (a) 그림 1의 블록 A에 대한 치환규칙, (b) 그림 1의 블록 B에 대한 치환규칙

Fig. 5. Substitution rules of symbolic substitution adder: (a) Substitution rules for block A in fig. 1; (b) Substitution rules for block B in fig. 1.

그림 5의 치환규칙에 의한 기호치환 가산기의 블럭 A 및 블럭 B 연산을 위한 기호치환 논리에 기본을 둔 구조는 각각 그림 5(a) 및 그림 5(b)에서 왼쪽의 패턴을 인식하여 오른쪽 패턴으로 치환하는 것이다. 특히 그림 5(a)의 기준패턴의 인식과정은 아래와 같이 4단계로 나눌 수 있다. 여기서, 기준패턴의 인식은 밝은 화소를 기준으로 하고 기준화소의 위치는 가장 아래의 화소로 한다.

(1) 왼쪽의 기준패턴들에 대해서 각 패턴에 따라 입력영상의 복사본 2개를 만든다.

(2) 기준패턴의 각 밝은 화소에 따라 입력영상의 복사본을 아래로 시프트한다.

- 기준패턴이 "00"인 경우 복사본 1은 그대로 두고 복사본 2는 2화소 아래로 시프트한다.
- 기준패턴이 "01"인 경우 복사본 1은 1화소 아래로 시프트하고 복사본 2는 2화소 아래로 시프트한다.
- 기준패턴이 "10"인 경우 복사본 1은 그대로 두고 복사본 2는 3화소 아래로 시프트한다.
- 기준패턴이 "11"인 경우 복사본 1은 1화소 아래로 시프트하고 복사본 2는 3화소 아래로 시프트

한다.

(3) 시프트된 복사본을 중첩시킨다.

(4) 기준화소의 위치에 마스크를 두어 인식한다.

치환과정은 위의 과정으로 인식된 위치에 그림 5(a)에서의 오른쪽 패턴으로 치환하면 된다. 또한 그림 1의 블럭 A의 기준패턴을 인식할 때 집단구분에 의해 얻는 올림수는 광시프트레지스터에 저장하고 이 값을 좌측 시프트해서 그림 5(b)의 기호치환 규칙을 수행하는데 제어신호로 사용한다. 블럭 B에 대한 기호치환 과정은 제어신호가 1인 경우의 셀에 대해서 그림 5(b)의 왼쪽패턴을 인식하여 그 자리에 오른쪽 패턴으로 치환하면 된다. 즉 광인버팅을 구하면 된다. 위의 인식 과정을 광학적으로 구현하기 위한 시스템구성은 그림 6에서와 같다.

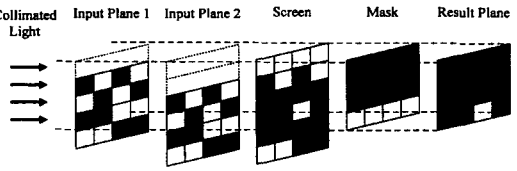


그림 6. 기호치환 가산기의 인식과정을 위한 광시스템
Fig. 6. Optical system for recognition process of SS adder.

그림 6에서 입력면 1과 입력면 2에는 각각 입력영상의 복사본 1과 복사본 2가 기준패턴에 따라 시프트되어 입력된다. 그림 6에서 입력면의 시프트는 앞의 인식 알고리즘에서 어떤 입력영상에 대해 기준패턴 "01"을 인식하는 경우를 보인 것으로 입력면 1은 아래로 1화소를, 입력면 2는 아래로 2화소를 시프트한 것이다. 스크린면에는 입력된 복사본 1과 복사본 2의 중첩패턴이 나타나고 이는 마스크에 의해 원하는 인식 패턴으로 출력된다. 그림 6의 광인식 시스템은 복사본의 시프트를 기준패턴에 따라 미리 입력하고, 입력면 1과 입력면 2를 직렬 연결하여 복사본의 중첩을 얻으므로써 광수동소자들의 수를 줄이고 시스템이 크기를 상당히 작게 할 수 있다. 또한 그림 6의 입력면 1과 입력면 2에 SLM을 사용하면 실시간 인식도 가능할 것으로 사료된다. 각 기준패턴에 대해서 이러한 광인식시스템을 사용한 광학적인 기호치환 가산기의 구성은 그림 7에서와 같다. 여기서, Mi는 거울을, BSi는 beam splitter를, ORS(optical recognition system)는 그림 6의 광인식시스템을 나타낸다.

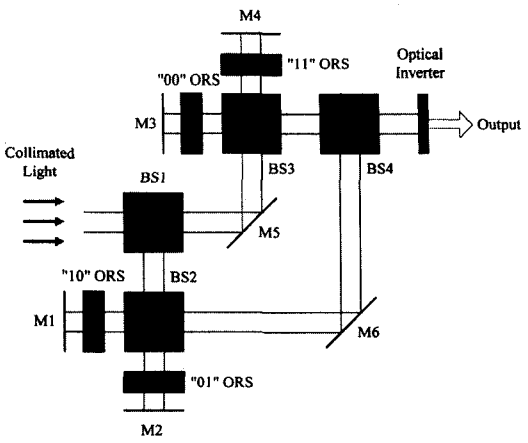
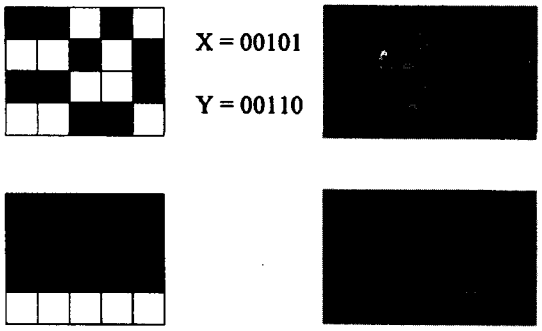


그림 7. 기호치환 가산기의 광학적인 구성도
Fig. 7. Optical organization arrangement of SS adder.

그림 7에서 광인식시스템의 각 블럭은 그림 6의 광시스템을 나타내며 기준패턴에 따라 복사본 1과 복사본 2가 각각 다르게 시프트되어 입력되도록 구성된다. 특히 모든 ORS내의 기준패턴들과 마스크의 크기를 동일하게 하여 치환패턴을 쉽게 얻을 수 있도록 했다. "01"과 "10"의 광인식시스템을 하단에 둔 것은 각각의 인식된 기준패턴에 대한 치환패턴이 인식시스템의 마스크 출력된 패턴과 반전된 패턴이므로 거울(mirror) M6를 마스크된 출력이 1화소 위로 시프트되도록 조정하여 치환패턴을 바로 얻기 위해서 이다. 기준패턴 "00"과 "11"의 마스크된 출력패턴이 각각 나타나는 BS3 다음에서 는 인식된 패턴과 치환패턴이 같으므로 치환과정이 필요없다. 특히 BS4는 각각의 기준패턴에 따른 치환패턴을 합하여 그림 1에서의 중간합을 구하는 것이다. 광인버터(optical inverter)는 그림 1에서 t_i 값이 1인 경우 중간합을 인버팅하기 위한 것이다.

3. 시뮬레이션 및 광실험

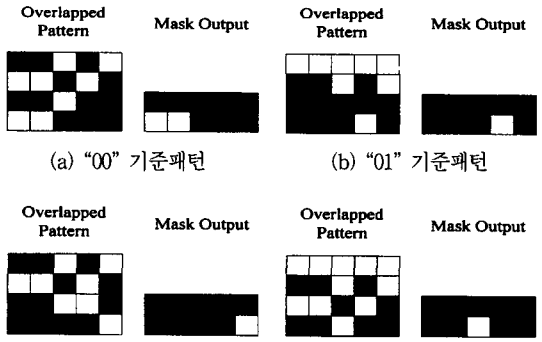
우선 기호치환 가산기를 이용하여 연산하는 예를 고찰하기 위하여 두 데이터 X=00101(5_{10})와 Y=00110(6_{10})에 대해서 그림 4의 듀얼-레일 논리를 이용하여 부호화한 입력패턴과 광인식시스템에 사용되는 마스크는 그림 8에서와 같다. 그림 8(a)와 8(b)는 각각 시뮬레이션과 광실험에 사용할 입력패턴과 마스크를 나타낸다. 특히 그림 8(b)의 입력패턴의 시작과 끝 위치의 세로줄은 출력평면에서 인식된 기준패턴의 위치를 구별하기 위해서 사용하였다.



(a) (b)

그림 8. 입력패턴과 마스크
Fig. 8. Input pattern and mask.

그림 9는 그림 8(a)의 입력패턴을 그림 6의 광인식 시스템에 입력했을 때 각각의 기준패턴에 따른 스크린 면의 중첩패턴과 마스킹 후의 인식된 출력패턴을 시뮬레이션으로 나타낸 것이다.

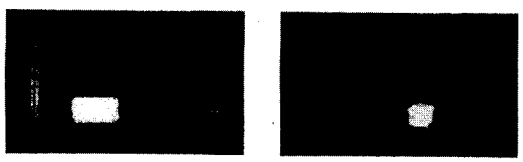


(a) "00" 기준패턴 (b) "01" 기준패턴

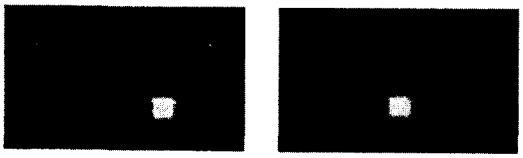
(c) "10" 기준패턴 (d) "11" 기준패턴

그림 9. 시뮬레이션에 의한 중첩패턴과 출력패턴
Fig. 9. Overlapped pattern and output pattern by simulation.

그림 10은 광원으로 코히런트광(laser)을 사용하여 그림 9에서와 같은 과정을 그림 8(b)의 입력패턴과 마스크를 사용하여 실제로 광실험을 통해 얻은 마스크뒤에서의 출력결과들을 나타낸 것이다. 이 결과는 그림 9의 시뮬레이션 결과와 같으며 이 결과로부터 광인식 시스템을 간단하게 구성할 수 있음을 확인하였다.



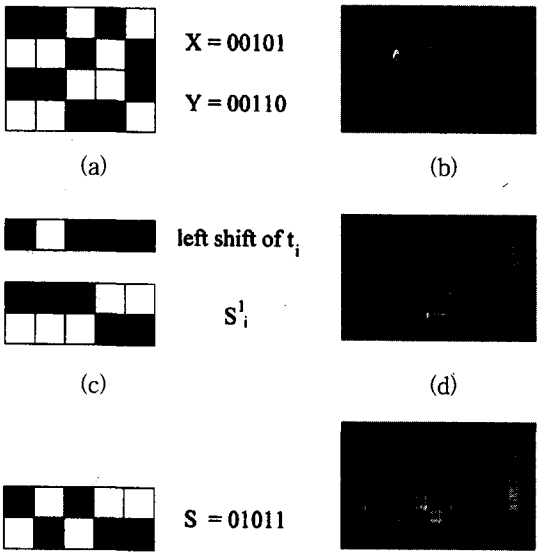
(a) "00" 기준패턴 (b) "01" 기준패턴



(c) "10" 기준패턴 (d) "11" 기준패턴

그림 10. 기준패턴에 따른 광인식시스템의 출력결과
Fig. 10. Output results of optical recognition system for each reference pattern.

그림 11은 두 데이터 X=00101(5_{10})와 Y=00110(6_{10})를 기호치환 가산기를 이용하여 연산하는 과정을 시뮬레이션과 그림 7의 광시스템으로 광실험하여 얻은 결과들을 나타낸 것이다.



(a) (b)

(c) (d)

(e) (f)

그림 11. 기호치환 가산기의 연산과정
Fig. 11. Operation process of SS adder.

그림 11에서 (a), (c), (e)는 시뮬레이션한 결과를, (b), (d), (f)는 각각에 대한 광실험 결과를 나타낸다. (a)와 (b)는 각각 두 입력데이터를 듀얼-레일 부호화한 입력패턴을 나타낸 것이고, (c)와 (d)는 각각 그림 5(a)의 기준패턴들에 대한 기호치환규칙을 (a)와 (b)의 입력값에 적용하여 얻은 s^1_i를 나타낸다. (e)와 (f)는 좌측시프트된 레지스터값을 제어값으로 해서 그림 5(b)의 기호치환규칙을 수행한 후의 최종결과 값인 s_i를 나타낸다.

그림 12는 그림 3에 적용된 제안된 곱셈알고리즘에 의한 피승수 X=0101(5_{10})와 승수 Y=0110(6_{10})의 곱셈

과정을 보여준다. 그림 12(a)는 각 레지스터의 초기상태를 듀얼-레일 부호화를 사용하여 나타낸 것으로 피승수는 M레지스터에, 승수는 Q레지스터에 입력되었다. 그림 12(b)는 Q값을 2비트 우측시프트한 제어신호인 D레지스터값(-2)에 따른 첫 번째 사이클 동안의 변화된 레지스터값들을 나타낸다. 이때 A와 S레지스터의 덧셈은 제안된 기호치환 가산기를 통하여 수행된다. 그림 12(c)는 D레지스터값(+2)에 따른 두 번째 사이클 동안의 레지스터값들을 나타내고 A레지스터의 값이 최종 결과값인 $00011110(30_{10})$이 된다.

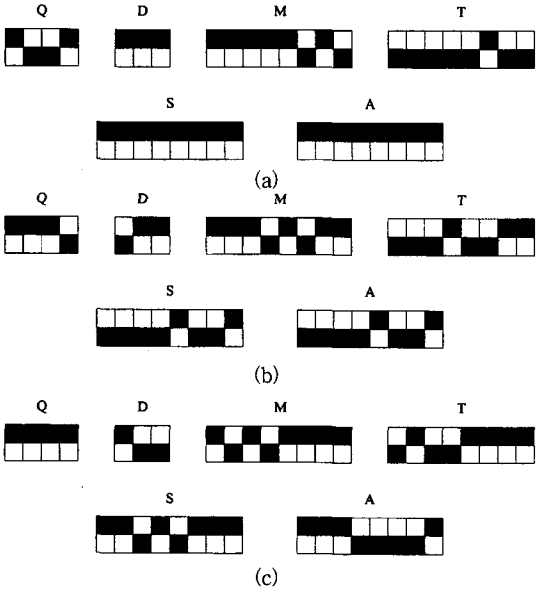


그림 12. 듀얼-레일 부호화된 기호치환 규칙을 이용한 승수 비트-쌍재코딩에 의한 곱셈의 과정, (a) 각 레지스터의 초기상태, (b) 첫 번째 사이클 후의 레지스터 상태, (c) 두번째 사이클 후의 레지스터 상태

Fig. 12. Process of the multiplication by multiplier bit-pair recoding using dual-rail coded symbolic substitution: (a) Initial configuration status of each register, (b) register status after the first cycle, (c) register status after the second cycle.

V. 결 론

변형된 승수 비트-쌍 재코딩 알고리즘에 따른 광곱셈기의 광학적인 구현은 몇 개의 레지스터와 광디코더 및 기호치환 가산기로 구성될 수 있음을 보였다. 또한 승수 비트-쌍 재코딩에 의한 제어비트를 입력으로 한 광디코더의 출력에 따라 피승수와 이의 2의보수를 선택하고 시프트 횟수를 제어할 수 있도록 하여 곱셈 연산의 병렬성을 높이도록 광곱셈기를 구성했다. 승수 비트-쌍 재코딩 방법의 사용으로 최소한 부분적의 수를 반이상 줄일 수 있어서 Booth알고리즘 사용할 때보다 최소한 2배이상 빠른 곱셈기를 구현할 수 있다. 기호치환 가산기의 가산규칙을 듀얼-레일 논리로 부호화하고, 기준패턴에 따라 시프트된 복사본 2개를 직렬로 나열하여 중첩패턴을 얻도록 가산기의 기호치환 인식시스템을 구성하여 기호치환 시스템의 수동소자를 줄이고 시스템의 크기도 작게 했다. 그래서 4개의 기준패턴에 따라 구성된 기호치환 시스템들을 병렬로 연결하여 기호치환 가산기를 구성하므로써 크기를 상당히 작게 할 수 있도록 했다.

제안한 광곱셈기는 곱셈연산에서 광의 병렬성을 충분히 이용하여 사용될 수 있음을 광 실험과 시뮬레이션을 통하여 확인하였다.

참 고 문 헌

[1] D. Ferrari and R. Stefanelli, "Some new schemes for parallel multipliers", Alta freq., vol. 38, no. 11, pp. 843-852, 1969.

[2] 김용성, 조원경, "고속 그래픽처리를 위한 잉여수계 승산기 설계에 관한 연구", 전자공학회 논문지, 33권, B편, 1호, pp. 25-37, 1996

[3] N. Takagi, H. Yasuura, and S. Yajima, "High-speed VLSI multiplication algorithm with a redundant binary addition tree", IEEE Trans. Comput., vol. C-34, no. 9, pp. 789-796, 1985.

[4] P. S. Guilfoyle, "Systolic acousto-optic binary convolver", Opt. Eng., vol. 23, no. 1, pp. 20-25, 1985.

[5] S. Barua, "High-speed multiplier for digital signal processing", Opt. Eng., vol. 30, no. 12, pp. 1997-2002, 1991.

[6] V. Chandran, T. F. Krile, and J. F. Walkup, "Optical techniques for real-time binary multiplication", Appl. Opt., vol. 25, no. 14, pp. 2272-2276, 1986.

[7] K. A. Ghoneim and D. Casasent, "High-accuracy pipelined iterative-tree optical multiplication", Appl. Opt., vol. 33, no. 8, pp. 1517-1527, 1994.

[8] K. Hwang and A. Louri, "Optical multiplication and division using modified-signed-digit symbolic substitution", Opt. Eng., vol. 28, no. 4, pp. 364-372, 1989.
[9] Yi-Der Wu, Dah-Shi Shen, V. K. Bykovsky, I. Rosetti, and M. A. Fiddy, "Digital optical computing with magneto-optic spatial modulators : a new and efficient multiplication algorithm", Appl. Opt., vol. 33, no. 32, pp. 7572-7578, 1994.
[10] W. Stallings, *Computer organization and architecture*, Macmillan publishing company, pp. 212-219, 1987.
[11] Jalil Fadavi-Ardekani, "M×N Booth encoded multiplier generator using optimized Wallace trees", IEEE Trans. on VLSI Systems, vol. 1, no. 2, pp. 120-125, 1993.
[12] 김영민, 조진호, "32×32비트 고속 병렬 곱셈기 구조", 전자공학회 논문지, 31권, B편, 10호, pp. 67-72, 1994
[13] 조웅호, 김수중, "1-비트 기호치환 가산기의 광학적인 구현", 전자공학회 논문지, 31권, A편, 8호, pp. 999-1006, 1994

저 자 소 개

曺 雄 鎬(正會員)
1959年 10月 22日生. 1982년 2월 경북대학교 전자공학과 졸업(학사). 1984년 2월 영남대학교 대학원 전자공학과 졸업(석사). 1993년 8월 경북대학교 대학원 전자공학과 졸업(박사). 1986년 3월 ~ 현재 대구공업대학 전자계산과 부교수. 주관심분야는 광컴퓨팅, 광신호처리 등임

盧 德 樹(正會員)
1954年 1月 14日生. 1977년 2월 경북대학교 전자공학과 졸업(학사). 1983년 2월 경북대학교 대학원 전자공학과 졸업(석사). 1996년 8월 경북대학교 대학원 전자공학과 졸업(박사). 1978년 2월 ~ 1979년 8월 (주) 대한전선 TV 개발부. 1983년 3월 ~ 현재 경일대학교 전자정보공학과 교수. 주관심분야는 광신호처리, 광컴퓨팅 및 패턴인식 등임

金 鐘 允(正會員) 第 35卷 D編 第 3號 參照
현재 경북대학교 대학원 전자공학과 박사과정 재학중

金 秀 重(正會員) 第 33卷 B編 第 7號 參照
현재 경북대학교 전자전기공학부 정교수