(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.
G06T 1/00 (2006.01)

(45) 공고일자 2007년01월10일
(11) 등록번호 10-0665485
(24) 등록일자 2006년12월29일

(21) 출원번호 10-1999-7007123
(22) 출원일자 1999년08월06일
심사청구일자 2003년12월05일
번역문 제출일자 1999년08월06일
(86) 국제출원번호 PCT/JP1998/005542
국제출원일자 1998년12월08일
(65) 공개번호 10-2000-0070858
(43) 공개일자 2000년11월25일
(87) 국제공개번호 WO 1999/30276
국제공개일자 1999년06월17일

(81) 지정국 국내특허 : 일본, 대한민국, 미국,

(30) 우선권주장 97-336717 1997년12월08일 일본(JP)

(73) 특허권자
소니 가부시끼 가이샤
일본국 도쿄도 시나가와쿠 키타시나가와 6쵸메 7반 35고

(72) 발명자
이자와다카시
일본도쿄도시나가와꾸기따시나가와6쪼메7-35소니가부시끼가이샤내

(74) 대리인
이병호
정상구
신현문
이범래

심사관 : 이상헌

전체 청구항 수 : 총 7 항

(54) 디지털 신호 처리 장치 및 디지털 신호 처리 방법

(57) 요약

각각이 복수의 기억 요소로 구성된 복수의 입력 기억 수단과, 입력 기억 수단을 제어하는 입력 제어 수단과, 입력 기억 수단의 복수의 기억 요소에 대응하는 복수의 요소 연산 수단을 갖고, 기억 요소에 기억된 데이터를 병렬로 연산 처리하는 연산 수단과, 연산 수단의 복수의 요소 연산 수단에 대응하는 복수의 기억 요소를 갖고, 해당 각 기억 요소에 대응하는 요소 연산 수단의 연산 결과 데이터를 유지하는 데이터 기억 수단과, 연산 수단의 복수의 요소 연산 수단에 대응하는 복수의 기억 요소로 구성되고, 상기 연산 결과 데이터를 기억하는 복수의 출력 기억 수단과, 출력 기억 수단을 제어하는 출력 제어 수단과, 제어 프로그램에 따라서, 입력 기억 수단, 연산 수단, 데이터 기억 수단, 출력 기억 수단을 제어하는 제어 수단으로 이루어진다.

대표도

도 1

특허청구의 범위

청구항 1.

단일의 공통 명령으로 복수의 데이터 처리를 병렬로 실행하는 디지털 신호 처리 장치에 있어서,

각각이 복수의 기억 요소로 구성된 복수의 입력 기억 수단과,

상기 입력 기억 수단을 제어하는 입력 제어 수단과,

상기 입력 기억 수단의 복수의 기억 요소에 대응하는 복수의 요소 연산 수단을 갖고, 상기 입력 기억 수단의 각 기억 요소에 기억된 데이터에 대하여 병렬로 연산 처리를 행하는 연산 수단과,

상기 연산 수단의 복수의 요소 연산 수단에 대응하는 복수의 기억 요소를 갖고, 상기 각 기억 요소에 대응하는 요소 연산 수단의 연산 결과 데이터를 보유하는 데이터 기억 수단과,

상기 연산 수단의 복수의 요소 연산 수단에 대응하는 복수의 기억 요소로 구성되고, 상기 연산 결과 데이터를 기억하는 복수의 출력 기억 수단과,

상기 출력 기억 수단을 제어하는 출력 제어 수단과,

일련의 제어 프로그램에 따라서, 상기 입력 기억 수단, 연산 수단, 데이터 기억 수단, 출력 기억 수단을 제어하는 1개의 제어 수단을 구비하는 것을 특징으로 하는 디지털 신호 처리 장치.

청구항 2.

제 1 항에 있어서, 입력 데이터를 시리얼/패러랠(serial-parallel)로 변환하여 데이터 입력 수단에 할당하는 것을 특징으로 하는, 디지털 신호 처리 장치

청구항 3.

제 1 항에 있어서, 입력 데이터가 화상 데이터이고,

상기 입력 제어 수단은, 상기 연산 수단의 각각의 요소 연산 수단에 대응하는 상기 입력 기억 수단의 각각의 기억 요소에 화상 데이터를 입력하는지, 혹은 스킵하는지를 결정하는 스킵 조작 수단(skip operating means)과,

상기 화상 데이터의 보간 연산을 하기 위한 중심이 되는 대상 화소의 근방의 화소 데이터를 인출하는 화소 인출 수단을 갖고,

상기 연산 수단은, 상기 대상 화소가 상기 데이터 입력 수단의 어느 것에 할당되는 지를 나타내는 제 1 플래그를 나타내는 제 1 플래그 수단과,

상기 제 1 플래그 수단이 나타내는 제 1 플래그를 사용하여 상기 입력 제어 수단의 스킵 조작 수단의 제어를 선택시키는 제 2 플래그를 연산하는 제 2 플래그 수단을 구비하는 것을 특징으로 하는, 디지털 신호 처리 장치.

청구항 4.

제 3 항에 있어서, 상기 출력 제어 수단은, 상기 연산 수단의 각각의 요소 연산 수단에 대응하는 상기 출력 기억 수단의 각각의 기억 요소에 기억되는 연산 결과 데이터를 출력하는지, 출력하지 않는지를 결정하는 출력 조작 수단을 갖고,

상기 연산 수단은, 상기 제 1 플래그 수단이 나타내는 제 1 플래그를 사용하여 상기 출력 제어 수단의 출력 조작 수단의 제어를 선택하는 제 3 플래그를 연산하는 제 3 플래그 수단을 구비하는 것을 특징으로 하는, 디지털 신호 처리 장치.

청구항 5.

일련의 데이터를 적어도 2조로 분할하여 입력하는 입력 과정과, 이 입력 과정에서 입력된 각 조의 데이터에 공통의 병렬 연산 처리를 행하는 연산 과정과, 이 연산 과정에서의 연산 처리 결과 데이터를 기억하는 결과 데이터 기억 과정과, 이 결과 데이터 기억 과정에서 기억된 데이터를 입력시의 조별로 각각 선택하여 출력하는 출력 과정과, 1개의 제어 수단에 의해 실행되는 일련의 제어 프로그램에 따라서, 상기 입력 과정, 연산 과정, 데이터 기억 과정, 출력 과정을 제어하는 제어 과정을 포함하는 것을 특징으로 하는 디지털 신호 처리 방법.

청구항 6.

제 5 항에 있어서, 입력 데이터가 화상 데이터이고,

상기 입력 과정에는, 각 조의 입력 데이터를 그대로 입력하는지, 일정한 비율로 스킵하는지를 결정하는 스킵 선택 과정과, 보간 연산을 하기 위한 중심이 되는 대상 화소의 근방의 화소 데이터를 인출하는 화소 데이터 인출 과정과, 상기 대상 화소가 어떤 조에 속해 있는가에 의해서 상기 입력 과정에서의 입력 데이터의 선택을 제어하는 입력 제어 과정을 포함하는 것을 특징으로 하는, 디지털 신호 처리 방법.

청구항 7.

제 6 항에 있어서, 상기 출력 과정에는, 각 조의 연산 결과 데이터를 그대로 출력하는지, 출력하지 않는지를 결정하는 출력 선택 과정과, 상기 연산 결과 데이터가 어떤 조에 속해 있는가에 의해서 상기 출력 과정에서의 출력 데이터의 선택을 제어하는 출력 제어 과정을 포함하는 것을 특징으로 하는, 디지털 신호 처리 방법.

명세서

기술분야

본 발명은, 예를 들어 화상의 확대/축소의 처리를 행하는 화상 처리 장치에 적용 가능한 디지털 신호 처리 장치 및 디지털 신호 처리 방법에 관한 것이다.

배경기술

화상 데이터에 대한 신호 처리의 분야에서는, 1 매의 화상을 구성하는 모든 화소에 대하여 동일한 연산 처리를 시행하는 일이 많다. 많은 데이터에 대하여 동일한 연산 처리를 고속으로 실행하기 위해서, SIMD(Single Instruction Multiple Data Stream, 단일 명령 복수 데이터)형 아키텍쳐가 제안되고, 화상 신호 처리에 한하지 않고 넓은 분야에서 이용되어 있다. SIMD 형 아키텍쳐는, 연산 장치를 필요한 개수만큼 나열하여, 각각의 연산 장치가 동일한 명령에 따라서 동작하도록 한 구성이다. 따라서, 각각의 연산 장치에 각각의 데이터를 제공하면, 각각의 데이터에 대한 연산 결과가 한번에 얻어진다.

SIMD 형 처리 장치의 화상 처리에의 적용으로서는 예를 들어, 「Kurokawa et al., "비디오 포맷 변환 n을 위한 5. 4GOPS 리니어 어레이 아키텍쳐(5. 4 GOPS Linear Architecture DSP for Video Format Conversion n)", IEEE 1996/Feb. ISSCC, FP15. 7.」에 개시된 장치가 공지되어 있다. 이 병렬 프로세서는 예를 들어 도 18에 도시되어 있는 것과 같은 것이다.

도 18의 장치는, 입력 화상 데이터(1), 입력 프레임 메모리(2), SIMD 형 화상 처리 프로세서(병렬 프로세서: 3a, 3b), 출력 프레임 메모리(14), 출력 화상 데이터(15)로 구성된다. 또한, 병렬 프로세서(3a, 3b)의 각각은, 입력 포인터(4), 입력 SAM(시리얼 액세스 메모리)부(5), 데이터 메모리부(7), ALU 어레이부(8), 출력 SAM 부(9), 출력 포인터(11), 프로그램 제어부(12) 등의 각 부로 구성된다.

이 중, 입력 SAM 부(5), 데이터 메모리부(7), ALU 어레이부(8), 출력 SAM 부(9)는 전체로 리니어 어레이(직선 배열)형에 다수 병렬화된 요소 프로세서군을 구성하고 있어, 이 다수의 요소 프로세서는, 프로그램 제어 기능 중에 포함되는 공통의 하나의 프로그램 제어부(12)에 의해 연동하여 제어(SIMD 제어)된다. 프로그램 제어부(12)에는 프로그램 메모리와 그 프로그램을 제어하기 위한 시퀀스 제어 회로 등이 있어, 프로그램 메모리에 미리 기록된 프로그램에 따라서 각 부분에 접속되어 있는 각종 제어 신호를 발생하여 각 부분을 제어한다.

프로그램 제어부(12), 데이터 메모리부(7), ALU 어레이부(8)를 정리하여 프로세서 블럭으로 하고, 이 프로세서 블럭을 다단으로 구성하는 것에 의해, 그 단수에 비례하여 처리 능력을 향상시킬 수 있다. 도 18에 도시된 장치에서, 각 프로세서 블럭 내에서는 SIMD 형 처리 장치이지만, 장치 전체로서는 복수의 프로그램을 병렬로 처리하는 것이 가능한 MIMD (Mu1tiple Instruction Multip1e Data Stream, 복수 명령 복수 데이터)형 처리 장치이다.

보통의 프로세서에서, 그 하드웨어는 일반적으로 워드 처리 프로세서이고, 워드를 단위로서 처리하지만, 도 18의 사선으로 나타낸 세로의 가늘고 긴 범위로 나타내는 한개의 요소 프로세서는, 입력 SAM 부(5), 데이터 메모리부(7), 출력 SAM 부(9)는 메모리의 「컬럼」으로 되어 있고, 또한 ALU 어레이부(8)는 1 비트 ALU이고, 사실상 전가산기를 주체로 한 회로로 되어 있다. 그 때문에, 보통의 프로세서와는 다른 비트 처리 프로세서이고, 비트를 단위로서 처리한다. 보통의 CPU에서 말하는 8 비트 머신이라든가 16 비트 머신이라는 표현에 대응시키면 1 비트 머신이다. 비트 처리 프로세서는 하드웨어가 작고, 보통으로는 실현할 수 없는 정도 다수의 병렬수를 실현할 수 있기 때문에, 화상용의 경우, 요소 프로세서 직선 배열의 병렬수는, 영상 신호의 1 수평 주사 기간의 화소수(H)에 일치시키고 있다.

도 18에 도시된 프로세서(3a)에 의한 화상 처리는 다음과 같은 순서로 행하여진다. 수평 주사 액티브 기간 내에, 입력 SAM 부(5)에 1 수평 주사선분의 입력 데이터를 인출하고, 수평 주사부 블랭킹 기간으로 입력 SAM 부(5)로부터 데이터 메모리부(7)에의 전송이 행하여진다. 데이터 메모리부(7), ALU 어레이부(8)로서는, 프로그램에 따라서 연산 처리가 행하여진다. 연산 처리 종료후, 처리 결과를 출력 SAM 부(9)에 전송하여, 수평 주사 액티브 기간 내에, 출력 SAM 부(9)로부터 1 수평 주사선분 데이터를 출력한다. 상기 처리중은 각 부는 전부 병렬로 동작하고 있다.

이러한 프로세서(3a)에서 화상 처리를 행하는 경우, 처리할 수 있는 화상의 크기는 요소 프로세서군의 수에 의존하고 있다. 요소 프로세서군의 수 이상의 화상 크기를 가지는 데이터를 처리하는 경우에는, 도 18에 도시된 바와 같이, 동일한 프로세서(3a 및 3b)를 2개 이상 나열하고, 프레임 메모리(2)를 통하여 각 프로세서의 입력 SAM 부(5)에 입력하는 데이터를 제어하는 복잡한 하드 구성을 갖게 된다.

그런데, 프로세서에서 화상 처리, 특히 화소수 변환을 실현할 때, 프로세서의 입출력 화상 크기를 고려할 필요가 있다. 입출력 화상 크기가 프로세서의 요소 프로세서군의 수 보다도 적으면, 1개의 프로세서에서 회로를 구성하게 된다. 또한, 입출력 화상 크기가 프로세서의 요소 프로세서군의 수 보다도 많으면, 복수개의 프로세서에서 회로를 구성하여 실현하게 된다.

오늘날, 요소 프로세서군 이상의 해상도(화소)의 화상 처리를 요구되고 있다. 도 18은, 그것을 실현하기 위해서, 2개의 프로세서(3a 및 3b)를 사용하고 있는 예이다. 그렇지만, 이와 같이 프로세서를 복수개 사용하기 때문에, 하드웨어의 구성을 복잡, 거대화시키는 문제점이 생긴다. 또한, 비용도 쓸데없이 많이 들게 되는 결과가 된다. 또한, 화소수 변환 처리를 행하는 경우, 프로세서를 연결하는 회로 구성도 중요하게 되지만, 회로 구성이 복잡하게 되어 버린다.

따라서, 본 발명의 목적은, 이러한 문제점을 해소하여, 회로 구성이 간단하고 비용이 들지 않는 프로세서를 사용하는 것이 가능한 화상 처리 장치 등의 디지털 신호 처리 장치 및 디지털 신호 처리 방법을 제공하는 것이다.

발명의 상세한 설명

본 발명은, 단일의 공통 명령으로 복수의 데이터 처리를 병렬로 실행하는 디지털 신호 처리 장치에 있어서,

각각이 복수의 기억 요소로 구성된 복수의 입력 기억 수단과,

입력 기억 수단을 제어하는 입력 제어 수단과,

입력 기억 수단의 복수의 기억 요소에 대응하는 복수의 요소 연산 수단을 갖고, 입력 기억 수단의 각 기억 요소에 기억된 데이터에 대하여 병렬로 연산 처리를 행하는 연산 수단과,

연산 수단의 복수의 요소 연산 수단에 대응하는 복수의 기억 요소를 갖고, 해당 각 기억 요소에 대응하는 요소 연산 수단의 연산 결과 데이터를 유지하는 데이터 기억 수단과,

연산 수단의 복수의 요소 연산 수단에 대응하는 복수의 기억 요소로 구성되고, 연산 결과 데이터를 기억하는 복수의 출력 기억 수단과,

출력 기억 수단을 제어하는 출력 제어 수단과,

제어 프로그램에 따라서, 입력 기억 수단, 연산 수단, 데이터 기억 수단, 출력 기억 수단을 제어하는 제어 수단을 구비하는 것을 특징으로 하는 디지털 신호 처리 장치이다.

또한, 본 발명은, 일련의 데이터를 적어도 2조로 분할하여 입력하는 입력 과정과, 이 입력 과정에서 입력된 각 조의 데이터에 공통의 보통의 병렬 연산 처리를 행하는 연산 과정과, 이 연산 과정에서의 연산 처리 결과 데이터를 기억하는 결과 데이터 기억 과정과, 이 결과 데이터 기억 과정에서 기억된 데이터를 입력시의 조별로 각각 선택하여 출력하는 출력과정을 갖는 것을 특징으로 하는 디지털 신호 처리 방법이다.

상기의 장치 및 방법에 의하면, 병렬 프로세서 1개로, 요소 프로세서군의 2배의 크기까지의 화상 처리 장치를 구성할 수 있어, 적은 병렬 프로세서로 회로를 구성할 수 있기 때문에, 주변 회로를 포함시킨 전체의 화상 처리 장치를 간단히 구성할 수 있다.

또한, 데이터 메모리부, ALU 어레이부 등을 공유하는 구성으로 되어 있기 때문에, 복수개의 병렬 프로세서를 사용하는 경우와 비교하여, 병렬 프로세서가 차지하는 비율이 적어져, 프로세서간의 접속도 없어지기 때문에, 전체적으로 작은 회로로 구성할 수 있다. 또한, 회로 전체로서의 부품 개수도 감소시키는 것이 가능하여, 이것이 비용 삭감으로 이어진다.

실시예

이하, 본 발명에 관계되는 디지털 신호 처리 장치 및 디지털 신호 처리 방법을 화상 처리에 대하여 적용한 일 실시 형태를 첨부 도면을 참조하여 상세히 설명한다. 도 1은, 본 발명에 의한 병렬 화상 처리 프로세서에 의해 구성된 화상 처리 장치의 일 실시 형태를 나타낸다.

도 1의 화상 처리 장치는, 입력 화상 데이터(1), 입력 프레임 메모리(2), SIMD 형 화상 처리 프로세서(3), 출력 프레임 메모리(14), 출력 화상 데이터(15)로 구성된다. 또한, SIMD 형 화상 처리 프로세서(병렬 프로세서: 3)는, 입력 포인터(4), 입력 1 SAM 부(5), 입력 2 SAM 부(6)[이하, 5와 6을 2개로 나눌 필요가 없는 경우는 입력 SAM 부(5, 6)라 기술한다], 데이터 메모리부(7), ALU 어레이부(8), 출력 1 SAM 부(9), 출력 2 SAM 부(10)[이하, 9, 10을 2개로 나눌 필요가 없는 경우는 출력 SAM 부(9, 10)라 기술한다], 출력 포인터(11), 프로그램 제어부(12), 요소 프로세서군(13)으로 구성되어 있다.

이하, 도 1의 각 부에 관해서 각각 설명한다.

입력 화상 데이터(1), 출력 화상 데이터(15)

입력 화상 데이터(1)는, 실제로 화상 처리를 행하는 기초가 되는 화상 데이터이다. 출력 화상 데이터(15)는, 처리후의 화상 데이터이다. 이 데이터의 포맷은, 컴퓨터 디스플레이와 같은 RGB 형식이어도 좋고, NTSC 같은 텔레비젼 신호라도 좋다. 입력 화상 데이터의 크기는, 도 1과 동일한 병렬 프로세서가 1개인 경우, 요소 프로세서군(13)의 수의 2배 까지의 크기로 한정된다.

입력 프레임 메모리(2)

입력 프레임 메모리(2)는, 메모리로 구성되어 있어도 좋고, 소프트웨어로 실현하기 위해서, 파일로서 존재하고 있어도 좋다.

입력 프레임 메모리(2)는, 입력 화상 데이터(1)로부터의 데이터를 입력 수단으로 하고, 입력 1 SAM 부(5), 입력 2 SAM 부(6)로 데이터를 출력하는 역활을 다 한다.

그 때, 입력 화상 데이터(1)를, 짝수번째의 화소 데이터와 홀수번째의 화소 데이터로 나눠, 짝수번째의 화소 데이터를 입력 1 SAM 부(5), 홀수번째의 화소 데이터를 입력 2 SAM 부(6)로 분배하여 공급한다. 즉, 입력 화상 데이터의 화소가 그 순서에 대응하여 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10....이라고 번호를 붙였을 때에, 입력 1 SAM 부에는, 0, 2, 4, 6, 8, 10....이 공급되고, 입력 2 SAM 부에는, 1, 3, 5, 7, 9....가 공급된다.

SIMD 형 화상 처리 프로세서(3)

SIMD 형 화상 처리 프로세서(3)는, 하드웨어로 구성되어 있어도 좋고, 소프트웨어로 실현되어 있어도 좋다. 입력 화상 데이터(1)의 입력 프레임 메모리(2)를 통한 것을 입력 수단으로 하고, 출력 프레임 메모리(14)를 통하여 출력 화상 데이터(15)를 출력 수단으로서 갖는 구성으로 되어 있다.

입력 포인터(4), 출력 포인터(11)

입력 포인터(4), 출력 포인터(11)는, SIMD 형 화상 처리 프로세서(3)의 내부에 존재하고 있고, 하드웨어의 메모리로서 구성되어 있어도 좋고, 소프트웨어상에서 실현되어 있어도 좋다.

입력 포인터(4)는, ALU 어레이부(8)의 기록 어드레스가 입력 포인터(4)를 찾고 있을 때에, ALU 어레이부(8)의 연산 결과를 입력수단으로 하고, 입력 1 SAM 부(5), 입력 2 SAM 부(6)를 제어한다. 입력 1 SAM 부(5)와 입력 2 SAM 부(6)의 제어용의 입력 포인터를 따로따로 가지고 있어도 좋고, 동일한 포인터 어드레스라도 좋다.

출력 포인터(11)는, ALU 어레이부(8)의 기록 어드레스가 출력 포인터(11)를 가리키고 있을 때에, ALU 어레이부(8)의 연산 결과를 입력 수단으로 하고 출력 1 SAM 부(9), 출력 2 SAM 부(10)를 제어한다. 출력 1 SAM 부(9)와 출력 2 SAM 부(10)제어용의 출력 포인터를 따로따로 가지고 있어도 좋고, 동일한 포인터 어드레스라도 좋다.

입력 SAM 부(5, 6), 출력 SAM 부(9, 10)의 제어의 방법에 관해서, 도 2(a, b), 도 3(a,b,c,d)을 참조하여 설명한다.

도 2(a, b)는, 입출력 포인터의 구조를 나타내고 있다. 도 2(a)는, 입출력 포인터가 통상의 상태인 때에, 이 때에는 IR(Input Register)/OR(Output Register)의 요소가 신호적으로 이어진 상태이다.

도 2(b)는, 입출력 포인터가 스킵 모드(skip mode)인 때에, 이때에는, IR/OR의 패스를 통과하지 않고서 스킵하는 구조로 되어 있다. IR/OR의 패스는, 입력 SAM 부(5, 6)/출력 SAM 부(9, 10)에 상당한다.

도 3(a,b,c,d)은, 입출력 SAM 부(5, 6)와 입출력 포인터의 관계를 나타내는 도면으로 되어 있다. 도 3(a)는, 입력 화소 데이터의 예이고, 각 화소를 A, B, C, …로 나타내고 있다.

도 3(b)은, 모든 IR/OR의 요소가 통상 상태의 경우이고, 이때, 도 3(a)의 입력 화소 데이터는 연속하여 각 요소에 들어 간다.

도 3(c)은, 3개에 1개의 비율로 도 2(b)의 스킵 모드가 발생한 경우이고, 이 경우에는, 3개에 1개씩 IR/OR의 요소가 스킵된다. 이 도면에서는, 도 3(b)의 "C"가 받아들이는 장소가 스킵되어 있고, 입력 화소 데이터가 어긋난 상태로 요소에 대입된다.

도 3(d)은, 2개에 1개의 비율로 도 2(b)의 스킵 모드가 발생한 경우이고, 이 경우 2개에 1개씩 IR/OR의 요소가 스킵된다. 이 결과, 스킵된 곳에서 1개씩 차이나서 요소에 화소가 대입된다.

입력 1 SAM 부(5), 입력 2 SAM 부(6)

입력 1 SAM 부(5), 입력 2 SAM 부(6)는 SIMD 형 화상 처리 프로세서(3)의 내부에 존재하고 있고, 하드웨어로 구성되어 있어도 좋고, 소프트웨어의 파일로서 구성되어 있어도 좋다. 입력 SAM 부(5, 6)는 입력 프레임 메모리(2)에 의해서 분배된 입력 화상 데이터를 입력 수단으로 하고, 입력 포인터(4)를 제어 수단으로 하고, 데이터 메모리부(7)로 데이터를 공급한다.

데이터 메모리부(7)

데이터 메모리부(7)는, SIMD 형 화상 처리 프로세서(3)의 내부에 존재하고 있고, 하드웨어의 메모리로서 구성되어 있어도 좋고, 소프트웨어의 메모리 배열로서 구성되어 있어도 좋다. 데이터 메모리부(7)는, 1 요소 프로세서당 수백 비트의 메모리로 구성되어 있고, 프로그램 제어부(12)의 제어 코드에 따라서, 입력 SAM 부(5, 6)로부터의 데이터를 유지하거나, ALU 어레이부(8)로 데이터를 보내어 연산을 하고, 그 결과를 유지하기도 한다.

ALU 어레이부(8)

ALU 어레이부(8)는, SIMD 형 화상 처리 프로세서(3)의 내부에 존재하고 있고, 하드웨어로 구성되어 있어도 좋고, 소프트웨어로 구성되어 있어도 좋다. ALU 어레이부(8)는, 데이터 메모리부(7)로부터의 데이터를 입력 수단으로 하고, 프로그램 제어부(12)로 지정된 어드레스 포인터에 연산 결과를 기록한다. 어드레스 포인터가 데이터 메모리부(7)인 경우는, 데이터 메모리부(7)의 지정된 어드레스에 결과가 기록되고, 또한 어드레스 포인터가 입력 포인터(4), 출력 포인터(11)인 경우에는, 입력 포인터(4), 출력 포인터(11)에 결과가 기록된다.

출력 1 SAM 부(9), 출력 2 SAM 부(10)

출력 1 SAM 부(9), 출력 2 SAM 부(10)는 SIMD 형 화상 처리 프로세서(3)의 내부에 존재하고 있어, 하드웨어로 구성되어 있어도 좋고, 소프트웨어로 실현하는 경우에는 파일로서 구성되어 있어도 좋다. 출력 SAM 부(9, 10)는, 프로그램 제어부(12)의 제어 코드의 기록 어드레스가 출력 SAM 부(9, 10)를 가리키고 있는 경우, ALU 어레이부(8)의 연산 결과를 입력 수단으로 하고, 출력 프레임 메모리(14)로 데이터를 공급한다.

출력 1 SAM 부(9)는, 출력 화상 데이터의 화소의 짝수번째의 데이터를 출력 프레임 메모리(14)에 공급하여, 출력 2 SAM 부(10)는, 출력 데이터의 화소의 홀수번째의 데이터를 출력 프레임 메모리(14)에 공급한다. 즉, 출력 화상 데이터의 화소를 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10....이라 번호를 붙이면, 출력 1 SAM 부(9)는, 0, 2, 4, 6, 8, 10....을 출력 프레임 메모리(14)에 공급하고, 출력 2 SAM 부(10)는, 1, 3, 5, 7, 9....을 출력 프레임 메모리(14)에 공급하게 된다.

프로그램 제어부(12)

프로그램 제어부(12)는, SIMD 형 화상 처리 프로세서(3)의 내부에 존재하고 있어, 하드웨어로 구성되어 있어도 좋고, 소프트웨어로 실현하는 경우에는, 제어 코드 파일로서 구성되어 있어도 좋다. 프로그램 제어부(12)는 제어 코드에 따라서, 입력 SAM 부(5, 6), 데이터 메모리부(7), ALU 어레이부(8), 출력 SAM 부(9, 10)를 제어한다.

요소 프로세서군(13)

요소 프로세서군(13)은, 동일한 화소에 대응하는 입력 SAM 부(5, 6) 데이터 메모리부(7), ALU 어레이부(8), 출력 SAM 부(9, 10)를 한개의 요소 프로세서라고 생각하였을 때의 집합의 것이다. 요소 프로세서의 수가 1개의 병렬 프로세서의 처리할 수 있는 화상의 크기를 결정한다. 본 발명에 있어서는, 입출력 2상으로 되어 있기 때문에 요소 프로세서의 수의 2배까지의 화상 처리에 적응한다.

또한, SIMD 형의 경우, 프로그램 제어 코드도 요소 프로세서 각각이, 중심이라고 생각한 코드가 생성되고, 각각은 일제히 동일한 처리로 행해진다.

출력 프레임 메모리(14)

출력 프레임 메모리(14)는, 하드웨어의 메모리로서 구성되어 있어도 좋고, 소프트웨어의 파일로서 구성되어 있어도 좋다.

출력 프레임 메모리(14)는, 출력 SAM 부(9, 10)로부터의 데이터를 입력으로서, 출력 화상 데이터를 생성하여 공급한다. 여기서는, 상술한 바와 같이 출력 1 SAM 부(9)가 짝수번째의 데이터를, 출력 2 SAM 부(10)가 홀수번째의 데이터를 공급하기 때문에, 그것들을 합성하여 1열의 출력 화상 데이터를 생성한다.

상술한 일 실시 형태에 있어서, 입출력을 2상(도 1과 같이 입력 1 SAM 부(5), 입력 2 SAM 부(6), 출력 1 SAM 부(9), 출력 2 SAM 부(10)를 구비하는 병렬 프로세서 구조. 이후 필요가 없는 한, 이것을 2상으로 하는 단어로 설명한다)로 한 경우의 실제의 제어 코드 생성 알고리즘을 도 4 내지 도 14의 흐름도에 도시한다.

도 4 내지 도 6에 도시한 흐름도는, 도 1의 화상 처리 프로세서의 입력 SAM 부(5, 6), 출력 SAM 부(9, 10)가 2상이 된 경우의 확대 화소수 변환으로 필요하게 되는 입력 포인터의 설정 알고리즘이다. 도 1의 프로그램 제어부(12)로 연산되어 설정된다. 도 4 내지 도 6는, 일련의 처리를 나타내는 것이지만, 작도 스페이스의 제약상, 3개의 도면으로 분할되어 있다.

도 7 내지 도 9에 도시한 흐름도는, 도 1의 화상 처리 프로세서의 입력 SAM 부(5, 6), 출력 SAM 부(9, 10)가 2상이 된 경우의 축소 화소수 변환으로 필요하게 되는 출력 포인터의 설정 알고리즘이다. 도 1의 프로그램 제어부(12)로 연산되어 설정된다. 도 7 내지 도 9는, 일련의 처리를 나타내는 것이지만, 작도 스페이스의 제약상, 3개의 도면으로 분할되어 있다.

도 10 내지 도 12는, 도 4 내지 도 6의 알고리즘으로 입력 포인터를 설정하였을 때, 근방 4화소의 특정 알고리즘이다. (4탭의 필터의 경우에는, 근방 4화소를 사용하여 필터 연산을 한다.) 도 10 내지 도 12는, 일련의 처리를 나타내는 것이지만, 작도 스페이스의 제약상, 3개의 도면으로 분할되어 있다.

도 13 내지 도 14는, 도 7 내지 도 9의 알고리즘으로 출력 포인터를 설정하였을 때, 근방 4화소의 특정 알고리즘이다. (4탭의 필터의 경우에는, 근방 4화소를 사용하여 필터 연산을 한다.) 도 13 내지 도 14는, 일련의 처리를 나타내는 것이지만, 작도 스페이스의 제약상, 3개의 도면으로 분할되어 있다.

최초로, 도 4 내지 도 6를 참조하여, 입출력이 2상 구조로 확대 처리를 할 때의 IRSKIP(입력 포인터)의 설정 알고리즘에 관해서 설명한다. 이 알고리즘은, 프로그램 제어 코드 생성 프로그램 으로서 실현되고, 소프트웨어로서 구성되어 있어도 좋고), 하드웨어로서 실현되어 있어도 좋다.

또한, 도 4 내지 도 9에 있어서, phase 1, flag 1은, 데이터 메모리부(7)에 존재하여 요소 프로세서의 출력 1 SAM 부(9)의 연산을 위한 변수이고, phase 2, flag 2 또한 데이터 메모리부(7)에 존재하고 요소 프로세서의 출력 2 SAM 부(10)의 연산을 위한 변수이다. [데이터 메모리부(7)는 1개의 메모리 구조이지만, 설명의 편의상, 출력 1 SAM 부(9)에 영향을 미치는 변수를 상측, 출력 2 SAM 부(10)에 영향을 미치는 변수를 하측이라 표현한다.] 도 4의 단계(32)에서는, 위상의 값(MAG)의 설정을 하고 있다. 일 실시 형태에서는, 위상 분할수를 256으로 하고있기 때문에, 확대율을 (1배 내지 2배)까지로 하면, 128<MAG<256이 된다. 여기서, 256/(MAG)는 확대율이다.

단계(33)에서는, 상측의 데이터 플래그의 초기치를 설정하고 있다. 이 데이터 플래그는, 위상 값의 기준이 되는 화소가 입력 1 SAM 부(5)에 있는가, 입력 2 SAM 부(6)에 있는가를 판별하는 플래그로서 사용된다.

SIMD의 처리에서는, 요소 프로세서군이 모두 동일한 동작을 행하는 것으로, 단계(34)로부터 단계(44)까지의 처리가 필요하게 된다. 또한 이 처리는, 데이터 메모리부(7)와 ALU 어레이부(8)를 사용하여 행하여져, 변수명은 데이터 메모리부(7)의 어드레스를 나타내고 있는 것과 같다.

단계(34, 35, 36)는,

$255<L1: \text{phase } 1 + 2*MAG \leq 512$

인 때, 하나 왼쪽의 flag 1의 값을 반전하여 flag 1에 대입하는 조작을 한다.

L1:phase 1은, 하나 왼쪽의 요소 프로세서의 phase 1의 값을 나타내고 있다.

또한, 단계(36)의 flag 1←￣L1:flag 1의 "￣"은, 반전을 의미하고 있고, "L1"은, 1개 왼쪽의 화소, "R1"은 1개 오른쪽의 화소라는 의미를 나타내고 있다. 이하 같다.

단계(37, 38, 40)에서는, 자기 자신의 새로운 phase 1의 계산을 한다.

L1:phase 1 + 2*MAG≤512인 때,

phase 1= L1:phase 1 + 2*MAG(단계 40)

L1:phase 1 + 2**MAG>512인 때,

phase 1= L1:phase 1+ 2*MAG-512(단계 38)

다음에, 도 5에 도시된 단계(44)에서 이곳까지의 조작이 전체 화소회 종료했는지 여부를 판정한다. 전체 화소회(요소 프로세서의 수) 종료하지 않고 있으면, 도 4의 단계(34)로 되돌아가고, 종료하고 있으면, 단계(45)로 옮긴다.

SIMD 형에서는, 1회의 조작으로 1화소의 phase 1을 확정할 수 있기 때문에, 이 조작이 필요하게 된다. 이것에서 왼쪽의 화소로부터 차례차례로 phase 1이 확정하여 간다.

단계(45, 46)는, 플래그와 phase 1의 초기치가 어긋나고 있는 것을 조정하기 위한 처리이다. (화상의 좌단을 기준으로 하기 위한 조작에 해당한다.)

단계(47)에서는, 2상의 하측의 phase 2를 구하기 위해서 상측의 phase 1을 기준치로서 대입하고 있다. phase 2는, phase 1으로부터 MAG만큼 어긋난 위상이 된다.

단계(48)는, 2상의 하측의 보간 연산의 중심이 되는 화소가 입력의 상측인가 하측인가를 나타내는 플래그인 flag 2를 초기화한다. flag 2는, flag 1의 값을 기준으로서, 계산된다.

단계(49, 50, 51)는,

255<phase 2≤512인 때에, flag 2를 반전한 것을 flag 2에 대입하고, 그 이외의 경우는 flag 2의 값 그대로 한다. 이것에 의해, 2상의 하측의 상의 중심 화소를 판정하는 플래그 flag 2의 설정이 종료한다.

그리고, 도 6에 도시한 단계(54, 55, 56, 57)는, phase 2의 값을 0<phase 2<256로 하도록 조정한다.

즉, 255<phase 2≤512인 때에는,

phase 2= phase 2-256(단계 56)

512< phase 2인 때에는,

phase 2= phase 2-512(단계 57)

로 한다.

단계(58, 59, 60, 61)는, 확대의 집합에 사용하는 입력 포인터(1 비트)의 설정 방법이다. 여기서는, 입력 포인터를 IR SKIP이라 표현하고 있다. IRSKIP이 0인 때에는 도 2(a)의 노멀 모드가 되고, 이것이 1인 때에는 도 2(b)의 스킵이 된다.

단계(58)에서는, 1개 오른쪽의 flag 1을 w에 대입한다. 단계(59)에서는, flag 1과 w와의 앤드를 취하고, 그 결과의 반전을 w에 대입한다.

단계(60)에서는, 단계(59)의 결과의 w와 flag 1의 앤드를 취하고, 그것을 IRSKIP에 대입한다. 이 결과, flag 1과 IRSKIP과의 관계는,

flag 1: 001101001100110

IRSKIP: 000100000100010

ORSKIP: 000000000000000

가 된다. 입력 1 SAM 부(5), 입력 2 SAM 부(6)에 IRSKIP에 따른 스킵 모드가 설정되어, 입력 데이터가 광범위하게 입력되고, 보간 연산을 함으로써 확대 처리를 할 수 있다.

단계(61)에서, ORSKIP(출력 포인터)에 0이 대입됨으로써, 확대 처리의 경우에서는, 항상 도 2(a)의 노멀 모드가 세트된다.

다음에, 도 7 내지 도 9의 흐름도를 참조하여, 입출력이 2상 구조로 축소 처리를 할 때의 ORSKIP(출력 포인터)의 설정 알고리즘에 관해서 설명한다.

도 7 중의 단계(72, 73)에 있어서, 위상과, 플래그의 초기치를 설정한다.

단계(74, 75, 76)에서는, 새로운 phase 1의 계산을 한다

L1:phase 1≤512인 때,

phase 1= L1:phase 1+ 2*MAG-512(단계 75)

그 이외인 때,

phase 1= L1:phase 1-512(단계 76)

단계(77, 78, 79)에서는, 구한 phase 1에 대응한 flag 1을 결정한다.

phase 1> 255인 때,

flag 1= 1(단계 78)

그 이외인 때,

flag 1= 0(단계 79)

이다.

축소 처리에서는, flag 1의 값이 0인 때에는 각 요소 프로세서의 메모리부가 입력 데이터의 상측의 상을 보간 중심 데이터로서 사용하고, flag 1의 값이 1인 때에는, 하측의 상을 보간 중심 데이터로 하고 있거나, 혹은, 스킵되는 것을 나타내고 있다.

단계(80)에서는, 여기까지의 조작이 전체 화소회(요소 프로세서의 수) 종료했는지 여부를 판정하고 있다. 전체 화소회 종료하지 않고 있으면, 단계(74)로 되돌아간다. 종료하고 있으면, 단계(81) 이후로 이동한다.
도 8의 단계(83, 84)는, 도 4 및 도 5와 같이 제일 왼쪽의 화소를 중심으로 하기 위해서, flag 1의 값과, phase 1의 값을 각각 1로 설정한다.

단계(85)에서는, phase 2의 값을 phase 1으로부터 구한다. 즉,

phase 2= phase 1+ MAG

에 의해서 계산된다.

단계(86, 87, 88)는, 단계(85)에서 계산된 phase 2의 값을 필요한 범위내의 수로 조정한다. 즉,

phase 2≤512인 때,

phase 2= phase 2 + 2*MAG-512(단계 87)

그 이외인 때,

phase 2= phase 2-512(단계 88)

가 된다.

단계(89, 90, 91)에서는, phase 2의 값에 의해서 flag 2의 값이 계산된다.

phase 2> 255인 때,

flag 2= 1(단계 90)

그 이외인 때,

flag 2= 0(단계 91)

가 된다.

다음에, 도 9 중의 단계(94, 95, 96)는, 출력 포인터(여기서는, ORSKIP이라 표현하고 있다)의 설정 방법이다.

phase 1≤512인 때에는,

ORSKIP= 0(normal mode)(단계 95)

phase 1> 512인 때에는,

ORSKIP= 1(skip mode)(단계 96)가 된다.

단계(97)에 있어서, IRSKIP(입력 포인터)에 0을 대입함으로써, 축소 처리의 경우, 도 2(a)의 노멀 모드로 설정하여 놓는다.

이와 같이, 출력 포인터(ORSKIP)를 설정하는 것으로, 출력 데이터에 영향을 미치지 않는 요소 프로세서를 만들어, 결과적으로 축소가 행하여지도록 설계할 수 있다.

다음에, 도 10 내지 도 12의 흐름도를 참조하여, 입출력 2상인 때 확대 처리를 하기 위한 근방 화소(4화소)의 인출 알고리즘에 관해서 설명한다. 입력 포인터는, 도 4 내지 도 6의 방법을 사용하여 이미 설정완료인 것을 전제로 한다. SIMD 형에서는, 모든 요소 프로세서군이 동일한 동작을 행하는 것이 기본으로 되어있기 때문에, 근방 화소의 인출 방법에서는, 도 4 내지 도 6의 입력 포인터의 설정 방법이 중요하게 된다.

또한, 도 10 내지 도 14에서 사용하고 있는 Cdata1, L1data1, R1data1, R2data1의 각각은, 출력 1 SAM 부(9)의 보간 연산을 하기 위한 근방 화소 데이터로, 이것들의 4개의 데이터와 필터 계수로 곱의 합 연산을 함으로써, 출력 1 SAM 부(9)의 결과가 얻어진다.

Cdata2, L1data2, R1data2, R2data2는, 출력 2 SAM 부(10)의 보간 연산을 하기 위한 근방 화소 데이터로, 4개의 데이터와 필터 계수로 곱의 합 연산을 함으로써, 출력 2 SAM 부(10)의 결과가 얻어진다.

도 10은, 상측의 상의 근방 화소의 인출, 도 11 및 도 12는, 하측의 상의 근방 화소의 인출 방법이다. 또한, IR1data는, 상측의 상인 입력 1 SAM 부(5)로부터의 입력 데이터, IR2data는, 하측의 상인 입력 2 SAM 부(6)로부터의 입력 데이터를 나타내고 있다.

단계(102)에서는, flag 1의 값에 의해서 2 가지로 처리가 분리된다. flag 1이 0인 때는, 입력 데이터의 상측의 상에 중심 데이터가 있고, flag 1이 1인 때에는 입력 데이터의 하측의 상에 중심 데이터가 있다.

(flag 1= 0)인 때에서는, 단계(103)에서 중심 데이터(Cdata1)가 인출된다.

단계(104, 105, 106)에서는, 1개 왼쪽의 입력 포인터의 상태를 보아 1개 왼쪽의 데이터(L1data1)를 넣는다.

단계(107)에서는, 1개 오른쪽의 근방 화소 데이터(R1data1)를 넣는다. 중심이 자기의 상측의 상의 데이터이기 때문에, 1개 오른쪽의 근방 화소 데이터는 자기의 하측의 상의 데이터가 된다.

단계(108, 109, 110)에서는, 1개 오른쪽의 입력 포인터의 상태를 보아, 2개 오른쪽의 근방 화소 데이터(R2data1)를 넣는다.

(flag 1= 1)인 때에서는, 단계(111)에 있어서, 중심의 화소 데이터(Cdata1)를 넣는다. (flag 1= 1)의 경우에는, 하나 왼쪽(L1:)의 하측이 중심이 되도록 입력 포인터를 설정하고 있다.

단계(112)에서는, 1개 왼쪽의 근방 화소 데이터(L1data1)를 넣는다.

단계(113, 114, 115, 116, 117)에서는, 자기의 입력 포인터의 상태를 보면서, 1개 오른쪽의 근방 화소 데이터(R1datal)와, 2개 오른쪽의 근방 화소 데이터(R2data1)를 넣는다.

도 11에 나타내는 단계(120) 이후의 처리는, 하측의 상의 근방 화소 데이터의 인출 순서를 나타내고 있다.

단계(120)에서는, flag 2의 값으로 2개의 처리로 나누어진다. (flag 2= 0)인 때에, 단계(121)에 있어서, 또한 자기의 입력 포인터 IRSKIP의 상태에 따라서 근방 화소의 인출이 2개의 처리로 분리된다.

단계(122)로부터 단계(129)까지는, 자기의 입력 포인터가 1이 아닌 경우의 처리이다. 단계(122)에서는, 하측의 상의 중심 데이터(Cdata2)의 인출을 하고, 단계(123)에서는, 1개 오른쪽의 근방 화소 데이터(R1data2)의 인출을 한다.

단계(124, 125, 126)에서는, 1개 왼쪽의 입력 포인터의 상태를 보아, 1개 왼쪽의 근방 화소 데이터(L1data2)를 넣는다.

단계(127, 128, 129)에서는, 1개 오른쪽의 입력 포인터의 상태를 보아, 2개 오른쪽의 근방 화소 데이터(R2data)를 넣는다.

단계(130)로부터 단계(135)까지는, (flag 2= 0)이고, 자기의 입력 포인터가 1(스킵 모드)인 때의 근방 화소 데이터의 인출 순서이다.

단계(130)에서는, 중심 화소 데이터(Cdata2)의 인출을 하고, 단계(131)에서는, 1개 오른쪽의 근방 화소 데이터(R1data2)의 인출을, 단계(132)에서는, 1개 왼쪽의 근방 화소 데이터(L1data2)의 인출을 한다.

단계(133, 134, 135)는, 2개 오른쪽의 입력 포인터의 상태를 보아 2개 오른쪽의 근방 화소 데이터(R2data2)의 인출을 한다.

도 12에 도시한 단계(136) 이후는, 단계(120)에서 flag 2가 1로 결정된 경우의 근방 화소의 인출 순서를 나타내고 있다. 이것은, 단계(136)에서의 자기의 입력 포인터 IRSKIP의 상태에서 2개의 처리로 분리된다.

단계(137)로부터 단계(143)까지는, 자기의 입력 포인터가 도 2(a)에 도시한 노멀 모드의 경우이다.

단계(137)에서는, 중심 데이터(Cdata2)의 인출을 하고, 단계(138)로서는, 1개 왼쪽의 근방 화소 데이터(L1data2)의 인출을 한다.

단계(139, 140, 141, 142, 143)에서는, 1개 오른쪽의 입력 포인터의 상태를 보아 1개 오른쪽의 근방 화소 데이터(R1data2), 2개 오른쪽의 근방 화소 데이터(R2data2)의 인출을 한다.

단계(144, 145, 146, 147)에서는, 자기의 입력 포인터의 상태가 도 2(b)의 스킵 모드의 경우의 근방 화소 데이터의 인출을 행하는 처리이다.

다음에, 도 13 내지 도 14의 흐름도를 참조하여, 입출력 2상인 때의 축소 처리를 하기 위한 근방 화소의 인출 순서에 관해서 설명한다. 이 때의 flag 1, flag 2의 설정과, 출력 포인터의 설정은 도 7 내지 도 9의 알고리즘에 따라서 구하는 것을 전제로 하고 있다. 또한, 여기서 사용되고 있는 변수 등은 도 10 내지 도 12의 경우와 같다.

도 13은 상측의 상의 근방 화소 데이터의 인출 순서이고, 도 14는 하측의 상의 근방 화소 데이터의 인출 순서이다.

도 13의 단계(152)에서는, flag 1의 값에 의해서, 인출 방법이 2개의 처리로 분리된다.

단계(153, 154, 155, 156)는, flag 1의 값이 0의 경우의 근방 화소 데이터의 인출 순서를 나타내고 있다.

단계(157, 158, 159, 160)는, flag 1의 값이 1의 경우의 근방 화소 데이터의 인출 순서를 나타내고 있다.

도 14에 도시한 단계(163) 이후는 하측의 상의 근방 화소 데이터의 인출 순서를 나타내고 있다.

단계(165, 166, 167, 168)은, 단계(163)과 단계(164)에서, flag 2와 flag 1이 함께 1로 결정된 경우의 근방 화소 데이터의 인출 순서를 나타내고 있다.

단계(169, 170, 171, 172)는, 단계(163)와 단계(164)에서, (flag 2=1, flag 1= 0)로 결정된 경우의 근방 화소 데이터의 인출 순서를 나타내고 있다.

단계(173, 174, 175, 176)는, 단계(163)에서, flag 2가 0으로 결정된 경우의 근방 화소 데이터의 인출 순서를 나타내고 있다.

상술한 본 발명의 일 실시 형태에 관해서, 보다 구체적으로 설명한다. 일례로서, 도 15a 에 도시한 입력 데이터로부터 도 15b에 도시한 출력 데이터를 작성하는 것으로 한다. 횡 일렬로 나열한 데이터로 생각하면, 입력 데이터가 22화소이고, 출력 데이터가 30화소이므로, 3개의 입력 데이터의 화소로부터 4개의 출력 데이터의 화소를 작성하는 것과 동등하고, 3:4의 보간 연산을 하게 된다.

이 예에서는, 도 4의 단계(32)의 위상치 설정에 있어서, MAG의 값은, 데이터 사이를 256 분할하였다고 생각하면,

MAG= 256 * 3/4= 192이 된다.

단계(33)(플래그의 초기치 설정)에서는, 요소 프로세서(도면에서는 PE로 나타낸다)의 flag 1과 phase 1의 메모리의 값을 제로로 초기화한다. 즉, 이 상태에서는, 전체 요소 프로세서의 메모리의 값이 제로로 되어 있다.

그리고, 단계(34, 35, 36)에 의해서, flag 1은, 전체 요소 프로세서 1로 된다. 단계(37, 40)에 의해서, phase 1의 값은, 모두 384가 된다.

단계(34) 이후의 처리를 다시 1회 되풀이하면, 가장 좌측의 요소 프로세서는, 전의 조작과 동일한 값이 되지만, 2번째 이후의 데이터는, flag 1은, 1인 그대로, phase 1은 256이 된다. 또한, 가장 좌측의 좌측(요소 프로세서가 없는 부분)은, 항상 제로인 것이 전제이다.

다시 한번 되풀이하면, 요소 프로세서의 순서로 flag 1은, 110000....으로 되고, phase 1은, 384 256 128 128 128....로 된다. 이와 같이, 1회의 조작으로, 왼쪽으로부터 순서로 1개씩의 요소 프로세서의 flag 1, phase 1이 결정하여 간다. 이 모양을 도 16에 도시한다.

다음에, 위상 원점을 제일 왼쪽의 요소 프로세서에 합치기 위해서, 도 5의 단계(45, 46)의 조작을 한다. 단계(48, 49, 50, 51)는, phase 1 + MAG의 값에 따라서 flag 2의 값을 결정하여 가는 처리이다.

if(255< phase1 + MAG<512) f1ag2는 flagl의 반전을 대입

else flag2는 flag1을 대입

단계(47) 및 도 6의 단계(54, 55, 56, 57)는, 하측의 위상의 값(phase 2)을 상측의 위상의 값(phase 1)으로부터 산출한다.

if(255< phase1 + MAG<512) phase2=phase1 + MAG - 256

else if(255> phasel) phase2=phase1 + MAG

else phase2=phasel + MAG - 512

결정한 flag 1, flag 2, phase 1, phase 2로부터 입력 포인터 IRSKIP를 결정하여 간다.

단계(58, 59, 60)으로부터

IRSKIP={(f1agl&Rl : flag1)의 반전)& f1ag1

확대 처리의 경우, 출력 포인터(ORSKIP)는 제로를 세트한다. 요소 프로세서의 개수와 동일한 회수, 처리를 되풀이함으로써, 도 17에 도시한 최종 결과가 얻어진다.

도 17에 있어서, 출력 데이터(1', 2', 3', ...) 등은, 입력 데이터를 사용하여 보간 연산을 하지만, 그 보간 연산에 사용하는 화소 데이터의 선택 방법을 도 10 내지 도 14에 도시하고 있다. 우선, 출력 데이터의 상측의 데이터에 대한 보간 연산을 하기 위한 근방 화소를 선택하는 방법이 도 10에 도시되고 있다.

보간 연산을 위한 중심이 되는 데이터는, flag 1이 0인 경우, 자기의 요소 프로세서의 상측에 있고, flag 1이 1인 경우에는, 하나 왼쪽의 하측에 있는 관계로 되어있는 것을 알았다.

요소 프로세서 n 번째의 출력 데이터는, 상측의 출력은, 2 n-1이 되고, 하측의 출력이 2n + 1이 되고, 그 요소 프로세서의 입력 데이터의 상측의 데이터를 i(n)로 하고, 그 하측의 데이터를 j(n)으로 하면,

if(f1ag(n)==1) Cdata(보간 연산의 중심 데이터)= i(n)

else Cdata(보간 연산의 중심 데이터)=j(n-1)

의 관계가 성립한다.

그 밖의 3점은, 이 법칙을 확장하여, 입력 포인터의 값을 보면서 선택되어 가는 것으로 결정한다.

일례로서, 도 17 중의 출력 데이터(15')를 구하기 위해서 필요한 데이터는, 상술한 법칙을 사용하고,

Cdata=11, L1data=10, R1data=12, R2data=13

가 된다.

출력 데이터(17')를 구하는 데 필요한 데이터는,

Cdata=12, L1data=11, R1data=13, R2data=14가 된다.

또한, 도 11 및 도 12는, 하측의 출력 데이터를 구하는 데 필요한 보간 데이터의 선택 처리를 나타내고, flag 1, flag 2, 입력 포인터에 의해서 제어된다.

이상 설명한 바와 같이, 2상 구조를 갖는 병렬 프로세서와 화상 데이터를 2상으로 분해하고 또한 2상 데이터를 합성하는 프레임 메모리를 가지는 화상 처리 장치와, 2상 데이터를 화상 처리할 수 있는 알고리즘을 가지는 것으로, 화상 처리, 주로 화소수 변환을 할 때에 이하에 기술한 바와 동일한 효과가 얻어진다.

종래와 비교하여, 병렬 프로세서 1개로 2배의 화상 크기 데이터를 취급하는 것이 가능하게 되고, 종래의 반의 수의 병렬 프로세서로 실현할 수 있게 된다.

종래와 비교하여, 전체적인 회로 구성을 작게 할 수 있다.

2상 데이터의 화소수 변환 알고리즘을 확립하는 것으로, 항상 동일한 알고리즘을 사용하여 2상 데이터의 임의의 비율의 화소수 변환을 실현할 수 있다.

도면의 간단한 설명

도 1은, 본 발명의 일 실시 형태에 있어서의 SIMD 형 화상 처리 프로세서를 사용한 화상 처리 장치의 블럭도.

도 2(a, b)는, 입출력 포인터의 구조를 나타내는 회로도.

도 3(a,b,c,d)은, 입출력 SAM 부와 입출력 포인터의 관계를 나타내는 개략도.

도 4는, 본 발명에 있어서의 입출력 2상 구조로서, 확대 처리를 할 때의 입력 포인터의 설정 알고리즘을 설명하기 위한 흐름도.

도 5는, 본 발명에 있어서의 입출력 2상 구조로서, 확대 처리를 할 때의 입력 포인터의 설정 알고리즘을 설명하기 위한 흐름도.

도 6은, 본 발명에 있어서의 입출력 2상 구조로서, 확대 처리를 행할 때의 입력 포인터의 설정 알고리즘을 설명하기 위한 흐름도.

도 7은, 본 발명에 있어서의 입출력 2상 구조로서, 축소 처리를 할 때의 출력 포인터의 설정 알고리즘을 설명하기 위한 흐름도.

도 8은, 본 발명에 있어서의 입출력 2상 구조로서, 축소 처리를 할 때의 출력 포인터의 설정 알고리즘을 설명하기 위한 흐름도.

도 9는, 본 발명에 있어서의 입출력 2상 구조로서, 축소 처리할 때의 출력 포인터의 설정 알고리즘을 설명하기 위한 흐름도.

도 10은, 설정된 입력 포인터에 대하여, 화소수 변환할 때의 근방 화소(4화소)의 인출 알고리즘을 설명하기 위한 흐름도.

도 11은, 설정된 입력 포인터에 대하여, 화소수 변환할 때의 근방 화소(4화소)의 인출 알고리즘을 설명하기 위한 흐름도.

도 12는, 설정된 입력 포인터에 대하여, 화소수 변환할 때의 근방 화소(4화소)의 인출 알고리즘을 설명하기 위한 흐름도.

도 13은, 설정된 출력 포인터에 대하여, 화소수 변환할 때의 근방 화소(4화소)의 인출 알고리즘을 설명하기 위한 흐름도.

도 14는, 설정된 출력 포인터에 대하여, 화소수 변환할 때의 근방 화소(4화소)의 인출 알고리즘을 설명하기 위한 흐름도.

도 15(a,b)는, 본 발명의 일 실시 형태의 구체적 설명에 사용하는 입력 데이터 및 출력 데이터의 일례를 도시하는 개략도.

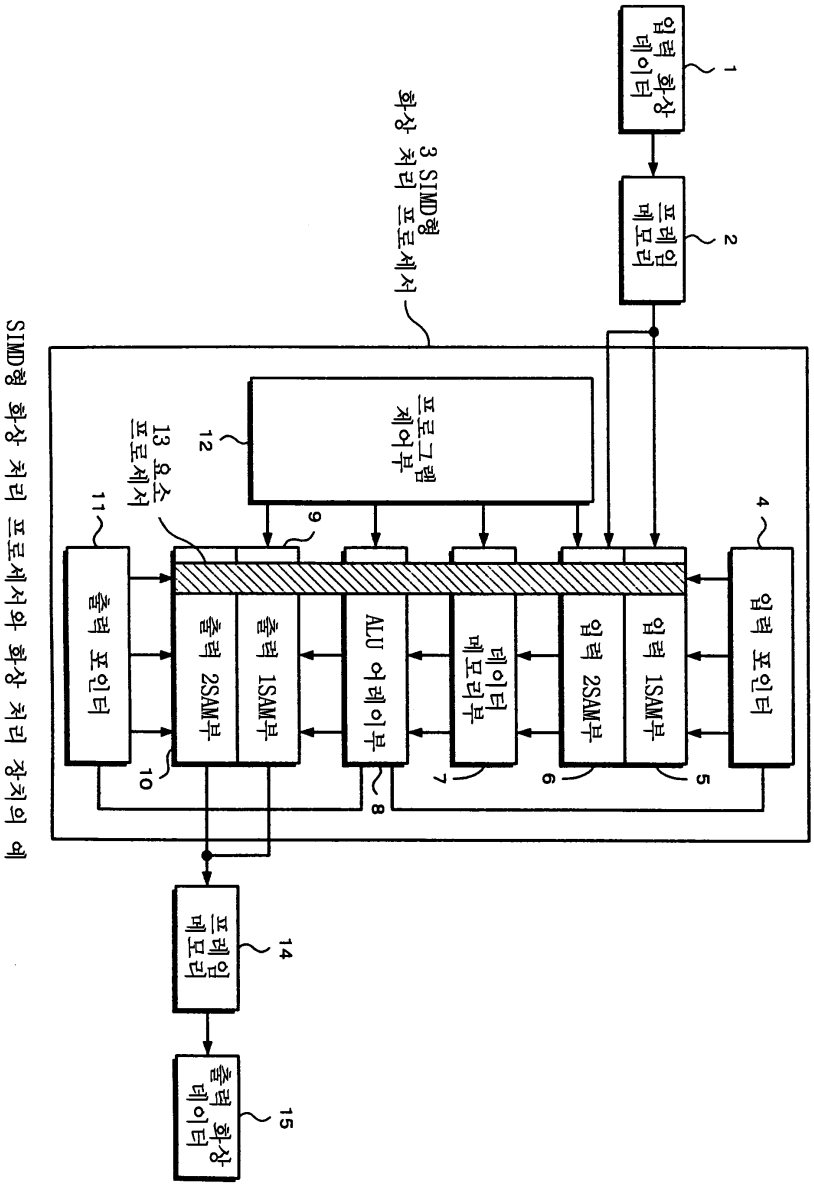
도 16은, 본 발명의 일 실시 형태의 구체적 설명에 사용하는 개략도.

도 17은, 본 발명의 일 실시 형태의 구체적 설명에 사용하는 개략도.

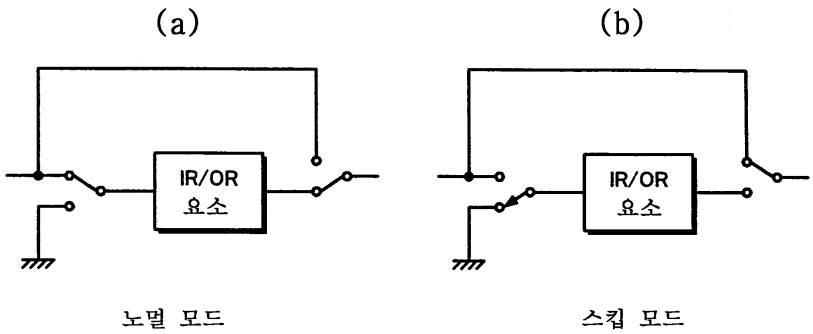
도 18은, 종래의 SIMD 형 화상 처리 프로세서를 사용한 화상 처리 장치의 일례를 나타내는 블럭도.

도면

도면1

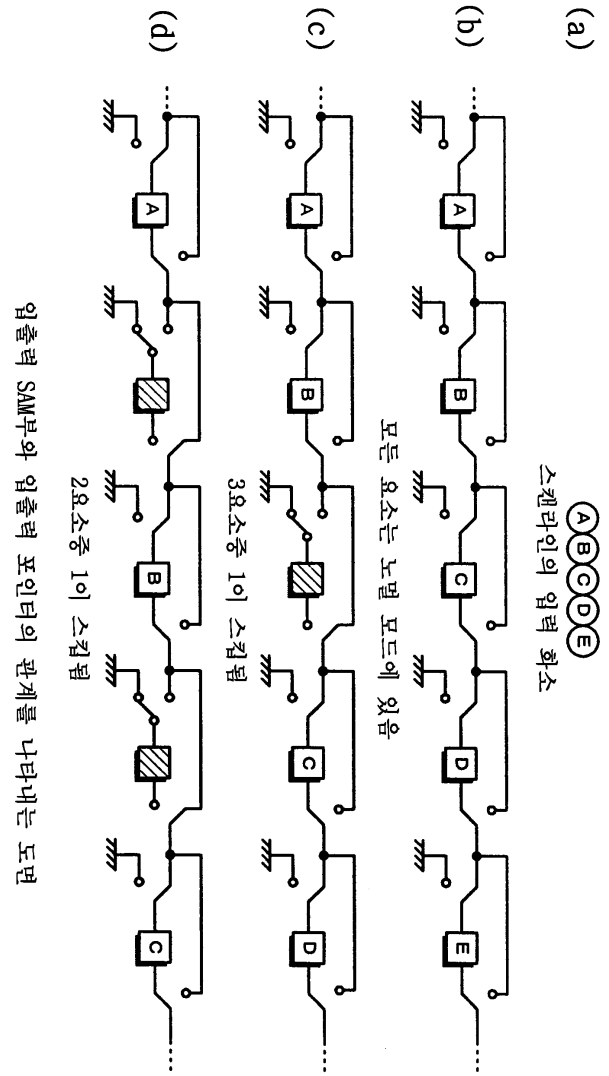


도면2



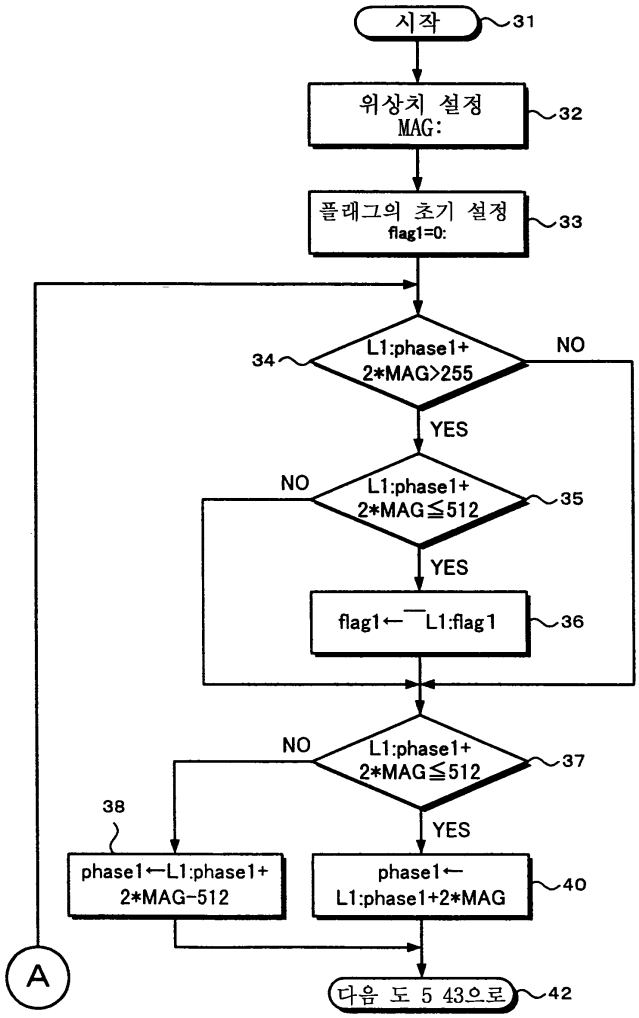
입출력 포인터의 구조

도면3



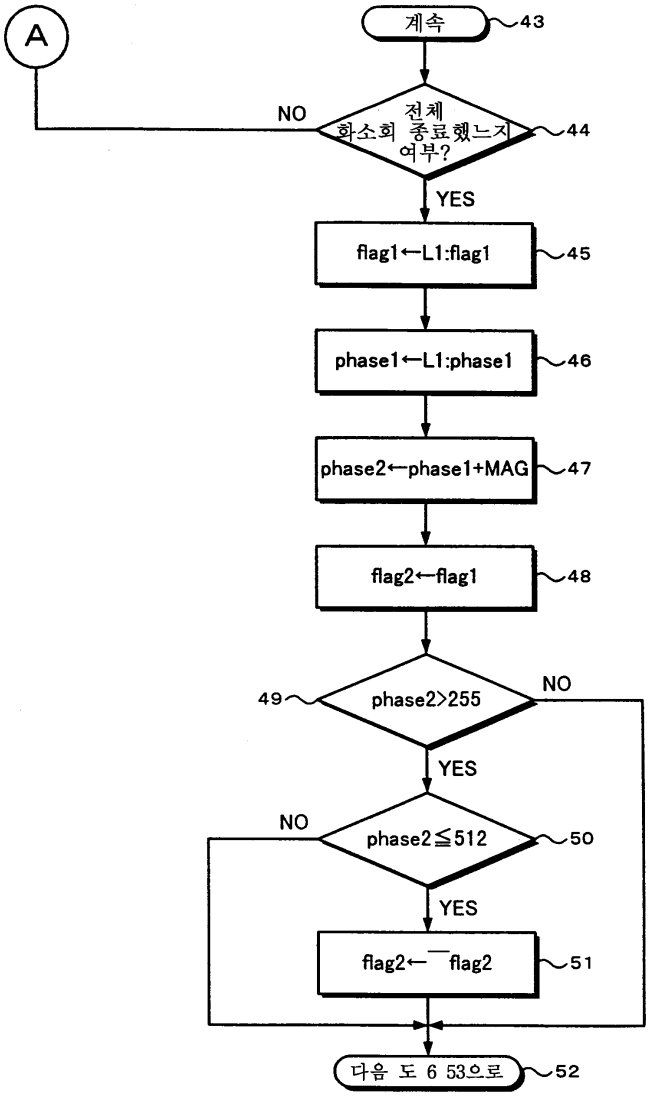
입출력 SAM부와 입출력 포인터의 관계를 나타내는 도면

도면4



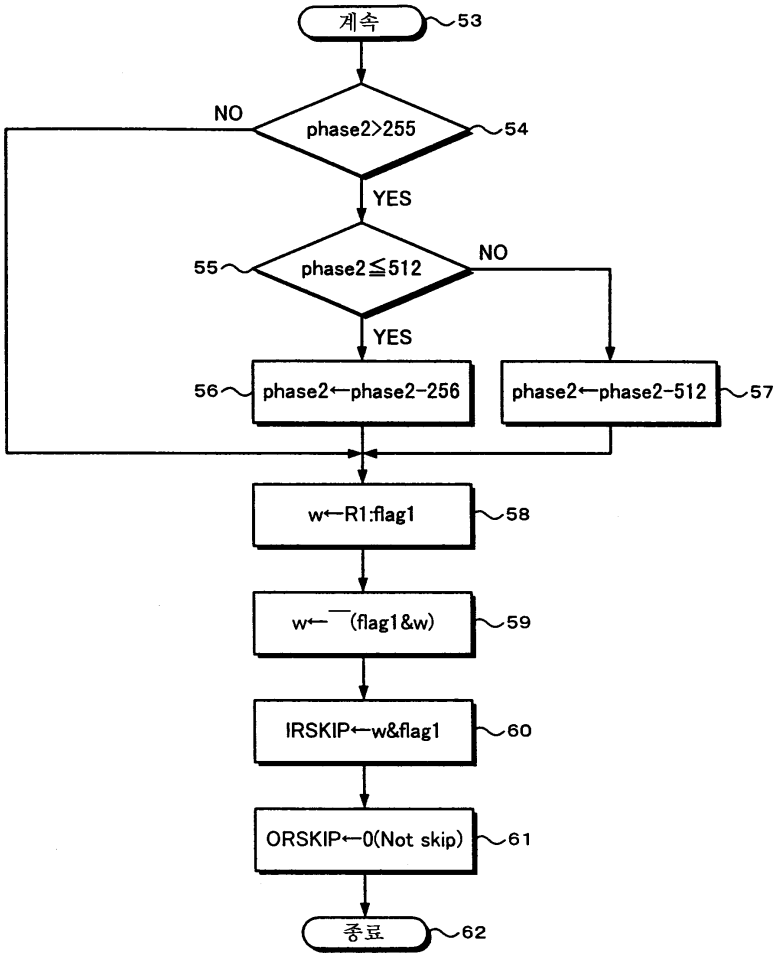
입출력이 2상 구조로 확대 처리를 행하는 때의 입력 포인터의 설정 알고리즘(그의 1)

도면5



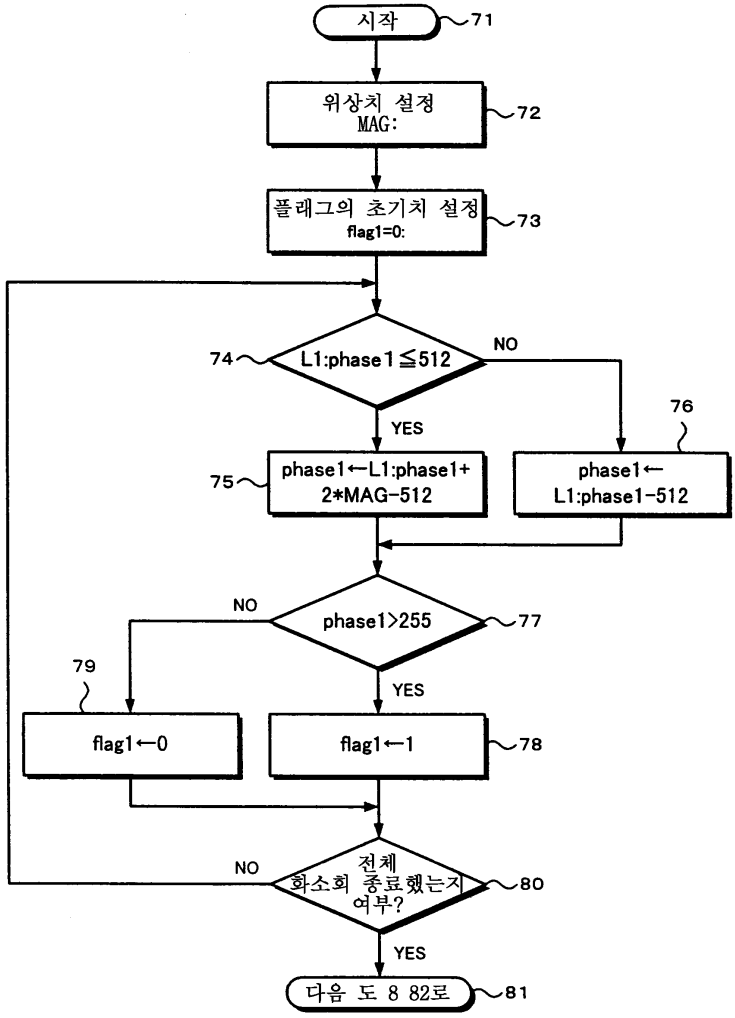
입출력이 2상 구조로 확대 처리를 행하는 때의 입력 포인터의 설정 알고리즘(그의 2)

도면6



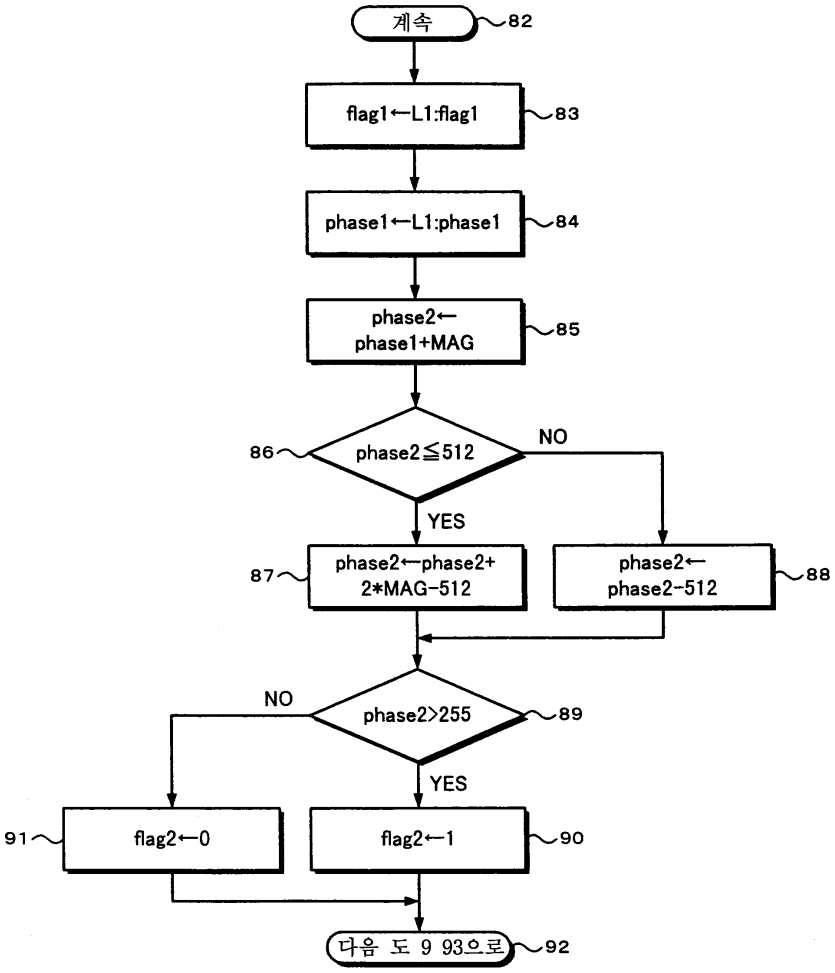
입출력이 2상 구조로 확대 처리를 행하는 때의 입력 포인터의 설정 알고리즘(그의 3)

도면7



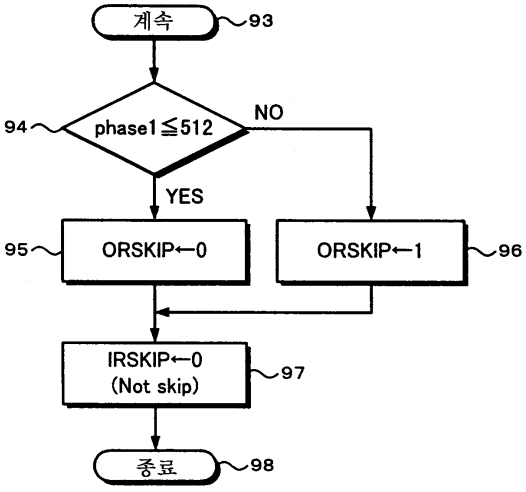
입출력이 2상 구조로 축소 처리를 행하는 때의 출력 포인터의 설정 알고리즘(그의 1)

도면8



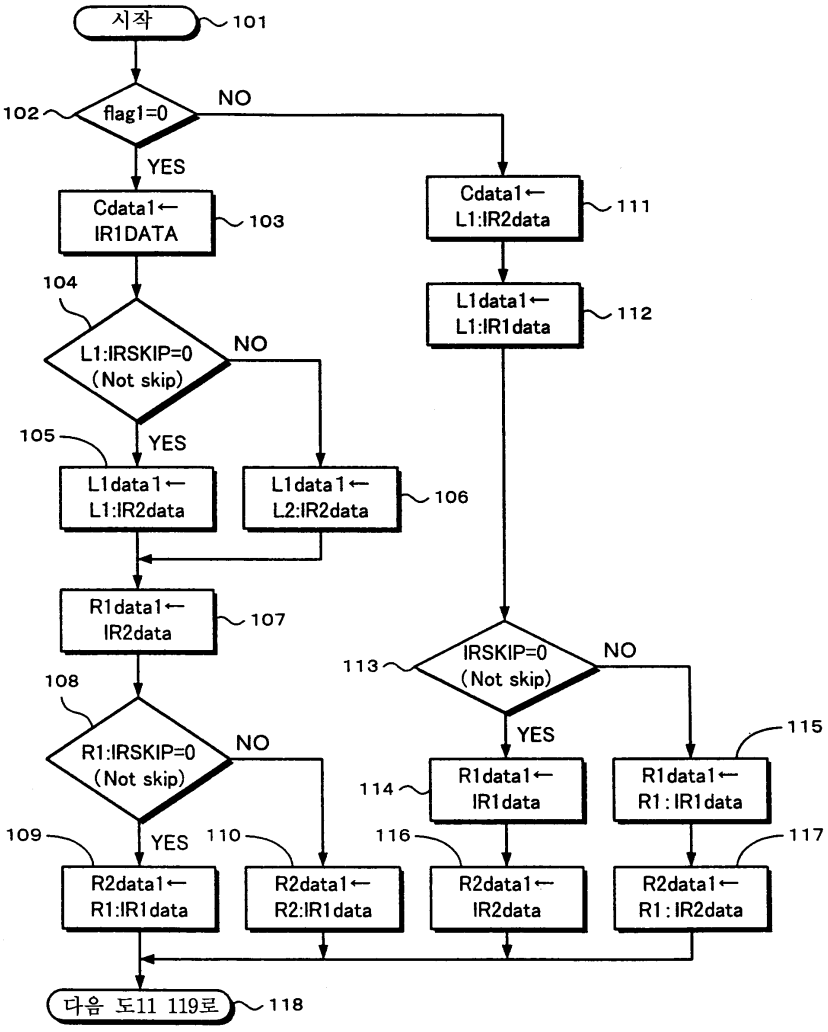
입출력이 2상 구조로 축소 처리를 행하는 때의 출력 포인터의 설정 알고리즘(그의 2)

도면9



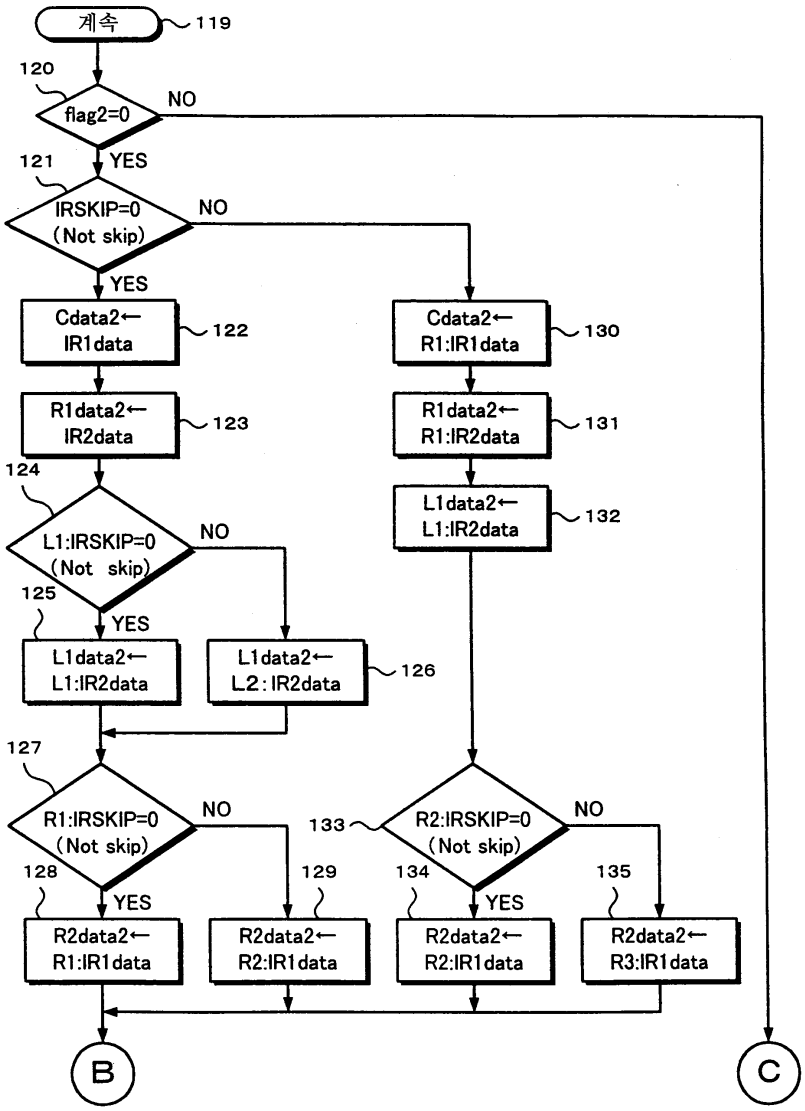
입출력이 2상 구조로 축소 처리를 행하는 때의 출력 포인터의 설정 알고리즘(그의 3)

도면10



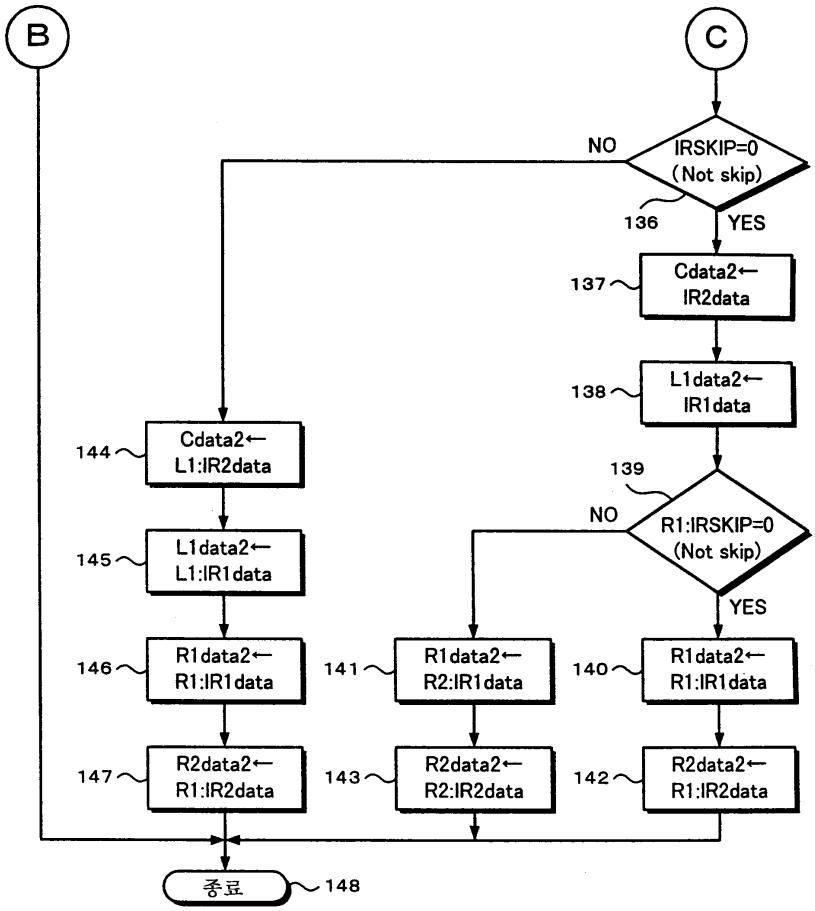
입력출 2상인 때 확대 처리를 행하기 위한 근방 화소(4 화소)의 인출 수단(그의 1)

도면11



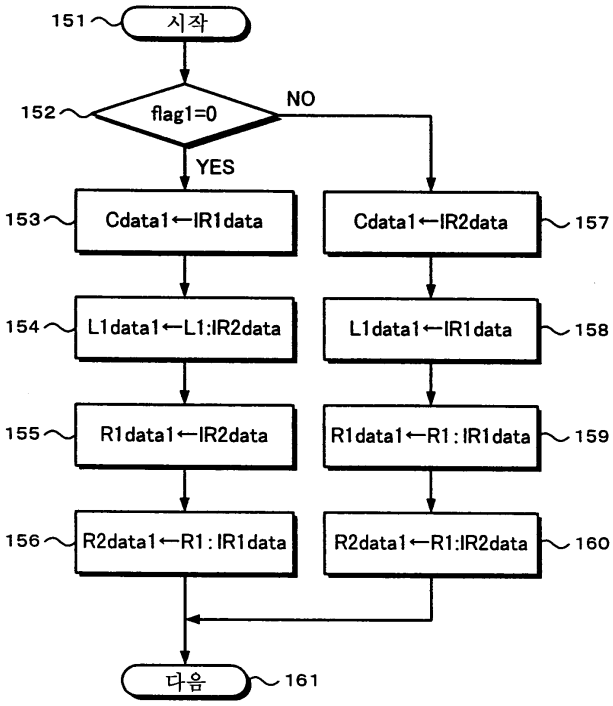
입력출 2상인 때 확대 처리를 행하기 위한 근방 화소(4 화소)의 인출 수단(그의 2)

도면12



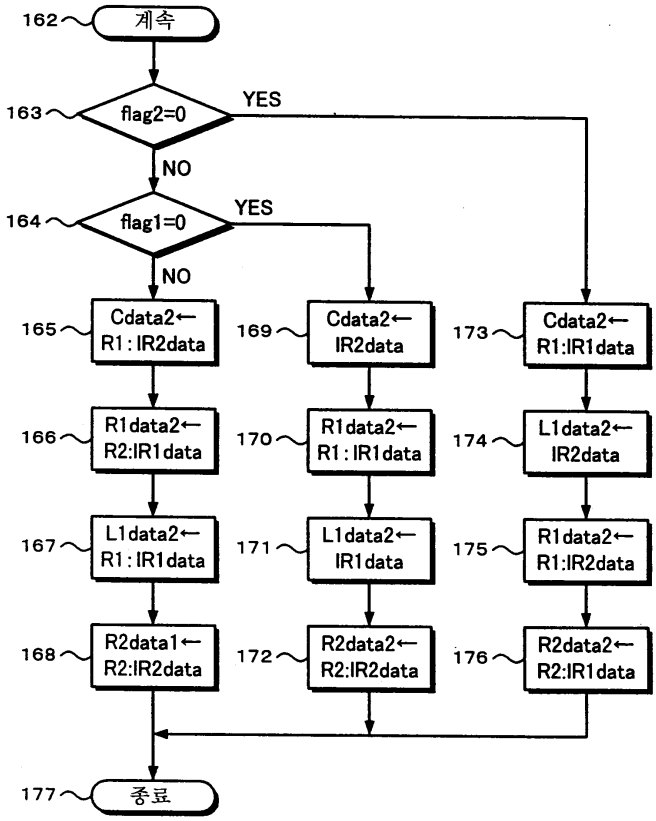
입력출 2상인 때 확대 처리를 행하기 위한 근방 화소(4 화소)의 인출 수단(그의 3)

도면13



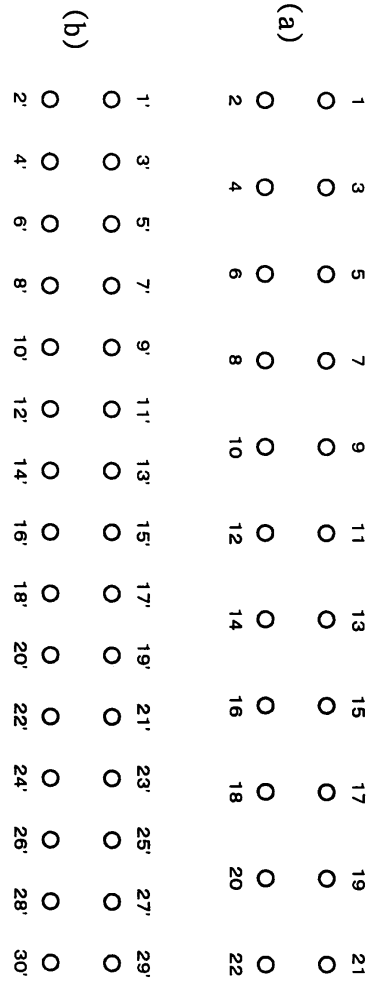
입출력 2상인 때 축소 처리를 행하기 위한 근방 화소(4 화소)의 인출 수단(그의 1)

도면14

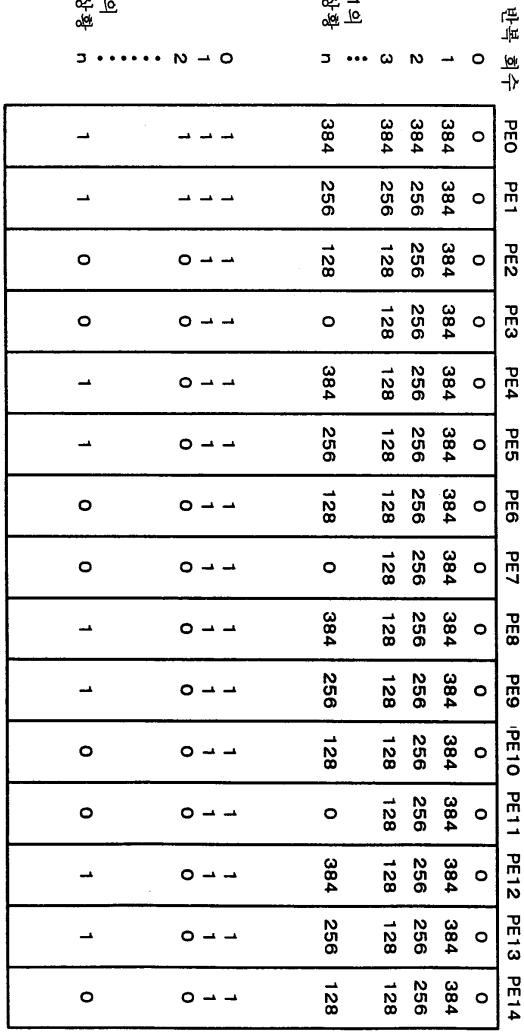


입출력 2상인 때 축소 처리를 행하기 위한 근방 화소(4 화소)의 인출 수단(그의 2)

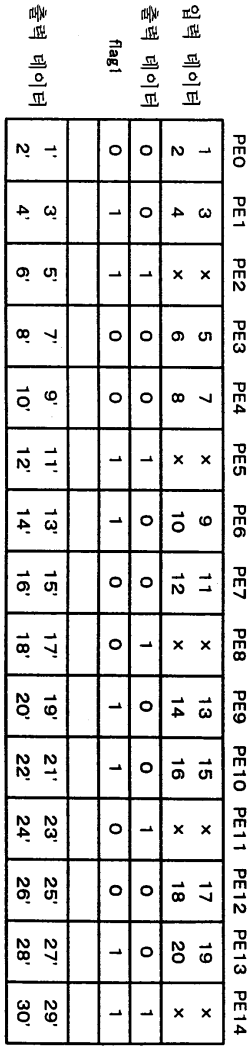
도면15



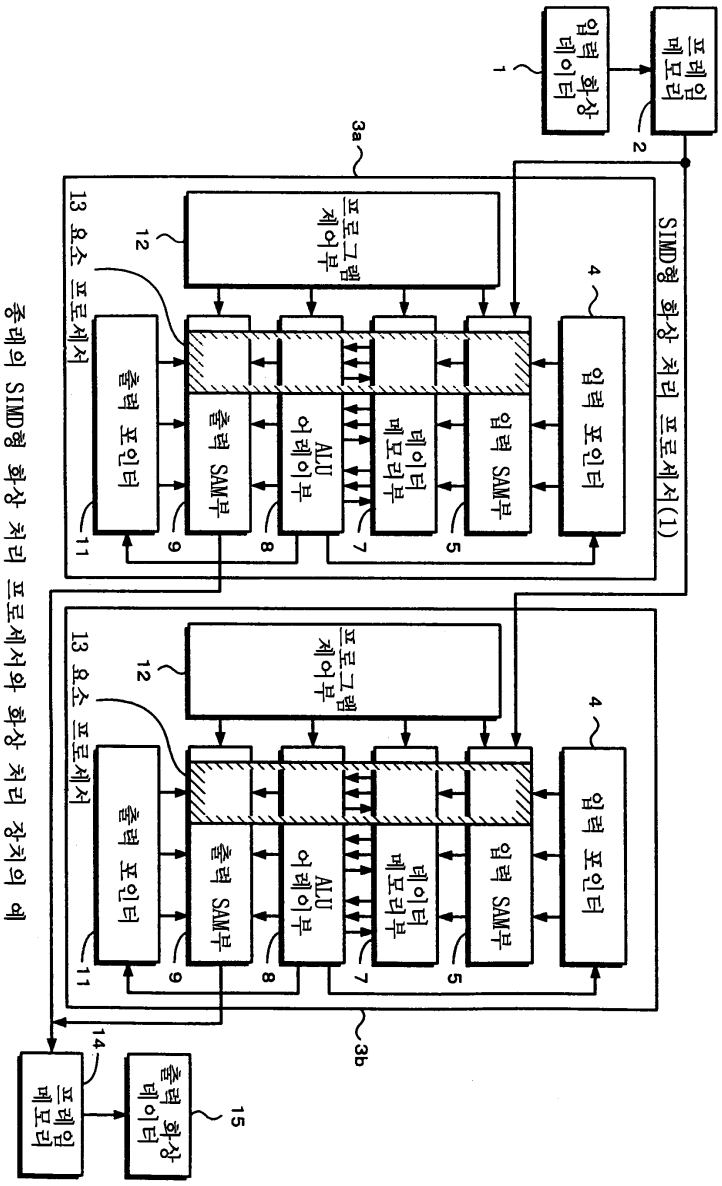
도면16


반복 회수
PE0
PE1
PE2
PE3
PE4
PE5
PE6
PE7
PE8
PE9
PE10
PE11
PE12
PE13
PE14
0
1
2
3
n
phase1의 천이 상황
flag1의 천이 상황

도면17


PE0
PE1
PE2
PE3
PE4
PE5
PE6
PE7
PE8
PE9
PE10
PE11
PE12
PE13
PE14
입력 데이터
출력 데이터
flag1
출력 데이터

도면18



종래의 SIMD형 화상 처리 프로세서와 화상 처리 장치의 예