(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2010년08월11일
(11) 등록번호 10-0975454
(24) 등록일자 2010년08월05일

(51) Int. Cl.
G06T 1/20 (2006.01)
(21) 출원번호 10-2005-7012009
(22) 출원일자(국제출원일자) 2003년12월12일
심사청구일자 2008년12월11일
(85) 번역문제출일자 2005년06월24일
(65) 공개번호 10-2005-0085899
(43) 공개일자 2005년08월29일
(86) 국제출원번호 PCT/JP2003/015976
(87) 국제공개번호 WO 2004/059576
국제공개일자 2004년07월15일
(30) 우선권주장
JP-P-2002-00374862 2002년12월25일 일본(JP)
(56) 선행기술조사문헌
JP08161271 A
JP2004080295 A
전체 청구항 수 : 총 9 항

(73) 특허권자
소니 주식회사
일본국 도쿄도 미나토쿠 코난 1-7-1
(72) 발명자
곤도, 데쯔지로
일본 141-0001 도꾜도 시나가와꾸 기따시나가와 6쪼메 7-35 소니가부시끼 가이샤 내
사또, 히로시
일본 141-0001 도꾜도 시나가와꾸 기따시나가와 6쪼메 7-35 소니가부시끼 가이샤 내
(74) 대리인
구영창, 이중희, 장수길

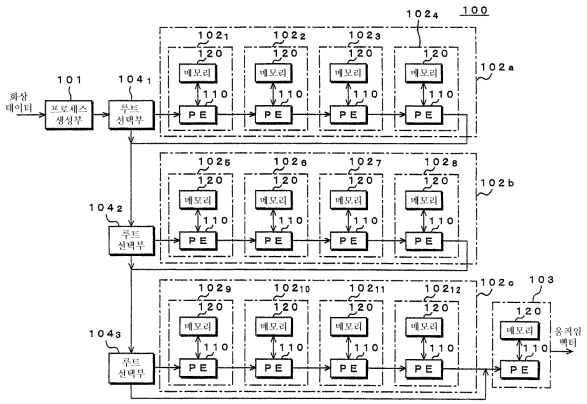
심사관 : 권성호

(54) 화상 처리 장치

(57) 요 약

본 발명은, 예를 들면 움직임 벡터의 검출 처리 등에 적용하기에 적합한 화소 처리 장치에 관한 것이다. 연산 처리 유닛(102_n)(n=1~12)은, 프로세스 생성부(101)로부터 출력되는 프로세스 패킷을 수신하고, 그 프로세스 패킷에 포함되는 명령에 따른 처리를 행한다. 유닛(102_1~102_{12})을 3개의 조(102a~102c)로 나누고, 각 조의 입력측에 루트 선택부(104_1~104_3)를 삽입한다. 루트 선택부(104_1~104_3) 각각은, 입력 프로세스 패킷에 따른 프로세스를 실행하는 유닛이, 직후에 위치하는 조(102a~102c)에 존재하지 않을 때, 상기 입력 프로세스 패킷을, 그 조의 입력측이 아니라, 그 조의 출력측에 공급한다. 프로세스 패킷은 그것에 따른 프로세스를 실행하기 위한 유닛이 존재하지 않는 조를 바이패스하고 이동하며, 따라서 처리 시간을 단축할 수 있으며, 또한 소비 전력을 저감할 수 있다.

대 표 도



특허청구의 범위

청구항 1

화상 데이터를 취급하는 화상 처리를 행하기 위한 각 프로세스를 생성하고, 그 각 프로세스를 실행하는 명령을 포함하는 프로세스 데이터를 순차적으로 출력하는 프로세스 생성 수단과,

상기 프로세스 데이터에 따라 상기 프로세스를 실행하는, 직렬 접속된 복수개의 실행 수단을 구비하며,

상기 프로세스 데이터는, 그 프로세스 데이터에 따른 프로세스를 실행하기 위한 실행 수단의 어드레스를 포함하고,

연속하는 실행 수단의 조마다, 그 조의 입력측에, 입력 프로세스 데이터를 그 조의 입력측 또는 출력측에 선택적으로 공급하는 루트 선택 수단이 삽입되며,

상기 루트 선택 수단은,

직후에 위치하는 조를 구성하는 각 실행 수단의 어드레스를 기억하는 제1 어드레스 기억 수단과,

입력 프로세스 데이터에 포함되는 실행 수단의 어드레스에 상기 제1 어드레스 기억 수단에 기억되어 있는 어드레스 중 적어도 어느 하나가 존재할 때 제1의 상태로 되는 판정 신호를 출력하는 제1 어드레스 판정 수단과,

상기 제1 어드레스 판정 수단으로부터 출력되는 판정 신호에 기초하여, 상기 판정 신호가 상기 제1의 상태일 때, 상기 입력 프로세스 데이터를 상기 직후에 위치하는 조의 입력측에 공급하고, 상기 판정 신호가 상기 제1의 상태가 아닐 때, 상기 입력 프로세스 데이터를 상기 직후에 위치하는 조의 출력측에 공급하는 제1 절환 수단

을 갖는 것을 특징으로 하는 화상 처리 장치.

청구항 2

제1항에 있어서,

상기 복수개의 실행 수단 각각은,

입력 프로세스 데이터에 포함되는 명령에 따라 화상 데이터의 처리를 행하며, 그 처리 결과에 기초하여 상기 입력 프로세스 데이터를 변경하여 출력할 프로세스 데이터를 얻음과 함께, 그 출력할 프로세스 데이터를 출력할 때 제2의 상태로 되는 요구 신호를 출력하는 처리 수단과,

자신의 어드레스를 기억하는 제2 어드레스 기억 수단과,

상기 입력 프로세스 데이터에 포함되는 실행 수단의 어드레스에 상기 제2 어드레스 기억 수단에 기억되어 있는 자신의 어드레스가 존재할 때 제1의 상태로 되는 판정 신호를 출력하는 제2 어드레스 판정 수단과,

상기 제2 어드레스 판정 수단으로부터 출력되는 판정 신호 및 상기 처리 수단으로부터 출력되는 요구 신호에 기초하여, 상기 요구 신호가 상기 제2의 상태일 때, 상기 처리 수단에 의해 얻어진 출력할 프로세스 데이터를 출력 프로세스 데이터로서 출력하고, 상기 요구 신호가 상기 제2의 상태가 아니며, 또한 상기 판정 신호가 상기 제1의 상태가 아닐 때, 상기 입력 프로세스 데이터를 출력 프로세스 데이터로서 출력하는 제2 절환 수단

을 갖는 것을 특징으로 하는 화상 처리 장치.

청구항 3

제2항에 있어서,

상기 복수개의 실행 수단 각각은, 화상 데이터를 기억하는 데이터 기억 수단을 더 가지며,

상기 처리 수단은, 상기 화상 데이터의 처리로서, 상기 데이터 기억 수단에의 화상 데이터의 기입, 그 데이터 기억 수단으로부터의 화상 데이터의 판독, 또는 화상 데이터의 차분의 연산 중 어느 하나의 처리를 행하는 것을 특징으로 하는 화상 처리 장치.

청구항 4

제2항에 있어서,

상기 제2 절환 수단은,

상기 요구 신호가 상기 제2의 상태가 아니며, 또한 상기 판정 신호가 상기 제1의 상태일 때에는, 하이 레벨 또는 로우 레벨의 데이터를 출력하는 것을 특징으로 하는 화상 처리 장치.

청구항 5

제1항에 있어서,

상기 화상 처리는, 움직임 벡터를 검출하는 처리인 것을 특징으로 하는 화상 처리 장치.

청구항 6

화상 데이터를 취급하는 화상 처리를 행하기 위한 각 프로세스를 생성하고, 그 각 프로세스를 실행하는 명령을 포함하는 프로세스 데이터를 순차적으로 출력하는 프로세스 생성 수단과,

상기 프로세스 데이터에 따라 그 프로세스를 실행하는, 직렬 접속된 복수개의 실행 수단을 구비하며,

상기 프로세스 데이터는, 상기 프로세스 데이터에 따른 프로세스를 실행하기 위한 실행 수단의 어드레스를 포함하고,

상기 복수개의 실행 수단 각각은,

입력 프로세스 데이터에 포함되는 명령에 따라 화상 데이터의 처리를 행하며, 그 처리 결과에 기초하여 그 입력 프로세스 데이터를 변경하여 출력할 프로세스 데이터를 얻음과 함께, 그 출력할 프로세스 데이터를 출력할 때 제2의 상태로 되는 요구 신호를 출력하는 처리 수단과,

자신의 어드레스를 기억하는 어드레스 기억 수단과,

상기 입력 프로세스 데이터에 포함되는 실행 수단의 어드레스에 상기 어드레스 기억 수단에 기억되어 있는 자신의 어드레스가 존재할 때 제1의 상태로 되는 판정 신호를 출력하는 어드레스 판정 수단과,

상기 어드레스 판정 수단으로부터 출력되는 판정 신호 및 상기 처리 수단으로부터 출력되는 요구 신호에 기초하여, 상기 요구 신호가 상기 제2의 상태일 때, 상기 처리 수단에 의해 얻어진 출력할 프로세스 데이터를 출력 프로세스 데이터로서 출력하고, 상기 요구 신호가 상기 제2의 상태가 아니며, 또한 상기 판정 신호가 상기 제1의 상태가 아닐 때, 상기 입력 프로세스 데이터를 출력 프로세스 데이터로서 출력하는 절환 수단

을 갖는 것을 특징으로 하는 화상 처리 장치.

청구항 7

제6항에 있어서,

상기 복수개의 실행 수단 각각은, 화상 데이터를 기억하는 데이터 기억 수단을 더 가지며,

상기 처리 수단은, 상기 화상 데이터의 처리로서, 상기 데이터 기억 수단에의 화상 데이터의 기입, 그 데이터 기억 수단으로부터의 화상 데이터의 판독, 또는 화상 데이터의 차분의 연산 중 어느 하나의 처리를 행하는 것을 특징으로 하는 화상 처리 장치.

청구항 8

제6항에 있어서,

상기 절환 수단은,

상기 요구 신호가 상기 제2의 상태가 아니며, 또한 상기 판정 신호가 상기 제1의 상태일 때에는, 하이 레벨 또는 로우 레벨의 데이터를 출력하는 것을 특징으로 하는 화상 처리 장치.

청구항 9

제6항에 있어서,

상기 화상 처리는, 움직임 벡터를 검출하는 처리인 것을 특징으로 하는 화상 처리 장치.

명 세 서

기 술 분 야

[0001] 본 발명은, 예를 들면 움직임 벡터의 검출 처리 등에 적용하기에 적합한 화상 처리 장치에 관한 것이다.

[0002] 상세하게는, 본 발명은, 화상 처리를 행하기 위한 각 프로세스를 실행하는 명령을 포함하는 프로세스 데이터를, 프로세스를 실행하는 직렬 접속된 복수개의 실행 수단을 이동시켜가며 화상 처리를 행하는 데 있어서, 소정수의 연속하는 실행 수단의 조마다, 그 입력측에 루트 선택 수단을 삽입하고, 루트 선택 수단은, 입력 프로세스 데이터에 따른 프로세스를 직후에 위치하는 조를 구성하는 실행 수단에 의해 실행하지 않을 때에는, 상기 입력 프로세스 데이터를 상기 직후에 위치하는 조의 출력측에 공급함으로써, 처리 시간의 단축 및 소비 전력의 저감을 도모하도록 한 화상 처리 장치에 관한 것이다.

[0003] 또한, 본 발명은, 화상 처리를 행하기 위한 각 프로세스를 실행하는 명령을 포함하는 프로세스 데이터를, 프로세스를 실행하는 직렬 접속된 복수개의 실행 수단을 이동시켜가며 화상 처리를 행하는 데 있어서, 실행 수단은, 입력 프로세스 데이터에 따른 프로세스를 자기가 실행하지 않을 때에는, 상기 입력 프로세스 데이터를 즉시 출력 프로세스 데이터로서 출력함으로써, 처리 시간의 단축을 도모하도록 한 화상 처리 장치에 따른 것이다.

배 경 기 술

[0004] 예를 들면, 화상 부호화 방식으로서의 MPEG(Moving Picture Experts Group)에서는, 임의의 크기의 블록 중, 주목하고 있는 주목 블록에 대하여, 움직임 벡터가 검출되고, 또한, 그 움직임 벡터에 기초하여 움직임 보상이 행해짐으로써, 주목 블록의 예측 화상이 구해진다. 그리고, MPEG에서는, 주목 블록의 각 화소와, 예측 화상의 대응하는 화소 간의 차분이 연산되어, 그 차분값이 부호화됨으로써, 고능률 압축이 실현되어 있다.

[0005] 움직임 벡터의 검출 알고리즘으로서는, 예를 들면, 블록 매칭법이 알려져 있다.

[0006] 예를 들면, 현재, 도 1에 도시한 바와 같이, 제f 프레임의 임의의 블록을, 주목 블록으로 함과 함께, 제f+1 프레임을, 움직임 벡터의 검출을 위해 참조하는 참조 프레임으로 하고, 제f+1 프레임으로부터 제f 프레임을 향한 움직임 벡터를, 주목 블록의 움직임 벡터로서 검출하는 경우, 블록 매칭법에서는, 제f+1 프레임의, 주목 블록의 위치를 중심으로 하는 소정의 범위가, 움직임 벡터의 탐사를 행하는 탐사 범위로서 설정된다. 또한, 제f+1 프레임의 탐사 범위로부터, 주목 블록과 동일한 크기의 블록이, 주목 블록의 예측 화상의 후보인 후보 블록으로서 선택되고, 주목 블록과 후보 블록 간의 차분에 관한 차분 정보가 구해진다.

[0007] 즉, 예를 들면, 주목 블록 및 후보 블록이, 가로×세로가 4×4 화소의 블록이라고 하면, 도 2에 도시한 바와 같이, 주목 블록의 각 화소와, 후보 블록의 대응하는 화소 간의 차분이 구해지고, 그 차분의 절대값(차분 절대값)이 구해진다. 또한, 그 차분 절대값의 총합이 구해지며, 탐사 범위로 선택할 수 있는 후보 블록 전체에 대하여, 전술한 바와 같은 차분 절대값의 총합이 구해진다.

[0008] 그리고, 탐사 범위로 선택할 수 있는 후보 블록 중에서, 차분 절대값의 총합을 최소로 하는 후보 블록(이하, 적절하게, 「최소 후보 블록」이라 함)이 구해지고, 그 최소 후보 블록으로부터 주목 블록을 향한 벡터가, 주목 블록의 움직임 벡터로서 구해진다.

[0009] 또한, 탐사 범위로서는, 주목 블록 및 후보 블록보다도 큰 범위가 이용되며, 주목 블록 및 후보 블록이, 전술한 바와 같이, 4×4 화소의 블록이라고 하면, 예를 들면, 30×30 화소 내지 50×50 화소 정도의 범위가, 탐사 범위로서 이용된다.

[0010] 도 3은, 블록 매칭법에 의해 움직임 벡터를 구하는, 종래의 움직임 벡터 검출 장치의 일례의 구성을 도시하고 있다.

[0011] 도 3의 움직임 벡터 검출 장치는, 화상 데이터를 기억하는 화상 메모리(201)와, 그 화상 데이터를 이용하여 연산을 행함으로써 움직임 벡터를 구하는 움직임 벡터 추출부(202)로 구성되어 있다. 화상 메모리(201)와, 움직임 벡터 추출부(202)는, 데이터 버스를 통해 접속되어 있다.

[0012] 이상과 같이 구성되는 움직임 벡터 검출 장치에서는, 화상 메모리(201)에, 주목 프레임과 참조 프레임의 화상 데이터가 기억된다. 그리고, 움직임 벡터 추출부(202)는, 화상 메모리(201)로부터, 주목 블록과 후보 블록을,

데이터 버스를 통해 판독하고, 그 주목 블록과 후보 블록 간의 차분 절대값의 총합을 구한다. 또한, 움직임 벡터 추출부(202)는, 탐사 범위 내에서 선택할 수 있는 후보 블록 중에서, 차분 절대값의 총합을 최소로 하는 후보 블록(최소 후보 블록)을 구하고, 그 최소 후보 블록으로부터 주목 블록을 향한 벡터를, 주목 블록의 움직임 벡터로서 구하여 출력한다.

[0013] 도 3의 움직임 벡터 검출 장치에서, 움직임 벡터를 검출하는 경우, 화상 메모리(201)로부터는, 빈번히, 대량의 화상 데이터가 판독되며, 데이터 버스를 통해, 움직임 벡터 추출부(202)에 공급된다.

[0014] 한편, 화상 데이터를 기억하는 화상 메모리(201)는, 일반적으로, 복수개의 메모리(반도체 메모리)로 구성된다. 즉, 도 3에서는, 화상 메모리(201)는, 6개의 메모리(201_1~201_6)로 구성되어 있다.

[0015] 화상 메모리(201)를 구성하는 메모리(201_1~201_6) 각각은, 비교적 넓은 면적을 점유하므로, 메모리(201_1~201_6) 각각으로부터, 움직임 벡터 추출부(202)에 화상 데이터를 전송하기 위해서는, 메모리(201_1~201_6) 각각과 움직임 벡터 추출부(202)를 연결하는 데이터 버스로서는, 비교적 긴 것이 필요하게 된다. 그리고, 긴 데이터 버스를 구동하는 경우에는, 여러가지 문제가 발생하는 것으로 된다.

[0016] 구체적으로는, 데이터 버스가 긴 경우, 그 데이터 버스를 구성하는 배선과 기판 간의 용량이 커져서, 데이터의 전송에, 큰 지연(배선 지연)이 발생한다. 또한, 데이터 버스를 구성하는 배선끼리 간에 발생하는 용량에 의해, 배선끼리 간에, 크로스토크가 발생하는 것으로 된다. 그리고, 최근에는, 반도체 프로세스의 미세화가 진행되고 있어서, 배선 간의 크로스토크가 큰 문제로 되고 있다.

[0017] 즉, 반도체 프로세스의 미세화에 의해, 배선끼리의 간격이 좁게 되면, 배선의 저항(배선 저항)이 커지므로, 이것을 방지하기 위해, 배선의 두께를 두껍게 할 필요가 있다. 그리고, 배선끼리의 간격이 좁아지며, 또한 배선의 두께가 두꺼워지면, 배선끼리 간의 용량이 커져서, 크로스토크를 무시할 수 없게 된다.

[0018] 또한, 종래에는, 배선에 대하여 발생하는 배선 용량에 대해서는, 배선과 기판 간의 용량을 고려하면 되며, 또한, 기판의 전위는 일정하기 때문에, 화상 메모리(201)의 시뮬레이션을 행하는 경우에, 배선 용량은, 그리 큰 문제로 되지 않았다.

[0019] 그러나, 배선 용량으로서, 전술한 바와 같이, 배선끼리 간의 용량이 커져서 지배적으로 되면, 주목하고 있는 배선에 인접한 배선에서의 신호 천이의 방법에 따라, 주목하고 있는 배선의 외관의 용량이 변화되어서, 배선 지연도 변화되기 때문에, 시뮬레이션을 행하는 것이 곤란하게 된다.

[0020] 또한, 화상 메모리(201)의 데이터 버스가 길면, 그 배선단에서의 반사에 의한 신호 파형의 혼란이 현저해진다.

[0021] 따라서, 도 4에 도시한 바와 같이, 화상 메모리(201)와 움직임 벡터 추출부(202) 사이에, 캐쉬 메모리(203)를 설치하고, 움직임 벡터 검출 장치를 구성하는 방법이 있다.

[0022] 도 4의 움직임 벡터 검출 장치에서, 캐쉬 메모리(203)는, 움직임 벡터 추출부(202)에서 빈번히 이용되는 화상 데이터를, 화상 메모리(201)로부터 판독하고 기억한다. 그리고, 움직임 벡터 추출부(202)는, 캐쉬 메모리(203)에 기억된 화상 데이터를 이용하여, 블록 매칭법에 의해 움직임 벡터를 구한다.

[0023] 도 4의 움직임 벡터 검출 장치에서는, 화상 메모리(201)로부터 캐쉬 메모리(203)에 전송되고 기억된 화상 데이터에 대해서는, 화상 메모리(201)로부터 판독할 필요가 없기 때문에, 긴 데이터 버스를 통해 화상 메모리(201)로부터 화상 데이터를 판독할 때의, 전술한 문제의 빈도를 저감할 수 있다.

[0024] 그러나, 이 경우, 화상 메모리(201)와는 별도로, 캐쉬 메모리(203)라는 용장 메모리가 필요하게 되어서, 그 오버헤드가 문제된다.

[0025] 따라서, 본 출원인은, 먼저, 움직임 벡터의 검출 처리를 행하기 위한 각 프로세스를 실행하는 명령을 포함하는 프로세스 데이터를, 프로세스를 실행하는 직렬 접속된 복수개의 실행 수단을 이동시켜가며 화상 처리를 행하는 것을 제안하였다(일본 특원 제2002-236877호 참조). 이것에 따르면, 예를 들면, 긴 데이터 버스나 캐쉬 메모리의 불필요한 설계 용이한 하드웨어에 의해, 움직임 벡터를 검출할 수 있다.

[0026] 〈발명의 개시〉

[0027] 본 발명은, 전술한 바와 같이, 프로세스 데이터를 프로세스를 실행하는 직렬 접속된 복수개의 실행 수단을 이동

시켜가며 화상 처리를 행하는 데 있어서, 처리 시간을 보다 더 단축하는 것을 도모하는 것을 목적으로 한다.

[0028] 본 발명에 따른 화상 처리 장치는, 화상 데이터를 취급하는 화상 처리를 행하기 위한 각 프로세스를 생성하며, 이 각 프로세스를 실행하는 명령을 포함하는 프로세스 데이터를 순차적으로 출력하는 프로세스 생성 수단과, 프로세스 데이터에 따라 프로세스를 실행하는, 직렬 접속된 복수개의 실행 수단을 구비하며, 프로세스 데이터는, 이 프로세스 데이터에 따른 프로세스를 실행하기 위한 실행 수단의 어드레스를 포함하고, 연속하는 실행 수단의 조마다, 이 조의 입력측에, 입력 프로세스 데이터를 이 조의 입력측 또는 출력측에 선택적으로 공급하는 루트 선택 수단이 삽입되는 것이다. 그리고, 루트 선택 수단은, 직후에 위치하는 조를 구성하는 각 실행 수단의 어드레스를 기억하는 제1 어드레스 기억 수단과, 입력 프로세스 데이터에 포함되는 실행 수단의 어드레스에 제1 어드레스 기억 수단에 기억되어 있는 어드레스 중 적어도 어느 하나가 존재할 때 제1의 상태로 되는 판정 신호를 출력하는 제1 어드레스 판정 수단과, 이 제1 어드레스 판정 수단으로부터 출력되는 판정 신호에 기초하여, 판정 신호가 제1의 상태일 때, 입력 프로세스 데이터를 직후에 위치하는 조의 입력측에 공급하고, 판정 신호가 제1의 상태가 아닐 때, 입력 프로세스 데이터를 직후에 위치하는 조의 출력측에 공급하는 제1 절환 수단을 갖는 것이다.

[0029] 본 발명에서, 프로세스 생성 수단은, 화상 데이터를 취급하는 화상 처리를 행하기 위한 각 프로세스를 생성하며, 이 각 프로세스를 실행하는 명령을 포함하는 프로세스 데이터를 순차적으로 출력한다. 이와 같이 순차적으로 출력되는 프로세스 데이터가 프로세스를 실행하는 직렬 접속된 복수개의 실행 수단을 이동해감으로써, 화상 처리가 행해진다. 화상 처리는, 예를 들면 움직임 벡터를 검출하기 위한 화상 처리이다.

[0030] 복수개의 실행 수단은, 소정수의 연속하는 실행 수단의 조로 나누어진다. 그리고, 각 조마다, 그 조의 입력측에, 입력 프로세스 데이터를 그 조의 입력측 또는 출력측에 선택적으로 공급하는 루트 선택 수단이 삽입된다.

[0031] 루트 선택 수단은, 직후에 위치하는 조를 구성하는 각 실행 수단의 어드레스를 기억하는 제1 어드레스 기억 수단을 갖고 있다. 또한, 프로세스 데이터는, 그 프로세스 데이터에 따른 프로세스를 실행하기 위한 실행 수단의 어드레스를 포함하고 있다.

[0032] 루트 선택 수단은, 입력 프로세스 데이터에 포함되는 실행 수단의 어드레스에 어드레스 기억 수단에 기억되어 있는 어드레스 중 적어도 어느 하나가 존재할 때, 입력 프로세스 데이터를 직후에 위치하는 조의 입력측에 공급하고, 존재하지 않을 때에는 입력 프로세스 데이터를 직후에 위치하는 조의 출력측에 공급한다.

[0033] 이에 따라, 프로세스 데이터는, 그 프로세스 데이터에 따른 프로세스를 실행하기 위한 실행 수단이 존재하지 않는 조를 바이패스하고 이동해가기 때문에, 처리 시간을 단축할 수 있다. 또한, 프로세스 데이터의 불필요한 이동이 없어지기 때문에, 소비 전력을 저감할 수 있다.

[0034] 또한, 본 발명에 따른 화상 처리 장치는, 화상 데이터를 취급하는 화상 처리를 행하기 위한 각 프로세스를 생성하며, 이 각 프로세스를 실행하는 명령을 포함하는 프로세스 데이터를 순차적으로 출력하는 프로세스 생성 수단과, 프로세스 데이터에 따라 프로세스를 실행하는, 직렬 접속된 복수개의 실행 수단을 구비하며, 프로세스 데이터는, 이 프로세스 데이터에 따른 프로세스를 실행하기 위한 실행 수단의 어드레스를 포함하는 것이다. 그리고, 복수개의 실행 수단 각각은, 입력 프로세스 데이터에 포함되는 명령에 따라 화상 데이터의 처리를 행하며, 그 처리 결과에 기초하여 입력 프로세스 데이터를 변경하여 출력할 프로세스 데이터를 얻음과 함께, 이 출력할 프로세스 데이터를 출력할 때 제2의 상태로 되는 요구 신호를 출력하는 처리 수단과, 자신의 어드레스를 기억하는 어드레스 기억 수단과, 입력 프로세스 데이터에 포함되는 실행 수단의 어드레스에 어드레스 기억 수단에 기억되어 있는 자신의 어드레스가 존재할 때 제1의 상태로 되는 판정 신호를 출력하는 어드레스 판정 수단과, 이 어드레스 판정 수단으로부터 출력되는 판정 신호 및 처리 수단으로부터 출력되는 요구 신호에 기초하여, 요구 신호가 제2의 상태일 때, 처리 수단에 의해 얻어진 출력할 프로세스 데이터를 출력 프로세스 데이터로서 출력하고, 요구 신호가 제2의 상태가 아니며, 또한 판정 신호가 제1의 상태가 아닐 때, 입력 프로세스 데이터를 출력 프로세스 데이터로서 출력하는 절환 수단을 갖는 것이다.

[0035] 본 발명에서, 프로세스 생성 수단은, 화상 데이터를 취급하는 화상 처리를 행하기 위한 각 프로세스를 생성하며, 이 각 프로세스를 실행하는 명령을 포함하는 프로세스 데이터를 순차적으로 출력한다. 이와 같이 순차적으로 출력되는 프로세스 데이터가 프로세스를 실행하는 직렬 접속된 복수개의 실행 수단을 이동해감으로써, 화상 처리가 행해진다. 프로세스 데이터는, 그 프로세스 데이터에 따른 프로세스를 실행하기 위한 실행 수단의 어드레스를 포함하고 있다.

[0036] 복수개의 실행 수단 각각은, 예를 들면 화상 데이터를 기억하는 데이터 기억 수단을 더 가지며, 화상 데이터의 처리로서, 데이터 기억 수단에의 화상 데이터의 기입, 이 데이터 기억 수단으로부터의 화상 데이터의 판독, 또는 화상 데이터의 차분의 연산 중 어느 하나의 처리를 행한다. 화상 처리는, 예를 들면 움직임 벡터를 검출하기 위한 화상 처리이다.

[0037] 복수개의 실행 수단 각각은, 처리 수단, 어드레스 기억 수단, 어드레스 판정 수단 및 절환 수단을 갖고 있다. 처리 수단은, 입력 프로세스 데이터에 포함되는 명령에 따라 화상 데이터의 처리를 행하며, 그 처리 결과에 기초하여 입력 프로세스 데이터를 변경하여 출력할 프로세스 데이터를 얻음과 함께, 이 출력할 프로세스 데이터를 출력할 때 제2의 상태로 되는 요구 신호를 출력한다. 어드레스 판정 수단은, 입력 프로세스 데이터에 포함되는 실행 수단의 어드레스에, 어드레스 기억 수단에 기억되어 있는 자신의 어드레스가 존재할 때 제1의 상태로 되는 판정 신호를 출력한다.

[0038] 그리고, 절환 수단은, 요구 신호가 제2의 상태일 때, 처리 수단에 의해 얻어진 출력할 프로세스 데이터를 출력 프로세스 데이터로서 출력하고, 요구 신호가 제2의 상태가 아니며, 또한 판정 신호가 제1의 상태가 아닐 때, 입력 프로세스 데이터를 출력 프로세스 데이터로서 출력한다.

[0039] 이에 따라, 프로세스 데이터는, 그 프로세스 데이터에 따른 프로세스를 실행하는 것이 아닌 실행 수단에 입력되었을 때, 즉시 다음의 실행 수단으로 이동하게 되므로, 처리 시간을 단축할 수 있다.

[0040] 또한, 절환 수단은, 요구 신호가 제2의 상태가 아니며, 또한 판정 신호가 제1의 상태일 때에는, 하이 레벨 또는 로우 레벨의 데이터를 출력하도록 하여도 된다. 이 경우, 실행 수단에 입력되는 프로세스 데이터에 따른 프로세스가, 그 실행 수단에 의해 실행된다. 입력 프로세스 데이터를 후단에 공급하지 않은 경우 및 처리 수단으로부터의 출력할 프로세스 데이터를 후단에 공급하지 않은 상태에서는, 하이 레벨 또는 로우 레벨의 데이터가 출력되므로, 후단에 잘못된 프로세스 데이터가 공급되는 것을 방지할 수 있다.

산업상 이용 가능성

[0232] 이상과 같이, 본 발명에 따른 화상 처리 장치는, 처리시간의 단축 및 소비 전력의 저감을 도모할 수 있으며, 예를 들면 블록 매칭에 의해 움직임 벡터를 검출하는 용도 등에 적용할 수 있다.

도면의 간단한 설명

[0041] 도 1은, 블록 매칭법을 설명하는 도면.

[0042] 도 2는, 블록 매칭법을 설명하는 도면.

[0043] 도 3은, 종래의 움직임 벡터 검출 장치의 일례의 구성을 도시하는 블록도.

[0044] 도 4는, 종래의 움직임 벡터 검출 장치의 다른 일례의 구성을 도시하는 블록도.

[0045] 도 5는, 실시 형태로서의 움직임 벡터 검출 장치의 구성을 도시하는 블록도.

[0046] 도 6은, 프로세스 패킷의 포맷을 나타내는 도면.

[0047] 도 7은, 프로세스 생성 처리를 설명하기 위한 흐름도.

[0048] 도 8은, 루트 선택부의 구성예를 도시하는 블록도.

[0049] 도 9는, 연산 처리 유닛의 구성예를 도시하는 블록도.

[0050] 도 10은, 판정 신호 s_1, 요구 신호 s_2와 절환 스위치의 전환 관계를 나타내는 도면.

[0051] 도 11은, 연산부의 프로세스 실행 처리를 설명하는 흐름도.

[0052] 도 12는, 연산부의 프로세스 실행 처리(기입 명령인 경우)를 설명하기 위한 도면.

[0053] 도 13a 및 도 13b는, 메모리에 화상 데이터가 기입된 상태를 나타내는 도면.

[0054] 도 14는, 연산부의 프로세스 실행 처리(판독 명령인 경우)를 설명하기 위한 도면.

[0055] 도 15는, 연산부의 프로세스 실행 처리(차분 절대값의 합 연산 명령인 경우)를 설명하기 위한 도면.

[0056] 도 16은, 연산 처리 유닛의 처리를 설명하기 위한 도면.

[0057] 도 17a 및 도 17b는, 프로세스 패킷의 변화를 나타내는 도면.

[0058] 〈발명을 실시하기 위한 최량의 형태〉

[0059] 이하, 본 발명의 실시 형태를 도면을 참조하여 설명한다.

[0060] 도 5는, 실시 형태로서의 움직임 벡터 검출 장치(100)의 구성을 도시하고 있다. 이 움직임 벡터 검출 장치(100)에서는, 전술한 블록 매칭법에 의해 움직임 벡터가 검출된다.

[0061] 이 움직임 벡터 검출 장치(100)는, 프로세스 생성부(101)를 갖고 있다. 이 프로세스 생성부(101)는, 움직임 벡터를 검출하는 처리를 행하기 위한 각 프로세스를 생성하고, 그 각 프로세스를 실행하는 명령을 포함하는 패킷인 프로세스 패킷(프로세스 데이터)을 순차적으로 출력한다. 이 프로세스 생성부(101)에는 움직임 벡터의 검출 대상인 화상 데이터가 공급된다. 프로세스 생성부(101)는, 필요에 따라, 프로세스 패킷에 그 화상 데이터도 배치한다.

[0062] 또한, 움직임 벡터 검출 장치(100)는, 프로세스 패킷에 따라 프로세스를 실행하는, 직렬 접속된 복수개, 본 실시 형태에서는 12개의 연산 처리 유닛(102_1～102_{12})을 갖고 있다. 연산 처리 유닛(102_1~102_{12})는, 각각 실행 수단을 구성하고 있다. 이들 12개의 연산 처리 유닛(102_1～102_{12})은, 프로세스 생성부(101)의 후단에 접속되어 있다.

[0063] 연산 처리 유닛(102_n)(n=1～12)은, 전단의 연산 처리 유닛(102_{n-1})(또는 루트 선택부)으로부터 공급되는 프로세스 패킷을 수신하고, 그 프로세스 패킷에 포함되는 명령에 따른 처리를 행한다. 또한, 연산 처리 유닛(102_n)은, 필요에 따라, 그 처리 결과를, 프로세스 패킷에 배치하고, 후단의 연산 처리 유닛(102_{n+1})(또는 루트 선택부, 또는 집계부)에 공급한다. 이에 따라, 최후의 연산 처리 유닛(102_{12})은, 후술하는 바와 같이, 주목 블록과 임의의 후보 블록 간의 차분 절대값의 총합을 배치한 프로세스 패킷을 출력한다. 루트 선택부 및 집계부에 대해서는 후술한다.

[0064] 또한, 움직임 벡터 검출 장치(100)는, 3개의 루트 선택부(104_1～104_3)를 갖고 있다. 전술한 12개의 연산 처리 유닛(102_1~102_{12})은, 4개의 연속하는 연산 처리 유닛의 조로 나누어진다. 즉, 유닛(102_1～102_4)의 조(102a)와, 유닛(102_5～102_8)의 조(102b)와, 유닛102_9～102_{12}의 조(102c)로 나누어진다. 전술한 3개의 루트 선택부(104_1～104_3)는, 각각 조(102a～102c)의 입력측에 삽입된다. 또한, 조의 개수는 3개가 아니어도 되며, 또한 각 조를 구성하는 연산 처리 유닛(102_n)의 개수는 동일하지 않아도 된다.

[0065] 루트 선택부(104_1～104_3) 각각은, 입력 프로세스 패킷을, 직후에 위치하는 조(102a～102c)의 입력측 또는 출력측에 선택적으로 공급한다. 즉, 루트 선택부(104_1～104_3) 각각은, 입력 프로세스 패킷에 따른 프로세스를 실행하는 연산 처리 유닛(102_n)이, 직후에 위치하는 조(102a～102c)에 존재할 때, 그 입력 프로세스 패킷을 그 조의 입력측에 공급하며, 존재하지 않을 때 그 입력 프로세스 패킷을 그 조의 출력측에 공급한다.

[0066] 또한, 움직임 벡터 검출 장치(100)는, 집계부(103)를 갖고 있다. 이 집계부(103)는, 최후의 연산 처리 유닛(102_{12})의 출력측에 접속되어 있으며, 이 연산 처리 유닛(102_{12})이 출력하는 프로세스 패킷에 배치된 주목 블록과 후보 블록 간의 차분 절대값의 총합으로부터, 움직임 벡터를 구하여 출력한다.

[0067] 여기서, 전술한 프로세스 생성부(101)로부터 출력되며, 연산 처리 유닛(102_1～102_{12})을 이동해가는, 프로세스 패킷의 포맷을 설명한다. 도 6은, 프로세스 패킷의 포맷예를 나타내고 있다.

[0068] 프로세스 패킷은, 예를 들면, 그 선두로부터, 처리 유닛 어드레스부, PID(Process Identification)부, 상태부, 명령부, 어드레스부, 데이터부가, 순차적으로 설치되어 구성되어 있다.

[0069] 처리 유닛 어드레스부에는, 상기 프로세스 패킷에 따른 프로세스를 실행하기 위한 1개 또는 복수개의 연산 처리 유닛의 어드레스가 배치된다.

[0070] PID부에는, PID가 배치된다. 여기서, PID로서는, 어떤 주목 블록의 움직임 벡터를 구할 때까지 행할 개개의 프로세스를 구별할 수 있는 정보이면, 어떠한 정보라도 채용하는 것이 가능하다. 즉, PID로서, 예를 들면, 주목 블록의 위치를 나타내는 어드레스와, 후보 블록의 위치를 나타내는 어드레스의 조합 등을 채용하는 것이 가능하

다.

[0071] 또한, 주목 블록에 대한 후보 블록에, 예를 들면 일련 번호로 되는 번호 정보를 부여해두는 경우에는, PID로서, 그 번호 정보를 채용하는 것이 가능하다. 또한, PID로서, 번호 정보를 채용하는 경우, 집계부(103)에서, 모든 일련 번호의 프로세스 패킷이 갖추어짐으로써, 어떤 주목 블록에 대하여, 탐사 범위 내에서 선택 가능한 모든 후보 블록 간의 차분 절대값의 총합이 얻어진 것을 인식하는 것이 가능하게 된다.

[0072] 상태부는, 예를 들면, 화상 기입 상태부, 화상 판독 상태부, 주목 블록 판독 상태부, 후보 블록 판독 상태부, 차분 절대값 연산 상태부로 구성된다.

[0073] 화상 기입 상태부에는, 후술하는 연산 처리 유닛(102_n)의 메모리(120)에 대하여, 어떤 화상 데이터의 기입이, 아직 행해져 있지 않은 것을 나타내는 상태 정보 「미(未)」, 그 화상 데이터의 기입이 행해지고 있는 도중인 것을 나타내는 상태 정보 「중(中)」, 그 화상 데이터의 기입이 종료된 것을 나타내는 상태 정보 「종(終)」 중 어느 하나가 세트된다.

[0074] 즉, 예를 들면, 지금, 연산 처리 유닛(102_n)을 구성하는 메모리(120)를, 메모리(120_n)으로 나타내기로 하고, 어떤 화상 데이터를, 예를 들면, 연산 처리 유닛(102_i)의 메모리(120_i)와 연산 처리 유닛(102_j)의 메모리(120_j)의 2개로 나누어 기입하는 경우를 생각하면, 메모리(120_i)와 메모리(120_j) 중 어디에도 화상 데이터가 기입되어 있지 않은 경우, 화상 기입 상태부에는, 상태 정보 「미」가 세트된다. 또한, 메모리(120_i)와 메모리(120_j) 중 어느 한쪽에만 화상 데이터가 기입된 경우, 화상 기입 상태부에는, 상태 정보 「중」이 세트된다. 또한, 메모리(120_i)와 메모리(120_j)의 양쪽에 화상 데이터가 기입된 경우, 화상 기입 상태부에는, 상태 정보 「종」이 세트된다.

[0075] 또한, 상태부에 세트되는 상태 정보는, 연산 처리 유닛(102_n)에 대하여, 화상 데이터의 기입 등의 상태가 어떻게 되어 있는지를 알리는 메시지의 역할을 완수한다.

[0076] 화상 판독 상태부에는, 메모리(120)로부터의, 어떤 화상 데이터의 판독이, 아직 행해져 있지 않은 것을 나타내는 상태 정보 「미」, 그 화상 데이터의 판독이 행해지고 있는 도중인 것을 나타내는 상태 정보 「중」, 그 화상 데이터의 판독이 종료된 것을 나타내는 상태 정보 「종」 중 어느 하나가 세트된다.

[0077] 주목 블록 판독 상태부에는, 메모리(120)로부터의, 주목 블록의 화상 데이터의 판독이, 아직 행해져 있지 않은 것을 나타내는 상태 정보 「미」, 주목 블록의 화상 데이터의 판독이 행해지고 있는 도중인 것을 나타내는 상태 정보 「중」, 주목 블록의 화상 데이터의 판독이 종료된 것을 나타내는 상태 정보 「종」 중 어느 하나가 세트된다.

[0078] 후보 블록 판독 상태부에는, 메모리(120)로부터의, 후보 블록의 화상 데이터의 판독이, 아직 행해져 있지 않은 것을 나타내는 상태 정보 「미」, 후보 블록의 화상 데이터의 판독이 행해지고 있는 도중인 것을 나타내는 상태 정보 「중」, 후보 블록의 화상 데이터의 판독이 종료된 것을 나타내는 상태 정보 「종」 중 어느 하나가 세트된다.

[0079] 차분 절대값 연산 상태부에는, 어떤 주목 블록과 후보 블록 간의 차분 절대값의 총합(차분 절대값의 합)의 연산이, 아직 행해져 있지 않은 것을 나타내는 상태 정보 「미」, 차분 절대값의 합의 연산이 행하여지고 있는 도중인 것을 나타내는 상태 정보 「중」, 차분 절대값의 합의 연산이 종료된 것을 나타내는 상태 정보 「종」 중 어느 하나가 세트된다.

[0080] 여기서, 상태 정보는, 예를 들면, 2 비트로 하여, 「미」, 「중」, 「종」에는, 각각 「11」, 「10」, 「00」을 할당할 수 있다.

[0081] 명령부에는, 행할 처리를 지시하는 명령이 배치된다. 여기서, 명령으로서는, 예를 들면, 메모리(120)에의 화상 데이터의 기입을 지시하는 기입 명령, 메모리(120)로부터의 화상 데이터의 판독을 지시하는 판독 명령, 주목 블록과 후보 블록 간의 차분 절대값의 합의 연산을 지시하는 차분 절대값의 합 연산 명령, 각 후보 블록에 대하여 구해진 차분 절대값의 합 중 최소값을 구하고, 그 최소값에 기초하여, 주목 블록의 움직임 벡터를 구할 것을 지시하는 최소값 판정 명령의 4개의 명령이, 적어도 준비되어 있다.

[0082] 또한, 명령이, 전술한 기입 명령, 판독 명령, 차분 절대값의 합 연산 명령, 및 최소값 판정 명령의 $4(=2^2)$개인

경우에는, 명령부는, 2 비트로 충분하다. 단, 명령부는, 장래의 명령의 종류의 확장도 고려하여, 2 비트보다 많은, 예를 들면, 4 비트 등으로 해두는 것이 바람직하다.

[0083] 어드레스부는, 필드 지정부, 판독 어드레스부, 기입 어드레스부, 주목 블록 어드레스부, 후보 블록 어드레스부로 구성된다.

[0084] 필드 지정부에는, 후보 블록의 필드를 나타내는 필드 정보가 세트된다. 즉, 본 실시 형태에서는, 후술하는 바와 같이, 연산 처리 유닛(102_n)의 메모리(120)에, 5 필드의 화상 데이터가 기억되는데, 필드 지정부에는, 그 메모리(120)에 기억된 화상 데이터 중, 후보 블록이 존재하는 필드를 나타내는 필드 정보가 세트된다. 또한, 5 필드의 화상 데이터를 기억하는 경우, 필드 정보는, 3 비트로 충분하지만, 장래의 확장을 고려하여, 3 비트보다 많은, 예를 들면, 4 비트 등으로 하는 것이 바람직하다.

[0085] 판독 어드레스부에는, 메모리(120)로부터 화상 데이터를 판독하는 경우에, 즉, 명령부에 판독 명령이 세트되어 있는 경우에, 그 화상 데이터를 판독하는 메모리(120)의 어드레스가 배치된다.

[0086] 기입 어드레스부에는, 메모리(120)에 화상 데이터를 기입하는 경우에, 즉, 명령부에 기입 명령이 세트되어 있는 경우에, 그 화상 데이터를 기입하는 메모리(120)의 어드레스가 배치된다.

[0087] 주목 블록 어드레스부에는, 주목 블록의 어드레스가 배치된다. 여기서, 주목 블록의 어드레스로서는, 예를 들면, 그 주목 블록의 필드(주목 필드)에서의, 주목 블록의 좌측위의 화소의 위치를 나타내는 위치 정보를 채용할 수 있다. 따라서, 예를 들면, 1 필드가, 720×240 화소로 구성됨과 함께, 주목 블록이, 4×2 화소로 구성되는 것으로 하면, 주목 블록의 어드레스는, 21600(=720/4×240/2) 가지가 존재하므로, 15 비트로 나타낼 수 있다.

[0088] 후보 블록 어드레스부에는, 후보 블록의 어드레스가 배치된다. 여기서, 후보 블록의 어드레스로서는, 예를 들면, 그 후보 블록의 필드(참조 필드)에서의, 후보 블록의 좌측위의 화소의 위치를 나타내는 위치 정보를 채용할 수 있다. 따라서, 예를 들면, 전술한 바와 같이, 1 필드가, 720×240 화소로 구성되는 것으로 하면, 후보 블록의 어드레스는, 172800(=720×240) 가지가 존재하므로, 18 비트로 나타낼 수 있다.

[0089] 데이터부는, 주목 블록 데이터부, 후보 블록 데이터부, 차분 절대값의 합부, 후보 벡터부로 구성된다.

[0090] 주목 블록 데이터부에는, 주목 블록의 화상 데이터, 즉, 주목 블록을 구성하는 화소(의 화소값)가 배치된다. 따라서, 예를 들면, 전술한 바와 같이, 주목 블록이, 4×2 화소로 구성되는 것으로 하고, 또한, 1 화소에, 예를 들면, 8 비트가 할당되어 있다고 한 경우에는, 주목 블록 데이터부는, 64(=4×2×8) 비트로 구성되게 된다.

[0091] 후보 블록 데이터부에는, 후보 블록의 화상 데이터, 즉, 후보 블록을 구성하는 화소가 배치된다. 따라서, 후보 블록이, 예를 들면, 전술한 바와 같이, 주목 블록과 마찬가지의 4×2 화소로 구성되는 것으로 하고, 또한, 1 화소에, 예를 들면, 8 비트가 할당되어 있다고 한 경우에는, 후보 블록 데이터부는, 주목 블록 데이터부와 마찬가지로, 64(=4×2×8) 비트로 구성되게 된다.

[0092] 차분 절대값의 합부에는, 주목 블록과 후보 블록의 차분 절대값의 합이 배치된다. 여기서, 전술한 바와 같이, 예를 들면, 1 화소에 8 비트가 할당되어 있는 경우, 주목 블록이 있는 화소와 그 화소에 대응하는 후보 블록의 화소의 차분 절대값은, 9 비트로 표시된다.

[0093] 후보 벡터부에는, 후보 블록으로부터 주목 블록을 향한 벡터가, 주목 블록의 움직임 벡터의 후보(후보 벡터)로서 배치된다. 또한, 후보 벡터는, 주목 블록 어드레스부에 배치된 주목 블록의 어드레스와, 후보 블록 어드레스부에 배치된 후보 블록의 어드레스로부터 구하는 것이 가능하다. 또한, 예를 들면, 현재, 탐사 범위를, 63×63 화소 이하로 하면, 후보 벡터의 x 방향(가로 방향) 성분과, y 방향(세로 방향) 성분은, 모두, 6 비트로 나타낼 수 있기 때문에, 후보 벡터부는, 12(=6+6) 비트로 구성할 수 있다.

[0094] 다음으로, 도 7의 흐름도를 참조하여, 전술한 프로세스 생성부(101)가, 어떤 1개의 블록을 주목 블록으로 하고, 그 주목 블록의 움직임 벡터를 검출하기 위해 행하는 처리(프로세스 생성 처리)에 대하여 설명한다.

[0095] 먼저 처음에, 단계 S1에서, 프로세스 생성부(101)는, 주목 블록이 존재하는 주목 필드 중 주목 블록의 화상 데이터와, 후보 블록이 존재하는 참조 필드 중 탐사 범위의 화상 데이터를, 적어도, 메모리(120)에 기입하는 기입 프로세스의 일부 또는 전부를 생성하며, 그 기입 프로세스를 실행하는 기입 명령을 포함하는 프로세스 패킷을 생성하고, 단계 S2로 진행한다.

[0096] 또한, 프로세스 생성부(101)는, 단계 S1에서, 도 6에 도시한 프로세스 패킷의 명령부에 기입 명령을 배치하는

것 외에, 메모리(120)에 기입하는 주목 블록이나 후보 블록의 화상 데이터를, 주목 블록 데이터부나 후보 블록 데이터부에 배치함과 함께, 그 화상 데이터를 기입하는 메모리(120)의 어드레스를, 기입 어드레스부에 배치한다.

[0097] 또한, 프로세스 생성부(101)는, 처리 유닛 어드레스부에, 그 기입 프로세스를 실행하는 1개 또는 복수개의 연산 처리 유닛(102_n)의 어드레스를 배치한다. 또한, 프로세스 생성부(101)는, 필요에 따라, 프로세스 패킷의 상태부에, 상태 정보를 배치한다.

[0098] 단계 S2에서는, 프로세스 생성부(101)는, 직전의 단계 S1에서 생성한 프로세스 패킷을 출력하고, 단계 S3으로 진행한다.

[0099] 단계 S3에서는, 프로세스 생성부(101)는, 필요한 화상 데이터, 즉, 여기서는, 주목 블록의 화상 데이터와, 후보 블록이 존재하는 참조 필드 중 탐사 범위의 화상 데이터를, 적어도, 메모리(120)에 기입하기 위한 프로세스 전체를 생성하였는지 여부를 판정한다. 단계 S3에서, 필요한 화상 데이터를 메모리(120)에 기입하기 위한 프로세스 전체를, 아직 생성하고 있지 않다고 판정한 경우, 단계 S1로 되돌아가고, 프로세스 생성부(101)는, 필요한 화상 데이터를 메모리(120)에 기입하기 위한 프로세스 중, 아직 기입되어 있지 않은 화상 데이터를 기입하기 위한 프로세스를 생성하고, 이하, 마찬가지의 처리를 반복한다.

[0100] 또한, 단계 S3에서, 프로세스 생성부(101)는, 필요한 화상 데이터를 메모리(120)에 기입하기 위한 프로세스 전체를 생성하였다고 판정한 경우, 단계 S4로 진행한다. 단계 S4에서는, 프로세스 생성부(101)는, 탐사 범위 내에서 선택할 수 있는 임의의 후보 블록으로부터, 주목 블록에의 벡터(후보 벡터)에 대하여, 주목 블록과 후보 블록 간의 차분 절대값의 합을 구하기 위한 차분 절대값의 합 연산 프로세스를 생성하고, 그 차분 절대값의 합 연산 프로세스를 실행하는 차분 절대값의 합 연산 명령을 포함하는 프로세스 패킷을 생성하고, 단계 S5로 진행한다.

[0101] 또한, 프로세스 생성부(101)는, 단계 S4에서, 도 6에 도시한 프로세스 패킷의 명령부에 차분 절대값의 합 연산 명령을 배치하는 것 외에, 메모리(120)에 기입된 주목 블록이나 후보 블록의 어드레스를, 주목 블록 어드레스부나 후보 블록 어드레스부에 배치함과 함께, 후보 벡터를, 후보 벡터부에 배치한다.

[0102] 또한, 프로세스 생성부(101)는, 처리 유닛 어드레스부에, 그 차분 절대값의 합 연산 프로세스를 실행하는 1개 또는 복수개의 연산 처리 유닛(102_n)의 어드레스를 배치한다. 또한, 프로세스 생성부(101)는, 후보 블록의 필드를 나타내는 필드 정보를, 필드 지정부에 배치함과 함께, 필요에 따라, 프로세스 패킷의 상태부에, 상태 정보를 배치한다.

[0103] 단계 S5에서는, 프로세스 생성부(101)는, 직전의 단계 S4로 생성한 프로세스 패킷을 출력하여, 단계 S6으로 진행한다.

[0104] 단계 S6에서는, 프로세스 생성부(101)는, 주목 블록의 탐색 범위 내에서 선택할 수 있는 후보 벡터 전체에 대하여, 차분 절대값의 합을 연산하기 위한 차분 절대값의 합 연산 프로세스를 생성하였는지 여부를 판정한다.

[0105] 단계 S6에서, 프로세스 생성부(101)는, 주목 블록의 탐색 범위 내에서 선택할 수 있는 후보 벡터 전체에 대하여, 차분 절대값의 합을 연산하기 위한 차분 절대값의 합 연산 프로세스를, 아직 생성하고 있지 않다고 판정한 경우, 단계 S4로 되돌아가고, 아직 차분 절대값의 합 연산 프로세스를 생성하고 있지 않은 후보 벡터에 대하여, 차분 절대값의 합을 연산하기 위한 차분 절대값의 합 연산 프로세스를 생성하고, 이하, 마찬가지의 처리를 반복한다.

[0106] 또한, 단계 S6에서, 프로세스 생성부(101)는, 주목 블록의 탐색 범위 내에서 선택할 수 있는 후보 벡터 전체에 대하여, 차분 절대값의 합을 연산하기 위한 차분 절대값의 합 연산 프로세스를 생성하였다고 판정한 경우, 단계 S7로 진행한다.

[0107] 단계 S7에서는, 프로세스 생성부(101)는, 주목 블록의 탐색 범위 내에서 선택할 수 있는 후보 벡터 전체에 대하여 구해진 차분 절대값의 합 중 최소값을 구하고, 그 최소값에 기초하여, 주목 블록의 움직임 벡터를 구하는 최소값 판정 프로세스를 생성하며, 그 최소값 판정 프로세스를 실행하는 최소값 판정 명령을 포함하는 프로세스 패킷을 생성하고, 단계 S8로 진행한다.

[0108] 단계 S8에서는, 프로세스 생성부(101)는, 단계 S7에서 생성한 프로세스 패킷을 출력하고, 그 후에 처리를 종료한다.

[0109] 도 5에 도시하는 움직임 벡터 검출 장치(100)에서, 전술한 바와 같이, 프로세스 생성부(101)는, 움직임 벡터를 검출하는 처리를 행하기 위한 각 프로세스를 생성하고, 이 각 프로세스를 실행하는 명령을 포함하는 프로세스 패킷을 순차적으로 출력한다. 이 각 프로세스 패킷은 연산 처리 유닛(102_1~102_{12})을 이동해가고, 이에 따라, 연산 처리 유닛(102_1~102_{12})에서 움직임 벡터를 검출하기 위한 각 프로세스가 순차적으로 실행되어 간다.

[0110] 그리고, 최종적으로, 주목 블록의 탐사 범위 내에서 선택할 수 있는 후보 벡터 각각에 대한 차분 절대값의 합이 배치된 프로세스 패킷이 집계부(103)에서 수신된다. 집계부(103)는, 최소값 판정 명령을 포함하는 프로세스 패킷을 수신하면, 차분 절대값의 합 연산 프로세스에 대응하는 프로세스 패킷으로부터, 최소의 차분 절대값의 합이 배치되어 있는 것을 선택하고, 그 프로세스 패킷에 배치되어 있는 후보 벡터를 주목 블록의 움직임 벡터로서 출력한다.

[0111] 또한, 본 실시 형태에서는, 프로세스 생성부(101)로부터 출력되는 각 프로세스 패킷이 연산 처리 유닛(102_1~102_{12}) 전체를 이동해가는 것은 아니다. 루트 선택부(104_1~104_3)에 의해, 각 프로세스 패킷은, 적절하게 바이패스되어 이동해간다.

[0112] 즉, 루트 선택부(104_1~104_3) 각각은, 입력 프로세스 패킷에 따른 프로세스를 실행하는 연산 처리 유닛(102_n)이, 직후에 위치하는 조(102a~102c)에 존재하지 않을 때에는, 상기 입력 프로세스 패킷을, 그 조(102a~102c)의 입력측에 공급하는 것은 아니며, 그 조의 출력측에 공급한다. 이와 같이, 프로세스 패킷이, 그 프로세스 패킷에 따른 프로세스를 실행하기 위한 연산 처리 유닛(102_n)이 존재하지 않는 조를 바이패스하고 이동해가기 때문에, 처리 시간을 단축할 수 있으며, 또한 프로세스 패킷의 불필요한 이동이 없어지기 때문에, 소비 전력을 저감할 수 있다.

[0113] 다음으로, 루트 선택부(104_n)(n=1~3)의 상세 내용을 설명한다. 도 8은, 루트 선택부(104_n)의 구성예를 도시하고 있다.

[0114] 이 루트 선택부(104_n)는, FIFO(First-In First-Out) 메모리(131)를 갖고 있다. 이 FIFO 메모리(131)는, 전단의 프로세스 생성부(101)(또는 루트 선택부, 또는 연산 처리 유닛)로부터 데이터 버스를 통해 공급되는 프로세스 패킷(입력 프로세스 패킷)을 보유한다. 이 FIFO 메모리(131)는, 거기에 공급되는 시스템 클럭에 동기하여, 입력 프로세스 패킷을 입력받아, 일시적으로 기억한다.

[0115] 또한, 루트 선택부(104_n)는, 어드레스 기억부(132) 및 어드레스 판정부(133)를 갖고 있다. 어드레스 기억부(132)에는, 미리, 직후에 위치하는 조에 존재하는 모든 연산 처리 유닛(102_n)의 어드레스가 기억되어 있다. 예를 들면, 루트 선택부(104_1)에 대해서는, 어드레스 기억부(132)에는, 직후에 대응하는 조(102a)에 존재하는 연산 처리 유닛(102_1~102_4)의 어드레스가 기억되어 있다. 루트 선택부(104_2, 104_3)에 대해서도 마찬가지다.

[0116] 어드레스 판정부(133)는, FIFO 메모리(131)로부터 출력되는 프로세스 패킷 PPe에 따른 프로세스가, 이 루트 선택부(104_n)의 직후에 위치하는 조에 존재하는 연산 처리 유닛(102_n) 중 어느 것에 의해 실행되는 것인지의 여부를 판정한다.

[0117] 즉, 어드레스 판정부(133)는, FIFO 메모리(131)로부터 출력되는 프로세스 패킷 PPe의 처리 유닛 어드레스부(도 6 참조)에 기억되어 있는 1개 또는 복수개의 어드레스에, 어드레스 기억부(132)에 기억되어 있는 어드레스 중 적어도 어느 하나가 존재하는지의 여부를 판정하고, 존재할 때 하나의 상태, 예를 들면 「1」로 되며, 존재하지 않을 때 다른 상태, 예를 들면 「0」으로 되는 판정 신호 s_3을 출력한다.

[0118] 또한, 루트 선택부(104_n)는, 절환 수단으로서의 절환 스위치(134)를 갖고 있다. 이 절환 스위치(134)는, 판정 신호 s_3에 기초하여, FIFO 메모리(131)로부터 출력되는 프로세스 패킷 PPe를, 상기 루트 선택부(104_n)의 직후에 위치하는 조의 입력측 또는 출력측에 선택적으로 공급한다. 이 절환 스위치(134)는, 판정 신호 s_3이 「1」일 때에는 A측에 접속되며, 판정 신호 s_3이 「0」일 때에는 B측에 접속된다.

[0119] FIFO 메모리(131)의 출력측은 시간 조정용 지연 회로(135)를 통해 절환 스위치(134)의 가동 단자에 접속된다. 이 절환 스위치(134)의 A측의 고정 단자는 상기 루트 선택부(104_n)의 직후에 위치하는 조의 입력측에 접속되며,

그 B측의 고정 단자는, 상기 루트 선택부(104_n)의 직후에 위치하는 조의 출력측에 접속된다.

[0120] 전술한 바와 같이, 어드레스 판정부(133)에서는 FIFO 메모리(131)로부터 출력되는 프로세스 패킷 PPe의 처리 유닛 어드레스부에 기억되어 있는 어드레스에 기초하여, 상기 프로세스 패킷 PPe에 따른 프로세스가, 이 루트 선택부(104_n)의 직후에 위치하는 조에 존재하는 연산 처리 유닛(102_n)에 의해 실행되는 것인지 여부를 판정한다. 그리고, 후술하는 바와 같이, 그 판정 결과인 판정 신호 s_3에 기초하여, 절환 스위치(134)의 전환이 제어되어, 상기 프로세스 패킷 PPe가, 상기 루트 선택부(104_n)의 직후에 위치하는 조의 입력측 또는 출력측에 공급된다.

[0121] 적어도, 어드레스 판정부(133)로 판정이 행해질 때에는, 프로세스 패킷 PPe의 처리 유닛 어드레스부의 부분은 이미 FIFO 메모리(131)로부터 출력된 상태에 있으며, FIFO 메모리(131)의 출력측을 절환 스위치(134)의 가동 단자에 직접 접속하였다면, 상기 프로세스 패킷 PPe를 빠뜨리지 않고 완전한 상태로 상기 절환 스위치(134)로부터 출력할 수 없는 경우가 발생한다.

[0122] 전술한 시간 조정용 지연 회로(135)는, 이 상기 프로세스 패킷 PPe를 빠뜨리지 않고 완전한 상태로 절환 스위치(134)로부터 출력하기 위한 것이며, 적어도, FIFO 메모리(131)로부터 상기 프로세스 패킷 PPe의 출력이 개시되고 나서 어드레스 판정부(133)로 판정 결과가 나오기까지의 지연 시간을 갖도록 된다.

[0123] 도 8에 도시하는 루트 선택부(104_n)의 동작을 설명한다.

[0124] 전단의 프로세스 생성부(101)(또는 루트 선택부, 또는 연산 처리 유닛)로부터 데이터 버스를 통해 공급되는 프로세스 패킷(입력 프로세스 패킷)은, FIFO 메모리(131)에 입력되어서, 일시적으로 기억된다. 그리고, 이 FIFO 메모리(131)로부터 출력되는 프로세스 패킷 PPe는, 지연 회로(135)를 통해 절환 스위치(134)의 가동 단자에 공급된다.

[0125] 어드레스 판정부(133)는, 프로세스 패킷 PPe의 처리 유닛 어드레스부(도 6 참조)에 기억되어 있는 어드레스에, 어드레스 기억부(132)에 기억되어 있는 어드레스 중 적어도 어느 하나가 존재하는지의 여부를 판정한다.

[0126] 그리고, 어드레스 판정부(133)는, 긍정의 판정 결과를 얻을 때에는, 상기 프로세스 패킷 PPe에 따른 프로세스는, 이 루트 선택부(104_n)의 직후에 위치하는 조 내의 연산 처리 유닛(102_n)에 의해 실행되는 것으로 하고, 판정 신호 s_3을 「1」로 한다. 이에 따라, 절환 스위치(134)의 A측에 접속되며, 상기 프로세스 패킷 PPe는 절환 스위치(134)의 A측을 통해, 상기 루트 선택부(104_n)의 직후에 위치하는 조의 입력측에 공급된다.

[0127] 한편, 어드레스 판정부(133)는, 부정의 판정 결과를 얻을 때에는, 상기 프로세스 패킷 PPe에 따른 프로세스는, 이 루트 선택부(104_n)의 직후에 위치하는 조 내의 연산 처리 유닛(102_n)에 의해 실행되는 것이 아니라고 하고, 판정 신호 s_3을 「0」으로 한다. 이에 따라, 절환 스위치(134)는 B측에 접속되며, 상기 프로세스 패킷 PPe는 절환 스위치(134)의 B측을 통해, 상기 루트 선택부(104_n)의 직후에 위치하는 조의 출력측에 공급된다.

[0128] 다음으로, 연산 처리 유닛(102_n)의 상세 내용을 설명한다. 도 9는, 연산 처리 유닛(102_n)의 구성예를 도시하고 있다.

[0129] 이 연산 처리 유닛(102_n)은, PE(Processing Element)(110) 및 메모리(120)로 구성되어 있다.

[0130] PE(110)는, FIFO 메모리(111, 112) 및 연산부(113)를 갖고 있다. 이들 FIFO 메모리(111, 112) 및 연산부(113)는 직렬로 접속되어 있다.

[0131] FIFO 메모리(111)는, 전단의 연산 처리 유닛(또는 루트 선택부)으로부터 데이터 버스를 통해 공급되는 프로세스 패킷(입력 프로세스 패킷)을 유지하기 위한 것이다. 이 FIFO 메모리(111)는, 거기에 공급되는 시스템 클럭에 동기하여, 입력 프로세스 패킷을 입력받아, 일시적으로 기억한다.

[0132] FIFO 메모리(112)는, 연산부(113)에 입력하는 프로세스 패킷을 보유하기 위한 것이다. FIFO 메모리(112)는, 거기에 공급되는 시스템 클럭에 동기하여, FIFO 메모리(111)로부터 순차적으로 출력되는 프로세스 패킷 PPa를 입력받아 일시적으로 기억한다.

[0133] 연산부(113)는, FIFO 메모리(112)로부터 순차적으로 출력되는 프로세스 패킷 PPb에 포함되는 명령에 따른 처리를 행하고, 처리 결과에 기초하여 상기 프로세스 패킷을 변경하여 출력할 프로세스 패킷 PPc를 얻는 것이다.

또한, 이 연산부(113)는, 이 출력할 프로세스 패킷 PPc를 출력할 때 하나의 상태, 예를 들면 「1」로 되며, 그 밖일 때 다른 상태, 예를 들면 「0」으로 되는 요구 신호 s_2를 출력한다.

[0134] 여기서, 연산부(113)는, 적어도, 프로세스 패킷에 포함되는 명령의 디코드, 디코드된 명령의 실행, 출력할 프로세스 패킷의 생성, 프로세스 패킷의 출력의 기능을 갖는다.

[0135] 연산부(113)는, 입력 프로세스 패킷 PPb에 포함되는 명령을 디코드하며, 그 명령의 실행이 가능하면, 그 명령을 실행한다. 그리고, 연산부(113)는, 명령을 실행함으로써 행한 처리의 결과에 기초하여 입력 프로세스 패킷 PPb를 변경하여 출력할 프로세스 패킷 PPc를 얻고, 그 프로세스 패킷 PPc를 출력한다.

[0136] 또한, 연산부(113)는, 명령의 실행이 불가능하면, 입력 프로세스 패킷 PPb를 그대로 출력할 프로세스 패킷 PPc로 하고, 그 프로세스 패킷 PPc를 출력한다.

[0137] 또한, 연산부(113)는, 입력 프로세스 패킷 PPb에 포함되는 명령의 실행은 가능하지만, 그 입력 프로세스 패킷 PPb와 동일한 프로세스 패킷이 과거에 입력되어 있고, 그것에 의한 처리를 종료하고 있을 때에는, 그 입력 프로세스 패킷 PPb에 관해서는 어떠한 처리도 행하지 않는다. 이 경우의 입력 프로세스 패킷 PPb는 불필요한 프로세스 패킷이기 때문이다.

[0138] 메모리(120)는, 연산부(113)와 데이터 버스를 통해 접속되어 있다. 연산부(113)는, 필요에 따라, 메모리(120)에 화상 데이터를 기억하거나, 혹은 메모리(120)로부터 화상 데이터를 판독한다. 즉, 프로세스 패킷에 배치되는 명령 중에는, 화상 데이터의 기입을 지시하는 기입 명령과, 화상 데이터의 판독을 지시하는 판독 명령이 있다.

[0139] 연산부(113)는, 프로세스 패킷에, 기입 명령과 화상 데이터가 배치되어 있는 경우에는, 그 화상 데이터를, 데이터 버스를 통해, 메모리(120)에 공급하고 기입한다. 또한, 연산부(113)는, 프로세스 패킷에, 판독 명령이 배치되어 있는 경우에는, 메모리(120)로부터, 데이터 버스를 통해, 화상 데이터를 판독하여, 프로세스 패킷에 배치한다.

[0140] 여기서, 본 실시 형태에서는, 도 9에 도시한 바와 같이, 메모리(120)는, 그 기억 영역이 5개의 뱅크로 분할되어 있으며, 이에 따라, 5 필드의 화상 데이터를, 별개의 뱅크에 기억할 수 있다. 단, 메모리(120)의 각 뱅크는, 1 필드분의 화상 데이터를 기억할 수 있는 만큼의 기억 용량을 갖고 있을 필요는 없다.

[0141] 본 실시 형태에서는, 메모리(120)의 각 뱅크는, 12개의 연산 처리 유닛(102_1~102_{12})의 합계로, 적어도 탐사 범위의 화상 데이터를 기억할 수 있는 기억 용량을 갖고 있으면 된다. 즉, 본 실시 형태에서는, 1 뱅크의 기억 용량의 12배의 기억 용량이, 탐사 범위의 화상 데이터의 데이터량 이상이면 된다. 예를 들면, 메모리(120)의 1 뱅크는, 1/12 필드분의 화상 데이터를 기억할 수 있는 기억 용량으로 된다. 따라서, 메모리(120)로서는, 도 3으로 설명한 바와 같은 배선 용량이 문제로 되지 않는 메모리, 즉, 그다지 기억 용량이 크지 않아서, PE(110) 사이의 데이터 버스를 짧게 할 수 있는 메모리를 채용할 수 있다.

[0142] 또한, PE(110)는, 어드레스 기억부(114) 및 어드레스 판정부(115)를 갖고 있다. 어드레스 기억부(114)에는, 미리, 연산 처리 유닛(102_n) 자체의 어드레스, 즉 자신의 어드레스가 기억되어 있다.

[0143] 어드레스 판정부(115)는, FIFO 메모리(111)로부터 출력되는 프로세스 패킷 PPa에 따른 프로세스가, 이 연산 처리 유닛(102_n)에 의해 실행되는 것인지 여부를 판정한다. 즉, 어드레스 판정부(115)는, FIFO 메모리(111)로부터 출력되는 프로세스 패킷 PPa의 처리 유닛 어드레스부(도 6 참조)에 기억되어 있는 1개 또는 복수개의 어드레스에, 어드레스 기억부(114)에 기억되어 있는 자신의 어드레스가 존재하는지의 여부를 판정하고, 존재할 때 하나의 상태, 예를 들면 「1」로 되며, 존재하지 않을 때 다른 상태, 예를 들면 「0」으로 되는 판정 신호 s_1을 출력한다.

[0144] 또한, PE(110)는, FIFO 메모리(111)로부터 출력되는 프로세스 패킷 PPa 또는 연산부(113)로부터 출력되는 프로세스 패킷 PPc를 선택적으로 취출하는, 절환 수단으로서의 절환 스위치(116)와, 이 절환 스위치(116)의 전환을 제어하는 절환 제어부(117)를 갖고 있다.

[0145] 연산부(113)의 출력측은 절환 스위치(116)의 A측의 고정 단자에 접속되며, FIFO 메모리(111)의 출력측은 시간 조정용 지연 회로(118)를 통해 절환 스위치(116)의 B측의 고정 단자에 접속된다. 또한, 절환 스위치(116)의 OFF측의 고정 단자는 전원 단자에 접속된다. 즉, 이 OFF측의 고정 단자에는, 하이 레벨 「1」의 데이터가 공급

된 상태로 된다. 또한, 이 OFF측의 고정 단자를 접지하여, 그것에 로우 레벨 「0」의 데이터가 공급된 상태로 하여도 된다.

[0146] 절환 스위치(116)의 가동 단자가 연산 처리 유닛(102_n)의 출력 단자로 된다. 즉, 이 가동 단자가 데이터 버스를 통해 후단의 연산 처리 유닛(또는 루트 선택부, 또는 집계부)에 접속된다.

[0147] 전술한 바와 같이, 어드레스 판정부(115)에서는 FIFO 메모리(111)로부터 출력되는 프로세스 패킷 PPa의 처리 유닛 어드레스부에 기억되어 있는 어드레스에 기초하여, 상기 프로세스 패킷 PPa에 따른 프로세스가 이 연산 처리 유닛(102_n)에 의해 실행되는 것인지의 여부를 판정한다.

[0148] 그리고, 후술하는 바와 같이, 상기 프로세스 패킷 PPa가 이 연산 처리 유닛(102_n)에 의해 실행되는 것이 아닐 때, 상기 프로세스 패킷 PPa를 절환 스위치(116)를 통해 즉시 후단에 출력하는 것이지만, 적어도 판정 시에는 상기 프로세스 패킷 PPa의 처리 유닛 어드레스부의 부분은 이미 FIFO 메모리(111)로부터 출력된 상태에 있어서, FIFO 메모리(111)의 출력측을 절환 스위치(116)의 B측의 고정 단자에 직접 접속하였다면, 상기 프로세스 패킷 PPa를 빠뜨리지 않고 완전한 상태로 후단에 공급할 수 없다.

[0149] 전술한 시간 조정용 지연 회로(118)는, 이 상기 프로세스 패킷 PPa를 빠뜨리지 않고 완전한 상태로 후단에 공급하기 위한 것이며, 적어도, FIFO 메모리(111)로부터 상기 프로세스 패킷 PPa의 출력이 개시되고 나서 어드레스 판정부(115)에서 판정 결과가 나오기까지의 지연 시간을 갖도록 된다.

[0150] 절환 제어부(117)에는, 연산부(113)로부터 출력되는 요구 신호 s_2 및 어드레스 판정부(115)로부터 출력되는 판정 신호 s_1이 공급된다. 그리고, 절환 제어부(117)는, 요구 신호 s_2가 「1」일 때에는, 판정 신호 s_1이 「1」인지 「0」인지에 상관없이, 절환 스위치(116)가 A측에 접속되도록 제어한다. 또한, 절환 제어부(117)는, 요구 신호 s_2가 「0」이며, 또한 판정 신호 s_1이 「0」일 때에는, 절환 스위치(116)가 B측에 접속되도록 제어한다. 또한, 절환 제어부(117)는, 요구 신호 s_2가 「0」이며, 또한 판정 신호 s_1이 「1」일 때에는, 절환 스위치(116)가 OFF측에 접속되도록 제어한다.

[0151] 도 10은, 판정 신호 s_1및 요구 신호 s_2와, 절환 스위치(116)의 전환 관계를 나타내고 있다.

[0152] 도 9에 도시하는 연산 처리 유닛(102_n)의 동작을 설명한다.

[0153] 전단의 연산 처리 유닛(또는 루트 선택부)으로부터 데이터 버스를 통해 공급되는 프로세스 패킷(입력 프로세스 패킷)은, FIFO 메모리(111)에 입력되어서, 일시적으로 기억된다. 이 FIFO 메모리(111)로부터 출력되는 프로세스 패킷 PPa는, FIFO 메모리(112)에 입력되어서, 일시적으로 기억된다. 또한, FIFO 메모리(111)로부터 출력되는 프로세스 패킷 PPa는, 지연 회로(118)를 통해 절환 스위치(116)의 B측의 고정 단자에 공급된다.

[0154] FIFO 메모리(112)로부터 순차적으로 출력되는 프로세스 패킷 PPb는, 연산부(113)에 공급된다. 연산부(113)는, 프로세스 패킷 PPb에 포함되는 명령을 디코드하고, 그 명령의 실행이 가능하면 그 명령을 실행한다. 그리고, 연산부(113)는, 명령을 실행함으로써 행한 처리의 결과에 기초하여 입력 프로세스 패킷 PPb를 변경하여 출력할 프로세스 패킷 PPc를 생성하고, 그 프로세스 패킷 PPc를 출력한다. 이 프로세스 패킷 PPc는 절환 스위치(116)의 A측의 고정 단자에 공급된다.

[0155] 단, 연산부(113)는, 입력 프로세스 패킷 PPb에 포함되는 명령의 실행은 가능하지만, 그 입력 프로세스 패킷 PPb와 동일한 프로세스 패킷이 과거에 입력되어 있으며, 그것에 의한 처리를 종료하고 있을 때에는, 그 입력 프로세스 패킷 PPb에 관해서는 어떠한 처리도 행하지 않는다. 동일한 프로세스 패킷인지 여부는, PID부(도 6 참조)에 배치된 PID에 기초하여 행할 수 있다.

[0156] 한편, 연산부(113)는, 그 명령의 실행이 불가능할 때에는, 입력 프로세스 패킷 PPb를 그대로 출력할 프로세스 패킷 PPc로 하고, 그 프로세스 패킷 PPc를 출력한다.

[0157] 연산부(113)로부터 출력되는 프로세스 패킷 PPc는, 절환 스위치(116)의 A측의 고정 단자에 공급된다. 연산부(113)는, 이와 같이 프로세스 패킷 PPc를 출력할 때, 요구 신호 s_2를 「1」로 한다.

[0158] 또한, 어드레스 판정부(115)는, 프로세스 패킷 PPa의 처리 유닛 어드레스부에 기억되어 있는 어드레스에 어드레스 기억부(114)에 기억되어 있는 자신의 어드레스가 존재하는지의 여부를 판정한다.

[0159] 이 경우, 어드레스 판정부(115)는, 처리 유닛 어드레스부에 기억되어 있는 어드레스에 자신의 어드레스가 존재할 때, 프로세스 패킷 PPa에 따른 프로세스가 이 연산 처리 유닛(102_n)에 의해 실행되는 것으로 하고, 판정 신호 s_1을 「1」로 한다. 한편, 어드레스 판정부(115)는, 처리 유닛 어드레스부에 기억되어 있는 어드레스에 자신의 어드레스가 존재하지 않을 때, 프로세스 패킷 PPa에 따른 프로세스가 이 연산 처리 유닛(102_n)에 의해 실행되는 것이 아니라고 하고, 판정 신호 s_1을 「0」으로 한다.

[0160] 연산부(113)로부터 출력되는 요구 신호 s_2 및 어드레스 판정부(115)로부터 출력되는 판정 신호 s_1은, 절환 제어부(117)에 공급된다. 절환 제어부(117)는, 요구 신호 s_2가 「1」일 때, 절환 스위치(116)를 A측에 접속한다. 즉, 연산부(113)가 프로세스 패킷 PPb의 처리를 종료하고 프로세스 패킷 PPc를 출력할 때, 절환 스위치(116)는 A측에 접속된다. 이 경우, 연산부(113)로부터 출력되는 프로세스 패킷 PPc는, 절환 스위치(116)의 A측을 통해 후단의 연산 처리 유닛(또는 루트 선택부, 또는 집계부)에, 출력 프로세스 패킷으로서 공급된다.

[0161] 또한, 절환 제어부(117)는, 요구 신호 s_2가 「0」이며, 또한 판정 신호 s_1이 「0」일 때, 절환 스위치(116)를 B측에 접속한다. 즉, 연산부(113)가 프로세스 패킷 PPb의 처리를 행할 때, 또한 FIFO 메모리(111)로부터 출력되는 프로세스 패킷 PPa에 따른 프로세스가 이 연산 처리 유닛(102_n)에 의해 실행되는 것이 아닐 때, 절환 스위치(116)는 B측에 접속된다. 이 경우, 그 프로세스 패킷 PPa는, 즉시 절환 스위치(116)의 B측을 통해, 후단의 연산 처리 유닛(또는 루트 선택부, 또는 집계부)에 출력 프로세스 패킷으로서 공급된다. 이에 따라, 처리 시간을 단축할 수 있다.

[0162] 또한, 절환 제어부(117)는, 요구 신호 s_2가 「0」이며, 또한 판정 신호 s_1이 「1」일 때에는, 절환 스위치(116)를 OFF측에 접속한다. 즉, 연산부(113)가 프로세스 패킷 PPb의 처리를 행할 때, 또한 FIFO 메모리(111)로부터 출력되는 프로세스 패킷 PPa에 따른 프로세스가 이 연산 처리 유닛(102_n)에 의해 실행되는 것일 때, 절환 스위치(116)는 OFF측에 접속된다. 이 경우, 후단의 연산 처리 유닛(또는 루트 선택부, 또는 집계부)에는, 하이 레벨 「1」 또는 로우 레벨 「0」의 데이터가 공급된다. 이에 따라, 후단의 연산 처리 유닛(또는 루트 선택부, 또는 집계부)에, 잘못된 프로세스 패킷, 예를 들면 FIFO 메모리(111)로부터 출력되는 프로세스 패킷 PPa(이 패킷 PPa는 이 연산 처리 유닛(102_n)의 연산부(113)에 의해 처리할 필요가 있으나, 그 처리를 거치고 있지 않음)이 공급되는 것을 방지할 수 있다.

[0163] 다음으로, 도 11의 흐름도를 참조하여, 도 9의 연산 처리 유닛(102_n)의 연산부(113)에 프로세스 패킷 PPb가 입력된 경우에서의, 상기 연산부(113)가 행하는 처리(프로세스 실행 처리)에 대하여 설명한다.

[0164] 단계 S11에서, 연산부(113)는, 그 입력 프로세스 패킷 PPb에 배치된 명령을 디코드하고, 단계 S12로 진행한다. 단계 S12에서는, 연산부(113)는, 단계 S11에서 디코드한 명령의 실행이 가능한지 여부를 판정한다.

[0165] 연산부(113)는, 단계 S12에서 명령의 실행이 가능하지 않다고 판정할 때, 단계 S13~S15를 스킵하고, 단계 S16으로 진행하며, 요구 신호 s_2를 「1」로 한다. 그리고, 연산부(113)는, 단계 S17에서, 입력 프로세스 패킷 PPb를 그대로 출력할 프로세스 패킷 PPc로서 출력하고, 그 후 단계 S18에서, 요구 신호 s_2를 「0」으로 되돌리고, 처리를 종료한다.

[0166] 또한, 연산부(113)는, 단계 S12에서 명령의 실행이 가능하다고 판정할 때, 단계 S13으로 진행하고, 그 입력 프로세스 패킷 PPb와 동일한 프로세스 패킷이 과거에 입력되어, 처리 완료되었는지의 여부를 판정한다. 연산부(113)는 처리 완료라고 판정할 때, 그 입력 프로세스 패킷 PPb에 관해서는 어떠한 처리도 행하지 않고, 처리를 종료한다. 이 경우, 입력 프로세스 패킷 PPb는 불필요한 프로세스 패킷이므로, 연산부(113)는, 이 입력 프로세스 패킷 PPb에 대응한 프로세스 패킷을 출력하지 않는다.

[0167] 또한, 연산부(113)는, 단계 S13에서 처리 완료가 아니라고 판정할 때, 단계 S14로 진행하여, 입력 프로세스 패킷 PPb에 배치된 명령을 실행하고, 그 명령에 따른 처리를 행한다. 그리고, 연산부(113)는, 단계 S14에서, 프로세스 패킷에 배치된 명령에 따른 처리를 행함으로써 얻어진 데이터를, 필요에 따라, 입력 프로세스 패킷 PPb에 배치하고, 단계 S15로 진행한다.

[0168] 단계 S15에서는, 연산부(113)는, 단계 S14에서 행한 처리에 대응하여, 필요에 따라, 입력 프로세스 패킷 PPb의 상태부를 재기입하고, 단계 S16으로 진행한다.

[0169] 단계 S16에서는, 연산부(113)는, 요구 신호 s_2를 「1」로 한다. 그리고, 연산부(113)는, 단계 S17에서, 단계 S14, S15에서 변경된 입력 프로세스 패킷 PPb를 출력할 프로세스 패킷 PPc로서 출력하고, 그 후 단계 S18에서, 요구 신호 s_2를 「0」으로 되돌리고, 처리를 종료한다.

[0170] 다음으로, 도 12의 흐름도를 참조하여, 도 9의 연산 처리 유닛(102_n)의 연산부(113)에 기입 명령을 갖는 프로세스 패킷 PPb가 입력된 경우에서의, 상기 연산부(113)가 행하는 처리(프로세스 실행 처리)에 대하여 설명한다.

[0171] 또한, 이 경우, 입력 프로세스 패킷 PPb에는, 기입 명령 외, 메모리(120)에 기입할 화상 데이터와, 그 화상 데이터를 메모리(120)에 기입하는 기입 어드레스가, 적어도 배치되어 있는 것으로 한다.

[0172] 단계 S21에서, 연산부(113)는, 그 입력 프로세스 패킷 PPb에 배치된 명령을 디코드하고, 단계 S22로 진행한다. 단계 S22에서는, 연산부(113)는, 단계 S21에서 디코드한 명령의 실행이 가능한지 여부, 여기서는, 입력 프로세스 패킷 PPb에 배치되어 있는 기입 어드레스가, 연산 처리 유닛(102_n)이 갖는 메모리(120_n)의 어드레스인지 여부를 판정한다.

[0173] 연산부(113)는, 단계 S22에서 프로세스 패킷에 배치되어 있는 기입 어드레스가, 연산 처리 유닛(102_n)이 갖는 메모리(120_n)의 어드레스가 아니라고 판정할 때, 단계 S23~S25를 스킵하고, 단계 S26으로 진행하며, 요구 신호 s_2를 「1」로 한다. 그리고, 연산부(113)는, 단계 S27에서, 입력 프로세스 패킷 PPb를 그대로 출력할 프로세스 패킷 PPc로서 출력하고, 그 후 단계 S28에서, 요구 신호 s_2를 「0」으로 반환하며, 처리를 종료한다.

[0174] 또한, 연산부(113)는, 단계 S22에서 프로세스 패킷에 배치되어 있는 기입 어드레스가, 연산 처리 유닛(102_n)이 갖는 메모리(120_n)의 어드레스라고 판정할 때, 단계 S23으로 진행하고, 그 입력 프로세스 패킷 PPb와 동일한 프로세스 패킷이 과거에 입력되어, 처리 완료되었는지의 여부를 판정한다. 연산부(113)는 처리 완료라고 판정할 때, 그 입력 프로세스 패킷 PPb에 관해서는 어떠한 처리도 행하지 않고, 처리를 종료한다. 이 경우, 입력 프로세스 패킷 PPb는 불필요한 프로세스 패킷이므로, 연산부(113)는, 이 입력 프로세스 패킷 PPb에 대응한 프로세스 패킷을 출력하지 않는다.

[0175] 또한, 연산부(113)는, 단계 S23에서 처리 완료가 아니라고 판정할 때, 단계 S24로 진행하여, 입력 프로세스 패킷 PPb에 배치된 명령을 실행한다. 즉, 연산부(113)는, 입력 프로세스 패킷 PPb에 배치되어 있는 화상 데이터를 메모리(120_n)에 기입하고, 단계 S25로 진행한다.

[0176] 단계 S25에서는, 연산부(113)는, 단계 S24에서 행한 처리에 대응하여, 필요에 따라, 입력 프로세스 패킷 PPb의 상태부를 재기입하고, 단계 S26으로 진행한다.

[0177] 단계 S26에서는, 연산부(113)는, 요구 신호 s_2를 「1」로 한다. 그리고, 연산부(113)는, 단계 S27에서, 단계 S24, S25에서 변경된 입력 프로세스 패킷 PPb를 출력할 프로세스 패킷 PPc로서 출력하고, 그 후 단계 S28에서, 요구 신호 s_2를 「0」으로 되돌리고, 처리를 종료한다.

[0178] 프로세스 생성부(101)는, 도 7의 프로세스 생성 처리의 단계 S1~S3에서, 기입 명령을 갖는 프로세스 패킷(이하, 적절하게, 「기입 프로세스 패킷」이라 함)을 생성하여 출력한다. 이 기입 프로세스 패킷이, 연산 처리 유닛(102_1~102_{12})을 순차적으로 이동해감에 따라, 예를 들면 도 13a 및 도 13b에 도시한 바와 같이, 적어도, 주목 블록과 후보 블록의 화상 데이터가, 메모리(120)에 기입된다.

[0179] 여기서, 도 13a에 도시한 바와 같이, 주목 블록의 화상 데이터를 크로스 해칭을 하여 나타냄과 함께, 후보 블록의 화상 데이터를 해칭을 하여 나타내는 것으로 하면, 도 13b에서는, 주목 블록의 일부가 메모리(120_1)에, 다른 일부가 메모리(120_2)에, 또 다른 일부가 메모리(120_5)에, 나머지가 메모리(120_6)에, 각각 기입되고, 후보 블록의 일부가 메모리(120_3)에, 다른 일부가 메모리(120_4)에, 또 다른 일부가 메모리(120_7)에, 나머지가 메모리(120_8)에, 각각 기입되어 있다.

[0180] 다음으로, 도 14의 흐름도를 참조하여, 도 9의 연산 처리 유닛(102_n)의 연산부(113)에 판독 명령을 갖는 프로세스 패킷 PPb가 입력된 경우에서의, 상기 연산부(113)가 행하는 처리(프로세스 실행 처리)에 대하여 설명한다.

[0181] 또한, 이 경우, 입력 프로세스 패킷 PPb에는, 판독 명령 외, 메모리(120)로부터 판독할 화상 데이터의 판독 어드레스가, 적어도 배치되어 있는 것으로 한다.

[0182] 단계 S31에서, 연산부(113)는, 그 입력 프로세스 패킷 PPb에 배치된 명령을 디코드하고, 단계 S32로 진행한다. 단계 S32에서는, 연산부(113)는, 단계 S31에서 디코드한 명령의 실행이 가능한지 여부, 여기서는, 입력 프로세스 패킷 PPb에 배치되어 있는 판독 어드레스가, 연산 처리 유닛(102_n)이 갖는 메모리(120_n)의 어드레스인지 여부를 판정한다.

[0183] 연산부(113)는, 단계 S32에서 프로세스 패킷에 배치되어 있는 판독 어드레스가, 연산 처리 유닛(102_n)이 갖는 메모리(120_n)의 어드레스가 아니라고 판정할 때, 단계 S33~S35를 스킵하고, 단계 S36으로 진행하며, 요구 신호 s_2를 「1」로 한다. 그리고, 연산부(113)는, 단계 S37에서, 입력 프로세스 패킷 PPb를 그대로 출력할 프로세스 패킷 PPc로서 출력하고, 그 후 단계 S38에서, 요구 신호 s_2를 「0」으로 되돌리고, 처리를 종료한다.

[0184] 또한, 연산부(113)는, 단계 S32에서 프로세스 패킷에 배치되어 있는 판독 어드레스가, 연산 처리 유닛(102_n)이 갖는 메모리(120_n)의 어드레스라고 판정할 때, 단계 S33으로 진행하고, 그 입력 프로세스 패킷 PPb와 동일한 프로세스 패킷이 과거에 입력되어, 처리 완료되었는지의 여부를 판정한다. 연산부(113)는 처리 완료라고 판정할 때, 그 입력 프로세스 패킷 PPb에 관해서는 어떠한 처리도 행하지 않고, 처리를 종료한다. 이 경우, 입력 프로세스 패킷 PPb는 불필요한 프로세스 패킷이므로, 연산부(113)는, 이 입력 프로세스 패킷 PPb에 대응한 프로세스 패킷을 출력하지 않는다.

[0185] 또한, 연산부(113)는, 단계 S33에서 처리 완료가 아니라고 판정할 때, 단계 S34로 진행하여, 입력 프로세스 패킷 PPb에 배치된 명령을 실행한다. 즉, 연산부(113)는, 메모리(120_n)로부터 화상 데이터를 판독하고, 그 화상 데이터를 입력 프로세스 패킷 PPb에 배치하고, 단계 S35로 진행한다.

[0186] 단계 S35에서, 연산부(113)는, 단계 S34에서 행한 처리에 대응하여, 필요에 따라, 입력 프로세스 패킷 PPb의 상태부를 재기입하고, 단계 S36으로 진행한다.

[0187] 단계 S36에서는, 연산부(113)는, 요구 신호 s_2를 「1」로 한다. 그리고, 연산부(113)는, 단계 S37에서, 단계 S34, S35에서 변경된 입력 프로세스 패킷 PPb를 출력할 프로세스 패킷 PPc로서 출력하고, 그 후 단계 S38에서, 요구 신호 s_2를 「0」으로 되돌리고, 처리를 종료한다.

[0188] 다음으로, 도 15의 흐름도를 참조하여, 도 9의 연산 처리 유닛(102_n)의 연산부(113)에 차분 절대값의 합 연산 명령을 갖는 프로세스 패킷 PPb가 입력된 경우에서의, 상기 연산부(113)가 행하는 처리(프로세스 실행 처리)에 대하여 설명한다.

[0189] 또한, 이 경우, 메모리(120)에는, 이미, 차분 절대값의 합의 연산 대상인 주목 블록과 후보 블록의 화상 데이터가 적어도 기입되어 있는 것으로 한다. 또한, 입력 프로세스 패킷 PPb에는, 차분 절대값의 합 연산 명령 외에, 메모리(120)에 기억된 주목 블록과 후보 블록의 어드레스, 후보 블록으로부터 주목 블록을 향한 후보 벡터가, 적어도 배치되어 있는 것으로 한다.

[0190] 단계 S41에서, 연산부(113)는, 그 입력 프로세스 패킷 PPb에 배치된 명령을 디코드하고, 단계 S42로 진행한다. 단계 S42에서는, 연산부(113)는, 단계 S41에서 디코드한 명령의 실행이 가능한지 여부, 여기서는, 주목 블록 또는 후보 블록 중 적어도 한쪽 화소가, 연산 처리 유닛(102_n)이 갖는 메모리(120_n)에 기억되어 있는지 여부를 판정한다.

[0191] 여기서, 주목 블록의 화소가, 연산 처리 유닛(102_n)이 갖는 메모리(120)에 기억되어 있는지 여부는, 입력 프로세스 패킷 PPb에 배치된 주목 블록의 어드레스로부터 판정할 수 있다. 후보 블록의 화소가, 연산 처리 유닛(102_n)이 갖는 메모리(120)에 기억되어 있는지 여부도, 입력 프로세스 패킷 PPb에 배치된 후보 블록의 어드레스로부터 판정할 수 있다.

[0192] 연산부(113)는, 단계 S42에서 주목 블록 및 후보 블록 모두, 연산 처리 유닛(102_n)이 갖는 메모리(120_n)에 기억

되어 있지 않다고 판정할 때, 단계 S43~S45를 스킵하고, 단계 S46으로 진행하며, 요구 신호 s_2를 「1」로 한다. 그리고, 연산부(113)는, 단계 S47에서, 입력 프로세스 패킷 PPb를 그대로 출력할 프로세스 패킷 PPc로서 출력하고, 그 후 단계 S48에서, 요구 신호 s_2를 「0」으로 되돌리고, 처리를 종료한다.

[0193] 또한, 연산부(113)는, 단계 S42에서 주목 블록 또는 후보 블록 중 적어도 한쪽 화소가, 연산 처리 유닛(102_n)이 갖는 메모리(120_n)에 기억되어 있다고 판정할 때, 단계 S43으로 진행하고, 그 입력 프로세스 패킷 PPb와 동일한 프로세스 패킷이 과거에 입력되어, 처리 완료되었는지의 여부를 판정한다. 연산부(113)는 처리 완료라고 판정할 때, 그 입력 프로세스 패킷 PPb에 관해서는 어떠한 처리도 행하지 않고, 처리를 종료한다. 이 경우, 입력 프로세스 패킷 PPb는 불필요한 프로세스 패킷이므로, 연산부(113)는, 이 입력 프로세스 패킷 PPb에 대응한 프로세스 패킷을 출력하지는 않는다.

[0194] 또한, 연산부(113)는, 단계 S43에서 처리 완료가 아니라고 판정할 때, 단계 S44로 진행하고, 입력 프로세스 패킷 PPb에 배치된 명령을 실행한다.

[0195] 즉, 먼저 처음에, 단계 S51에서, 연산부(113)는, 주목 블록의 화소가, 연산 처리 유닛(102_n)이 갖는 메모리(120_n)에 기억되어 있는지 여부를 판정한다. 연산부(113)는, 단계 S51에서 주목 블록의 화소가 연산 처리 유닛(102_n)이 갖는 메모리(120_n)에 기억되어 있지 않다고 판정할 때, 단계 S52를 스킵하고, 단계 S53으로 진행한다.

[0196] 한편, 연산부(113)는, 단계 S51에서 주목 블록의 화소가 연산 처리 유닛(102_n)이 갖는 메모리(120_n)에 기억되어 있다고 판정할 때, 단계 S52로 진행하고, 메모리(120_n)에 기억되어 있는 주목 블록의 화소를 판독하여, 입력 프로세스 패킷 PPb에 배치하고, 단계 S53으로 진행한다.

[0197] 단계 S53에서는 연산부(113)는, 후보 블록의 화소가 연산 처리 유닛(102_n)이 갖는 메모리(120_n)에 기억되어 있는지 여부를 판정한다. 연산부(113)는, 단계 S53에서 후보 블록의 화소가 연산 처리 유닛(102_n)이 갖는 메모리(120_n)에 기억되어 있지 않다고 판정할 때, 단계 S54를 스킵하고, 단계 S55로 진행한다.

[0198] 한편, 연산부(113)는, 단계 S53에서 후보 블록의 화소가 연산 처리 유닛(102_n)이 갖는 메모리(120_n)에 기억되어 있다고 판정할 때, 단계 S54로 진행하고, 메모리(120_n)에 기억되어 있는 후보 블록의 화소를 판독하여, 입력 프로세스 패킷 PPb에 배치하여, 단계 S55로 진행한다.

[0199] 단계 S55에서는, 연산부(113)는, 차분 절대값의 합의 계산이 가능한지 여부를 판정한다. 이 경우, 연산부(113)는, 입력 프로세스 패킷 PPb에, 주목 블록의 화소가 배치되어 있고, 또한, 그 주목 블록의 화소에 대응하는 후보 블록의 화소도 배치되어 있는지 여부에 따라, 차분 절대값의 합의 계산이 가능한지 여부를 판정한다.

[0200] 연산부(113)는, 단계 S55에서 차분 절대값의 합의 계산이 가능하지 않다고 판정할 때, 즉 입력 프로세스 패킷 PPb에, 주목 블록의 화소가 배치되어 있지 않은지, 또는 배치되어 있어도, 그 주목 블록의 화소에 대응하는 후보 블록의 화소가 배치되어 있지 않은 경우, 단계 S56을 스킵하며, 이에 따라, 차분 절대값의 합 연산 명령의 실행을 종료하고, 단계 S45로 진행한다.

[0201] 한편, 연산부(113)는, 단계 S55에서 차분 절대값의 합의 계산이 가능하다고 판정할 때, 즉 입력 프로세스 패킷 PPb에, 주목 블록의 화소가 배치되어 있고, 또한, 그 주목 블록의 화소에 대응하는 후보 블록의 화소도 배치되어 있는 경우, 단계 S56으로 진행하고, 입력 프로세스 패킷 PPb에 배치되어 있는 주목 블록의 화소 각각과, 그 화소에 대응하는 후보 블록의 화소 각각 간의 차분 절대값을 계산하고, 또한, 그 총합을 계산한다.

[0202] 그리고, 연산부(113)는, 그 차분 절대값의 총합과, 입력 프로세스 패킷 PPb의 차분 절대값의 합부에 배치되어 있는 차분 절대값의 합을 가산하고, 그 가산값을 새로운 차분 절대값의 합으로 하여 입력 프로세스 패킷 PPb의 차분 절대값의 합부에 덮어쓰기하고, 이에 의해, 차분 절대값의 합 연산 명령의 실행을 종료하고, 단계 S45로 진행한다.

[0203] 단계 S45에서는, 연산부(113)는, 단계 S44에서 행한 처리에 대응하여, 필요에 따라, 입력 프로세스 패킷 PPb의 상태부를 재기입하고, 단계 S46으로 진행한다.

[0204] 단계 S46에서는, 연산부(113)는, 요구 신호 s_2를 「1」로 한다. 그리고, 연산부(113)는, 단계 S47에서, 단계

S44, S45에서 변경된 입력 프로세스 패킷 PPb를 출력할 프로세스 패킷 PPc로서 출력하고, 그 후 단계 S48에서, 요구 신호 s_2를 「0」으로 되돌리고, 처리를 종료한다.

[0205] 프로세스 생성부(101)는, 도 7의 프로세스 생성 처리의 단계 S4～S7에서, 차분 절대값의 합 연산 명령을 갖는 프로세스 패킷(이하, 적절하게, 「차분 절대값의 합 프로세스 패킷」이라 함)을 생성하여 출력한다. 이 차분 절대값의 합 프로세스 패킷이, 연산 처리 유닛(102_1～102_{12})을 순차적으로 이동해감에 따라, 주목 블록과 후보 블록의 차분 절대값의 합이, 도 16 및 도 17a 및 도 17b에 도시한 바와 같이 구해진다.

[0206] 즉, 현재, 도 16에 도시한 바와 같이, 메모리(120_n)에, 주목 블록의 일부가, 메모리(120_{n+1})에, 주목 블록의 나머지와 후보 블록의 일부가, 메모리(120_{n+2})에, 후보 블록의 나머지가, 각각 기억되어 있는 것으로 한다. 또한, 연산 처리 유닛(102_n)의 PE(110)를, 이하, 적절하게, 「PE(110_n)」로 나타내기로 한다.

[0207] 이 경우, 연산 처리 유닛(102_n)이 프로세스 패킷을 수신하면, PE(110_n)는, 메모리(120_n)에 기억되어 있는 주목 블록의 일부를 판독하여, 프로세스 패킷에 배치하고, 다음 단의 연산 처리 유닛(102_{n+1})에 전송한다.

[0208] 여기서, PE(110_n)에서는, 도 17a 및 도 17b에 도시하는 프로세스 패킷이 송수신된다. 또한, 도 17a 및 도 17b에서는, 도면을 간단히 나타내기 위해, 프로세스 패킷에 대하여, 도 17a에 도시한 바와 같이, 그 선두로부터, PID부, 주목 블록 판독 상태부, 후보 블록 판독 상태부, 명령부, 주목 블록 어드레스부, 후보 블록 어드레스부, 주목 블록 데이터부, 후보 블록 데이터부, 차분 절대값의 합부만을 도시하고 있다.

[0209] 주목 블록 및 후보 블록은, 도 16에서 설명한 바와 같이, 메모리(120_n～120_{n+2})에 기억되어 있기 때문에, 프로세스 패킷이, PE(110_n)에서 수신되기 전에는, 주목 블록 및 후보 블록 중 어떠한 판독도 행해지지 않으며, 따라서, 주목 블록 판독 상태부와, 후보 블록 판독 상태부의 상태 정보는, 도 17b에 도시한 바와 같이, 모두 「미」로 되어 있다. 또한, 이 경우, 주목 블록 어드레스부와 후보 블록 어드레스부에는, 각각, 주목 블록의 어드레스 add1과 후보 블록의 어드레스 add2가 세트되어 있다. 또한, 차분 절대값의 합부에는, 초기값으로서의 0이 세트되어 있다. 또한, 명령부에는, 차분 절대값의 합 연산 명령을 나타내는 「ME」가 세트되어 있다.

[0210] 이상과 같은 프로세스 패킷이, PE(110_n)에 입력되며, 도 16으로 설명한 바와 같이 처리됨으로써, 다음과 같은 프로세스 패킷이, PE(110_n)로부터 PE(110_{n+1})에 전송된다.

[0211] 즉, PE(110_n)로부터 PE(110_{n+1})에 전송되는 프로세스 패킷에는, 도 17b에 도시한 바와 같이, 메모리(120_n)에 기억되어 있는 주목 블록의 일부의 화소 data_a1이, 새롭게 배치된다. 또한, PE(110_n)에서는, 메모리(120_n)로부터 주목 블록의 일부의 화소 data_a1이 판독되었기 때문에, 프로세스 패킷의 주목 블록 판독 상태부의 상태 정보가, 「미」로부터 「중」으로 재기입된다.

[0212] 연산 처리 유닛(102_{n+1})이, 연산 처리 유닛(102_n)으로부터의 프로세스 패킷을 수신하면, PE(110_{n+1})는, 도 16에 도시한 바와 같이, 메모리(120_{n+1})에 기억되어 있는 주목 블록의 나머지와, 후보 블록의 일부를 판독한다. 여기서, 연산 처리 유닛(102_n)으로부터의 프로세스 패킷에는, 주목 블록의 일부가 배치되어 있기 때문에, PE(110_{n+1})는, 메모리(120_{n+1})로부터 판독한 주목 블록의 나머지와 합쳐서, 주목 블록 전체의 화소를 취득하게 된다.

[0213] PE(110_{n+1})는, 주목 블록 전체의 화소와, 메모리(120_{n+1})로부터 판독한 후보 블록의 일부의 화소를 이용하여 계산 가능한 차분 절대값의 합을 구하고, 프로세스 패킷에 배치한다. 또한, PE(110_{n+1})는, 차분 절대값의 합의 연산에 이용되지 않은 주목 블록의 화소를, 프로세스 패킷에 배치하고, 다음 단의 연산 처리 유닛(102_{n+2}) 전송한다.

[0214] 즉, PE(110_{n+1})로부터 PE(110_{n+2})에 전송되는 프로세스 패킷에는, 도 17b에 도시한 바와 같이, 주목 블록의 화소 중, 차분 절대값의 합의 연산에 이용되지 않은 화소 data_a2가, 주목 블록 데이터부의 화소 data_a1 대신에 배치됨과 함께, PE(110_{n+1})에서 구해진 차분 절대값의 합 sum1이, 차분 절대값의 합부의 초기값 0 대신에 배치된다. 또한, PE(110_{n+1})에서는, 주목 블록의 화소의 전부가 취득됨과 함께, 후보 블록의 일부의 화소가 취득되었기 때문에, 프로세스 패킷의 주목 블록 판독 상태부의 상태 정보가 「중」으로부터 「종」으로 재기입됨

과 함께, 후보 블록 판독 상태부의 상태 정보는, 「미」로부터 「중」으로 재기입된다.

[0215] 연산 처리 유닛(102_{n+2})이, 연산 처리 유닛(102_{n+1})으로부터의 프로세스 패킷을 수신하면, PE(110_{n+2})는, 도 16에 도시한 바와 같이, 메모리(120_{n+2})에 기억되어 있는 후보 블록의 나머지를 판독한다.

[0216] PE(110_{n+2})는, 메모리(120_{n+2})로부터 판독한 후보 블록의 나머지 화소와, 프로세스 패킷에 배치되어 있는 주목 블록의 화소를 이용하여 계산 가능한 차분 절대값의 합을 구하고, 프로세스 패킷에 배치되어 있는 차분 절대값의 합과 가산한다. 그리고, PE(110_{n+2})는, 그 가산값을, 새로운 차분 절대값의 합으로 하여, 프로세스 패킷에 덮어쓰기하는 형태로 배치하고, 다음 단의 연산 처리 유닛(102_{n+3})(도시 생략)에 전송한다.

[0217] 즉, PE(110_{n+2})에서는, 주목 블록과 후보 블록의 모든 화소에 대한 차분 절대값의 합 sum2가 구해지므로, PE(110_{n+2})로부터 PE(110_{n+3})(도시 생략)에 전송되는 프로세스 패킷에는, 도 17b에 도시한 바와 같이, 그 차분 절대값의 합 sum2가, 차분 절대값의 합부의 차분 절대값의 합 sum1 대신에 배치된다. 또한, 프로세스 패킷에서 주목 블록 데이터부에 배치되어 있던 주목 블록의 화소 data_a2는, 모두, 차분 절대값의 합의 계산에 이용되므로, 주목 블록 데이터부로부터, 주목 블록의 화소 data_a2가 삭제된다. 또한, PE(110_{n+2})에서는, 후보 블록의 나머지 화소가 취득되었기 때문에, 프로세스 패킷의 후보 블록 판독 상태부의 상태 정보가, 「중」으로부터 「종」으로 재기입된다.

[0218] 이 경우, 프로세스 패킷이, PE(110_{n+2})로부터 출력되는 시점에서, 그 프로세스 패킷에는, 주목 블록과 후보 블록의 모든 화소에 대한 차분 절대값의 합 sum2가 배치되어 있다. 따라서, 연산 처리 유닛(102_{n+3}) 이후에는, 프로세스 패킷은, 단순히 전송되어가고, 최종적으로, 집계부(103)(도 5 참조)에서 수신된다.

[0219] 다음으로, 집계부(103)의 상세 내용을 설명한다. 집계부(103)도, 연산 처리 유닛(102_n)과 마찬가지로 구성되어 있다. 즉, 집계부(103)는, PE(110)과 메모리(120)로 구성되어 있다. 단, 집계부(103)는, 연산 처리 유닛(102_n)과 다른 구성으로 하는 것도 가능하지만, 연산 처리 유닛(102_n)과 동일한 구성으로 하는 편이, 움직임 벡터 검출 장치(100)의 제조 코스트를 낮게 억제할 수 있다.

[0220] 프로세스 생성부(101)에서는, 도 7의 단계 S4～S6의 처리에 의해, 주목 블록의 탐사 범위 내에서 선택할 수 있는 후보 벡터 전체에 대하여, 차분 절대값의 합을 연산하기 위한 차분 절대값의 합 연산 프로세스가 생성되는데, 이 프로세스에 대응하는 프로세스 패킷 전체가 집계부(103)에서 수신되면, 집계부(103)는, 주목 블록의 탐사 범위 내에서 선택할 수 있는 후보 벡터 전체에 대하여, 차분 절대값의 합을 취득한다.

[0221] 프로세스 생성부(101)에서는, 도 7의 단계 S4～S6의 처리 후, 단계 S7 및 S8에서, 최소값 판정 프로세스가 생성되고, 최소값 판정 명령을 포함하는 프로세스 패킷이 출력된다. 이 프로세스 패킷은, 연산 처리 유닛(102_1～102_{12})을 경유하여, 집계부(103)에서 수신된다. 집계부(103)는, 최소값 판정 명령을 포함하는 프로세스 패킷을 수신하면, 지금까지 수신한, 차분 절대값의 합 연산 프로세스에 대응하는 프로세스 패킷으로부터, 최소의 차분 절대값의 합이 배치되어 있는 것을 선택하고, 또한, 그 프로세스 패킷에 배치되어 있는 후보 벡터를, 주목 블록의 움직임 벡터로서 출력한다.

[0222] 이상 설명한 바와 같이, 전술한 실시 형태에서는, PE(110_n)와 메모리(120_n)로 이루어지는 복수의 연산 처리 유닛(102_n)을 일차원적으로 접속하고, 그 복수의 연산 처리 유닛(102_n)에서, 명령과 필요한 데이터가 배치된 프로세스 패킷을 전송(이동)시킴으로써, 움직임 벡터를 검출하기 위한 처리를 행하도록 하였기 때문에, 긴 데이터 버스를 구동할 때의 문제로서의, 예를 들면 배선 지연이나 배선 간의 크로스토크, 반사의 영향 등을 피할 수 있다.

[0223] 즉, PE(110_n)와 메모리(120_n) 간이나, 연산 처리 유닛(102_n)끼리 연결하는 데이터 버스는, 짧게 되어 있기 때문에, 긴 데이터 버스를 구동할 때의 문제를 피할 수 있다. 따라서, 움직임 벡터 검출 장치의 설계에 대하여, 긴 데이터 버스를 구동할 때의 문제를 고려할 필요가 없으므로, 설계 용이한 하드웨어에 의해, 움직임 벡터를 검출할 수 있다.

[0224] 또한, 전술한 실시 형태에서는, 12개의 연산 처리 유닛(102_1～102_{12})이 4개의 연속하는 연산 처리 유닛의 조로

나뉘며, 각 조(102a～102c)의 입력측에 각각 루트 선택부(104_1～104_3)가 삽입된다. 그리고, 루트 선택부(104_1～104_3) 각각은, 입력 프로세스 패킷에 따른 프로세스를 실행하는 연산 처리 유닛(102_n)이 직후에 위치하는 조(102a～102c)에 존재하지 않을 때에는, 상기 입력 프로세스 패킷을, 그 조(102a～102c)의 입력측에 공급하지 않고, 그 조의 출력측에 공급한다.

[0225] 따라서, 프로세스 패킷이, 그 프로세스 패킷에 따른 프로세스를 실행하기 위한 연산 처리 유닛(102_n)이 존재하지 않는 조를 바이패스하고 이동해가기 때문에, 처리 시간을 단축할 수 있으며, 또한 프로세스 패킷의 불필요한 이동이 없어지므로, 소비 전력을 저감할 수 있다.

[0226] 또한, 전술한 실시 형태에서, 연산 처리 유닛(102_n)(도 9 참조)에서는, 연산부(113)가 프로세스 패킷 PPb의 처리를 행할 때, 또한 FIFO 메모리(111)로부터 출력되는 프로세스 패킷 PPa에 따른 프로세스가 이 연산 처리 유닛(102_n)에 의해 실행되는 것이 아닐 때, 절환 스위치(116)는 B측에 접속된다. 따라서, 그 프로세스 패킷 PPa는, 즉시 절환 스위치(116)의 B측을 통해, 후단의 연산 처리 유닛(또는 루트 선택부, 또는 집계부)에 출력 프로세스 패킷으로서 공급되므로, 처리 시간을 단축할 수 있다.

[0227] 또한, 연산부(113)가 프로세스 패킷 PPb의 처리를 행할 때, 또한 FIFO 메모리(111)로부터 출력되는 프로세스 패킷 PPa에 따른 프로세스가 이 연산 처리 유닛(102_n)에 의해 실행되는 것일 때, 절환 스위치(116)는 OFF측에 접속된다. 이 경우, 후단의 연산 처리 유닛(또는 루트 선택부, 또는 집계부)에는, 하이 레벨 「1」 또는 로우 레벨 「0」의 데이터가 공급된다. 이에 따라, 후단에, 잘못된 프로세스 패킷, 예를 들면 FIFO 메모리(111)로부터 출력되는 프로세스 패킷 PPa가 공급되는 것을 방지할 수 있다.

[0228] 또한, 전술한 실시 형태에서는, 본 발명을 움직임 벡터 검출 장치에 적용한 것이지만, 본 발명은 화상 데이터를 취급하는 그 밖의 화상 처리를 행하는 것에도 마찬가지로 적용할 수 있다. 예를 들면, 전술한 움직임 벡터 검출 장치와 마찬가지로 화상 데이터의 차분 연산을 행하고, 그 차분 연산 결과에 기초하여 참조 필드에 소정의 화상 혹은 문자가 존재하는지를 검색하는 장치에도 양호하게 적용할 수 있다.

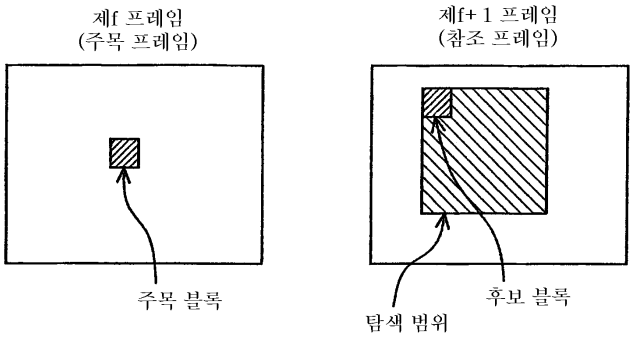
[0229] 또한, 상기에서 흐름도를 참조하여 설명한 처리 단계는, 반드시 흐름도로서 기재된 순서를 따라 시계열로 처리할 필요는 없으며, 병렬적으로 혹은 개별적으로 실행되는 처리(예를 들면, 병렬 처리 혹은 오브젝트에 의한 처리)도 포함하는 것이다.

[0230] 본 발명에 따르면, 화상 처리를 행하기 위한 각 프로세스를 실행하는 명령을 포함하는 프로세스 데이터를, 프로세스를 실행하는 직렬 접속된 복수개의 실행 수단을 이동시켜가며 화상 처리를 행하는 데 있어서, 소정수의 연속하는 실행 수단의 조마다, 그 입력측에 루트 선택 수단을 삽입하고, 루트 선택 수단은, 입력 프로세스 데이터에 따른 프로세스를 직후에 위치하는 조를 구성하는 실행 수단에 의해 실행하지 않을 때에는, 상기 입력 프로세스 데이터를 상기 직후에 위치하는 조의 출력측에 공급하는 것이고, 처리 시간의 단축 및 소비 전력의 저감을 도모할 수 있다.

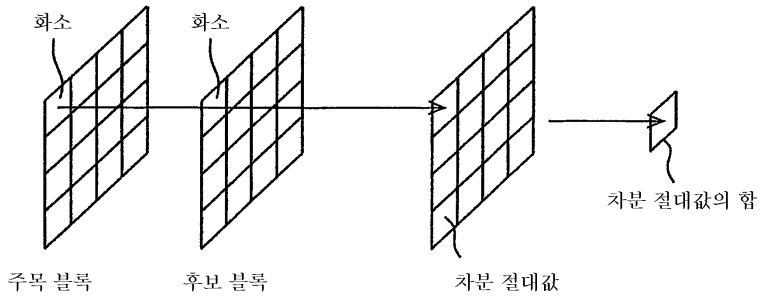
[0231] 또한, 본 발명에 따르면, 화상 처리를 행하기 위한 각 프로세스를 실행하는 명령을 포함하는 프로세스 데이터를, 프로세스를 실행하는 직렬 접속된 복수개의 실행 수단을 이동시켜가며 화상 처리를 행하는 데 있어서, 실행 수단은, 입력 프로세스 데이터에 따른 프로세스를 자기가 실행하지 않을 때에는, 상기 입력 프로세스 데이터를 그대로 출력 프로세스 데이터로서 출력하는 것이므로, 처리 시간의 단축을 도모할 수 있다.

도면

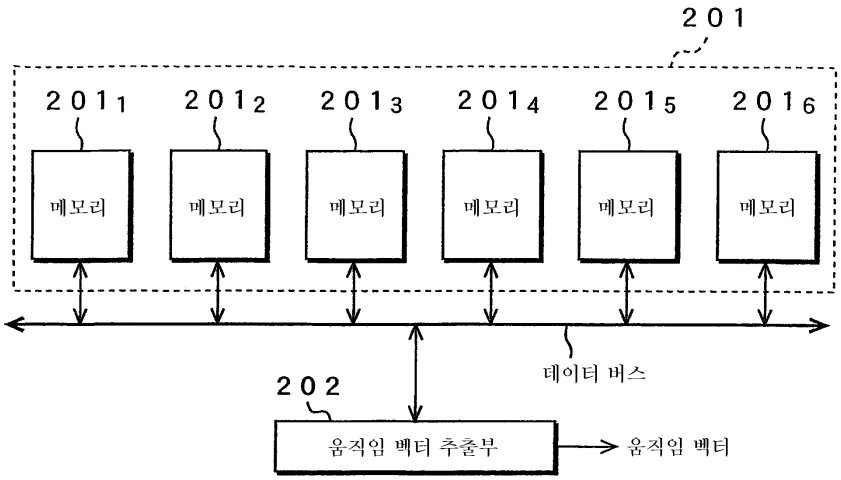
도면1



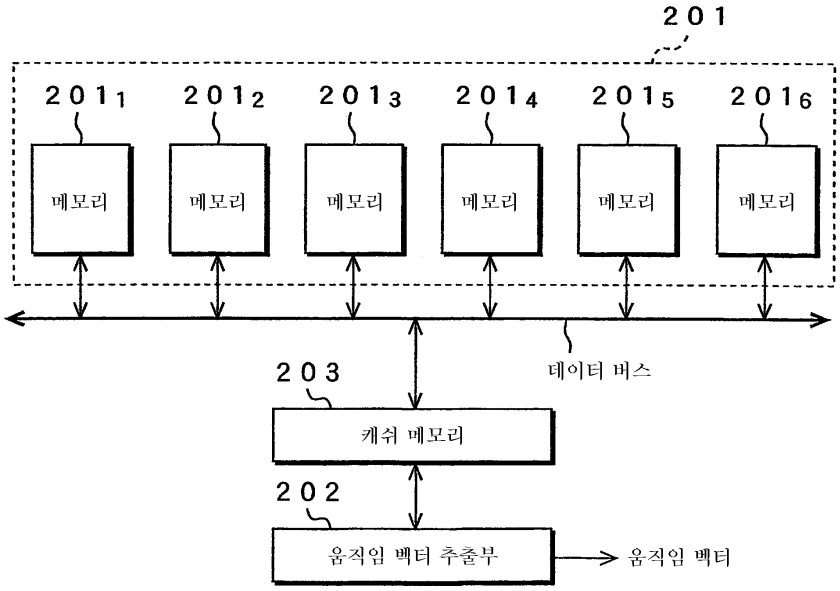
도면2



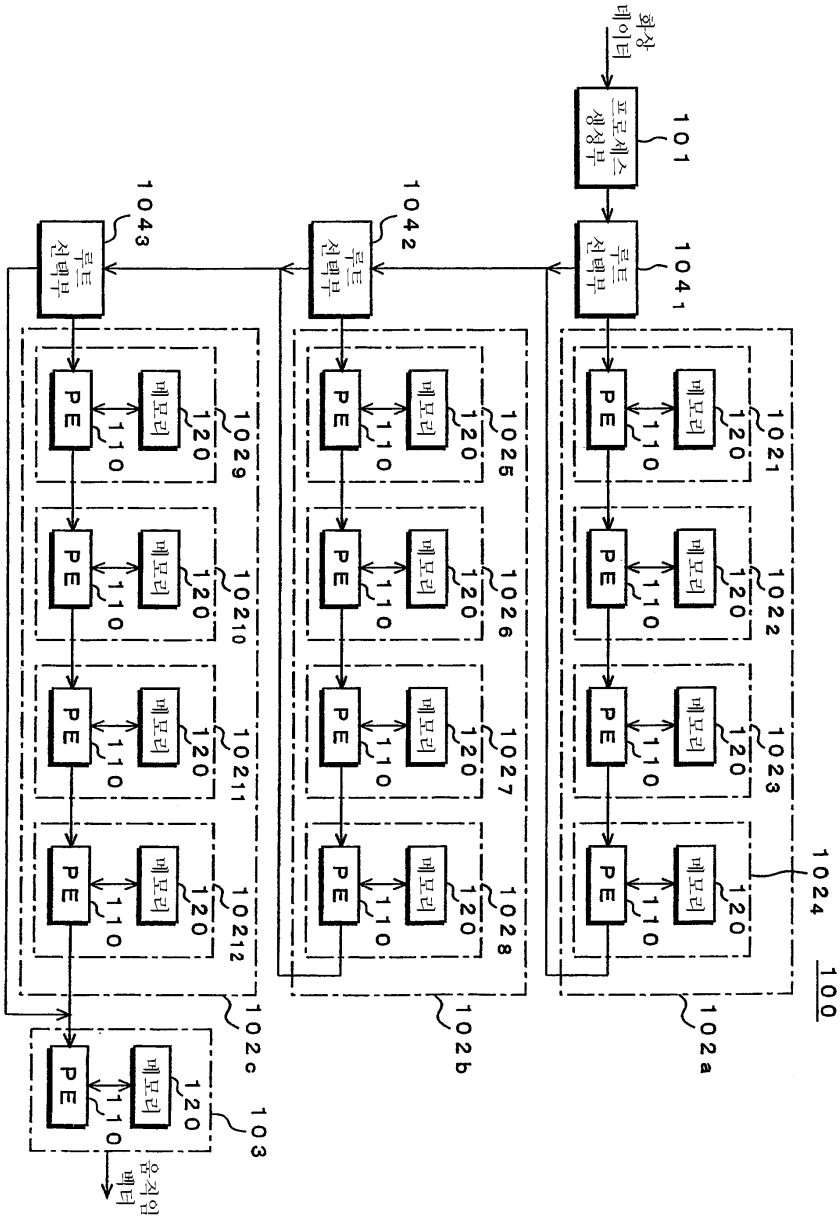
도면3



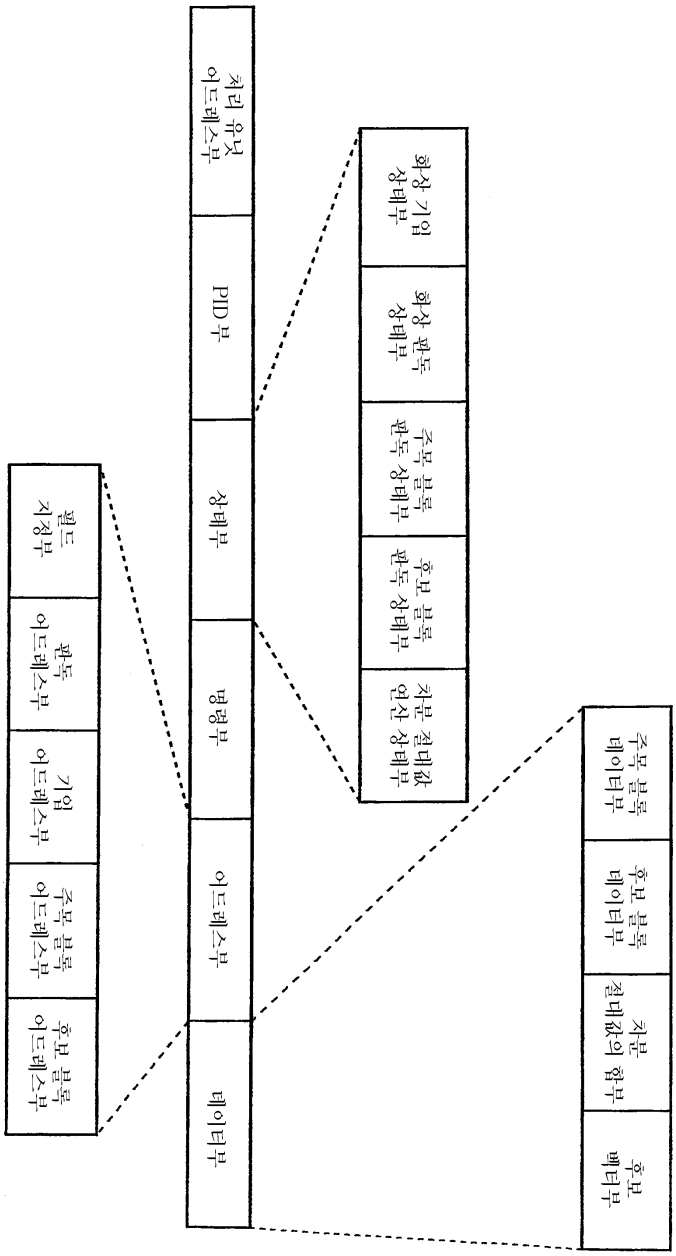
도면4


201
2011
2012
2013
2014
2015
2016
메모리
메모리
메모리
메모리
메모리
메모리
데이터 버스
203
캐쉬 메모리
202
움직임 벡터 추출부
움직임 벡터

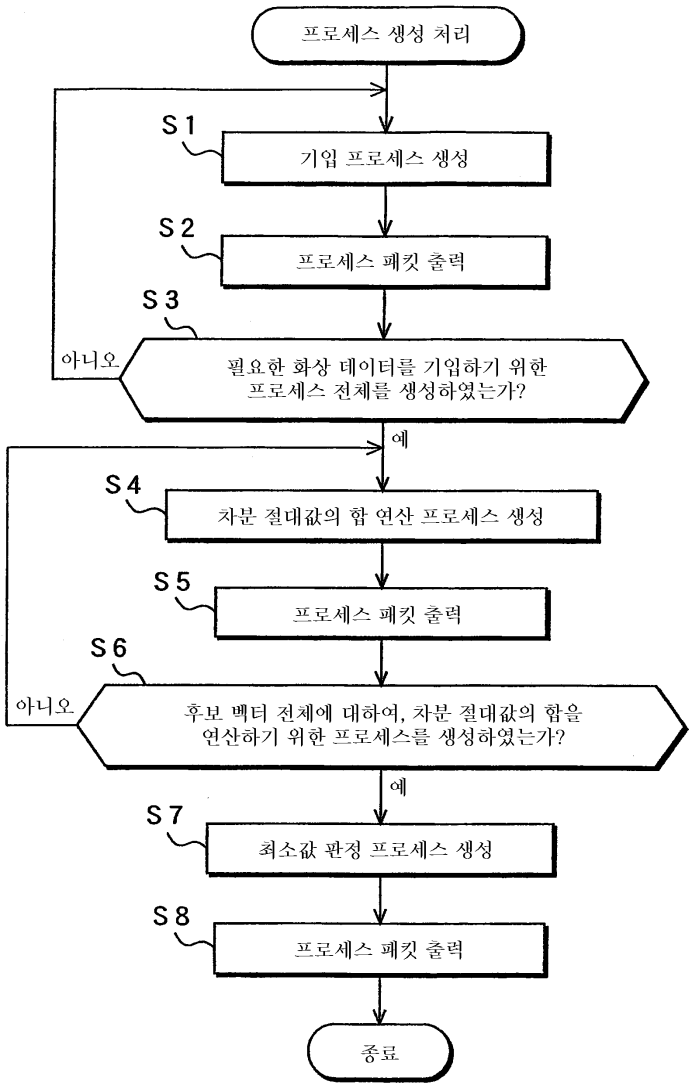
도면5


화상 데이터
프로세스 생성부
101
쿼드 선택부
1041
쿼드 선택부
1042
쿼드 선택부
1043
PE
메모리
110
120
1021
1022
1023
1024
100
1025
1026
1027
1028
1029
10210
10211
10212
102a
102b
102c
103
출력 버퍼

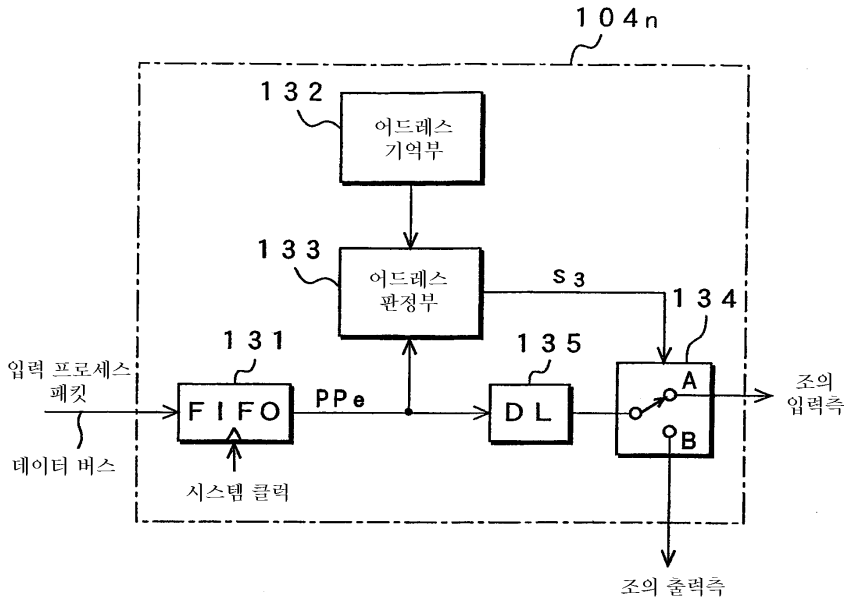
도면6


처리 유닛 어드레스부
PID 부
상태부
명령부
어드레스부
데이터부
화상 기입 상태부
화상 판독 상태부
주목 블록 판독 상태부
후보 블록 판독 상태부
차분 절대값 연산 상태부
웰드 지정부
판독 어드레스부
기입 어드레스부
주목 블록 어드레스부
후보 블록 어드레스부
주목 블록 데이터부
후보 블록 데이터부
차분 절대값의 합부
후보 벡터부

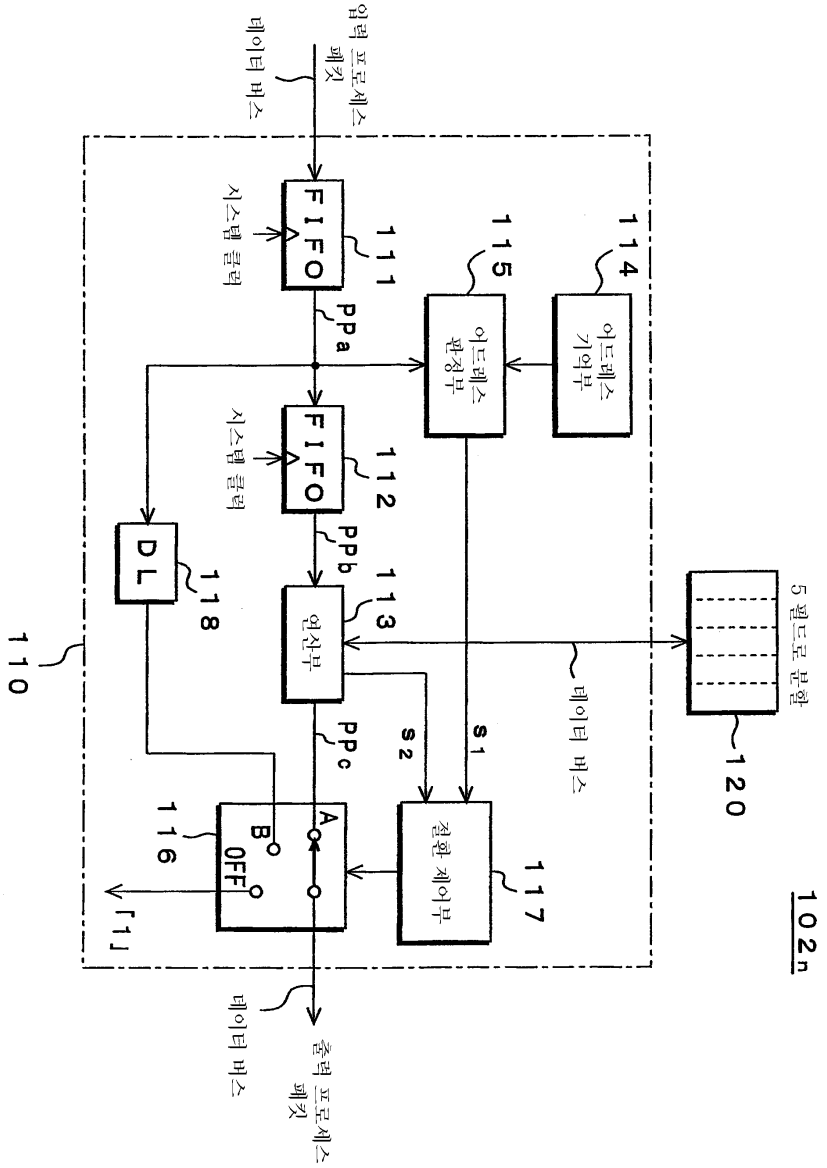
도면7


프로세스 생성 처리
S1
기입 프로세스 생성
S2
프로세스 패킷 출력
S3
아니오
필요한 화상 데이터를 기입하기 위한 프로세스 전체를 생성하였는가?
예
S4
차분 절대값의 합 연산 프로세스 생성
S5
프로세스 패킷 출력
S6
아니오
후보 벡터 전체에 대하여, 차분 절대값의 합을 연산하기 위한 프로세스를 생성하였는가?
예
S7
최소값 판정 프로세스 생성
S8
프로세스 패킷 출력
종료

도면8


104n
132
어드레스
기억부
133
어드레스
판정부
S3
134
131
135
입력 프로세스
패킷
FIFO
PPe
DL
A
B
조의
입력측
데이터 버스
시스템 클럭
조의 출력측

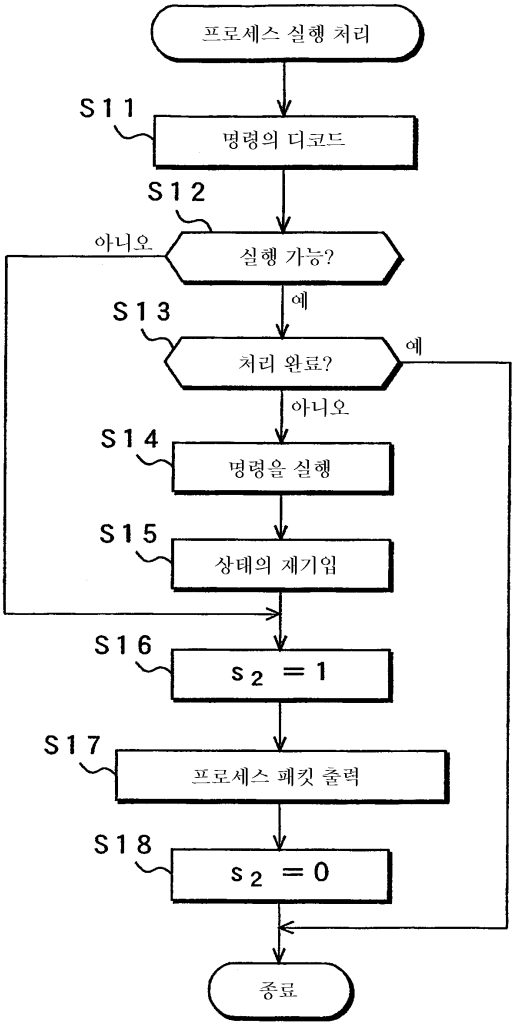
도면9



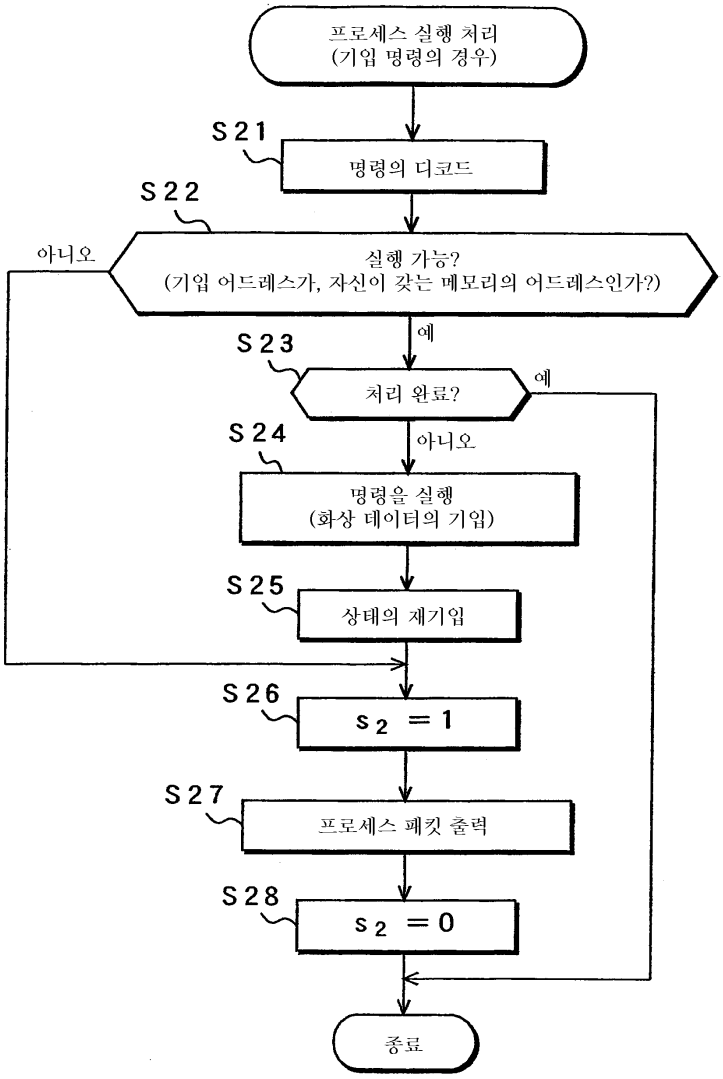
도면10

판정 신호 S_1	요구 신호 S_2	절환 스위치 116
0	0	B
0	1	A
1	0	OFF
1	1	A

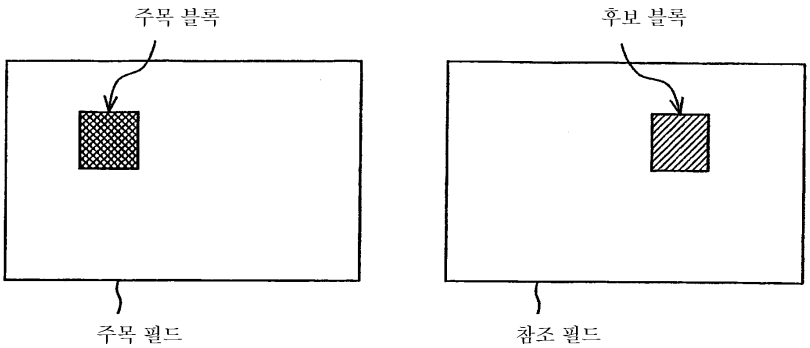
도면11


프로세스 실행 처리
S11
명령의 디코드
S12
아니오
실행 가능?
예
S13
처리 완료?
예
아니오
S14
명령을 실행
S15
상태의 재기입
S16
s2 = 1
S17
프로세스 패킷 출력
S18
s2 = 0
종료

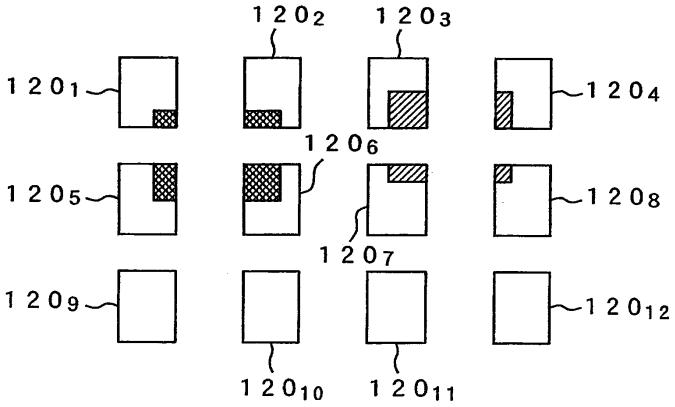
도면12


프로세스 실행 처리
(기입 명령의 경우)
S21
명령의 디코드
S22
아니오
실행 가능?
(기입 어드레스가, 자신이 갖는 메모리의 어드레스인가?)
예
S23
처리 완료?
예
S24
아니오
명령을 실행
(화상 데이터의 기입)
S25
상태의 재기입
S26
s2 = 1
S27
프로세스 패킷 출력
S28
s2 = 0
종료

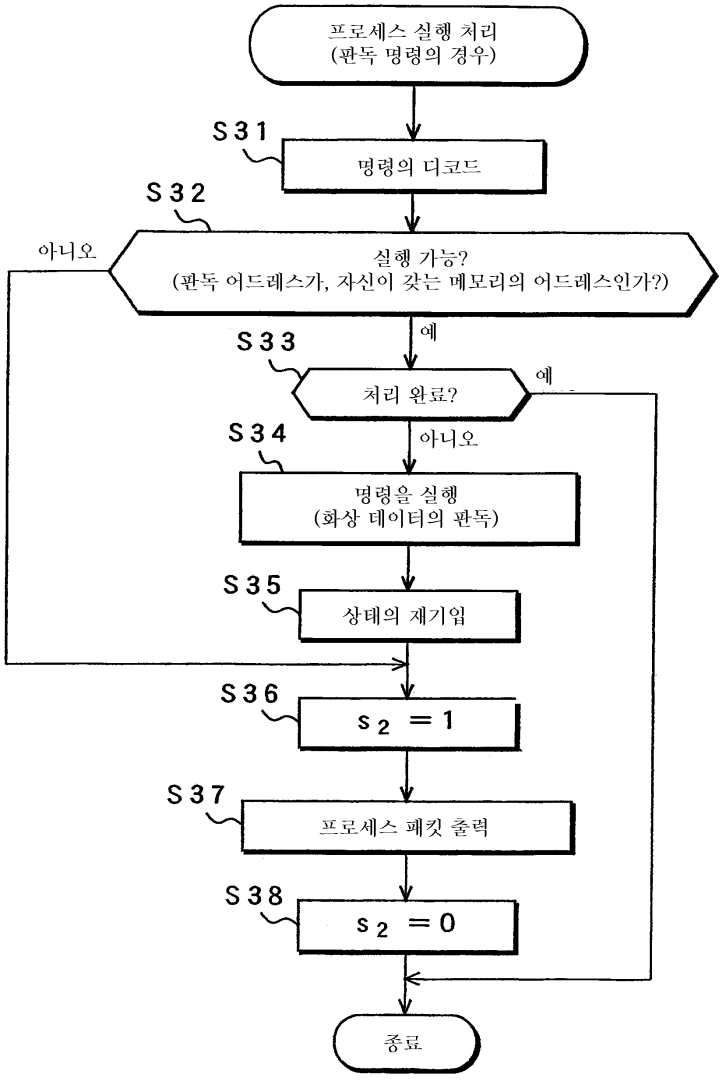
도면13a


주목 블록
후보 블록
주목 필드
참조 필드

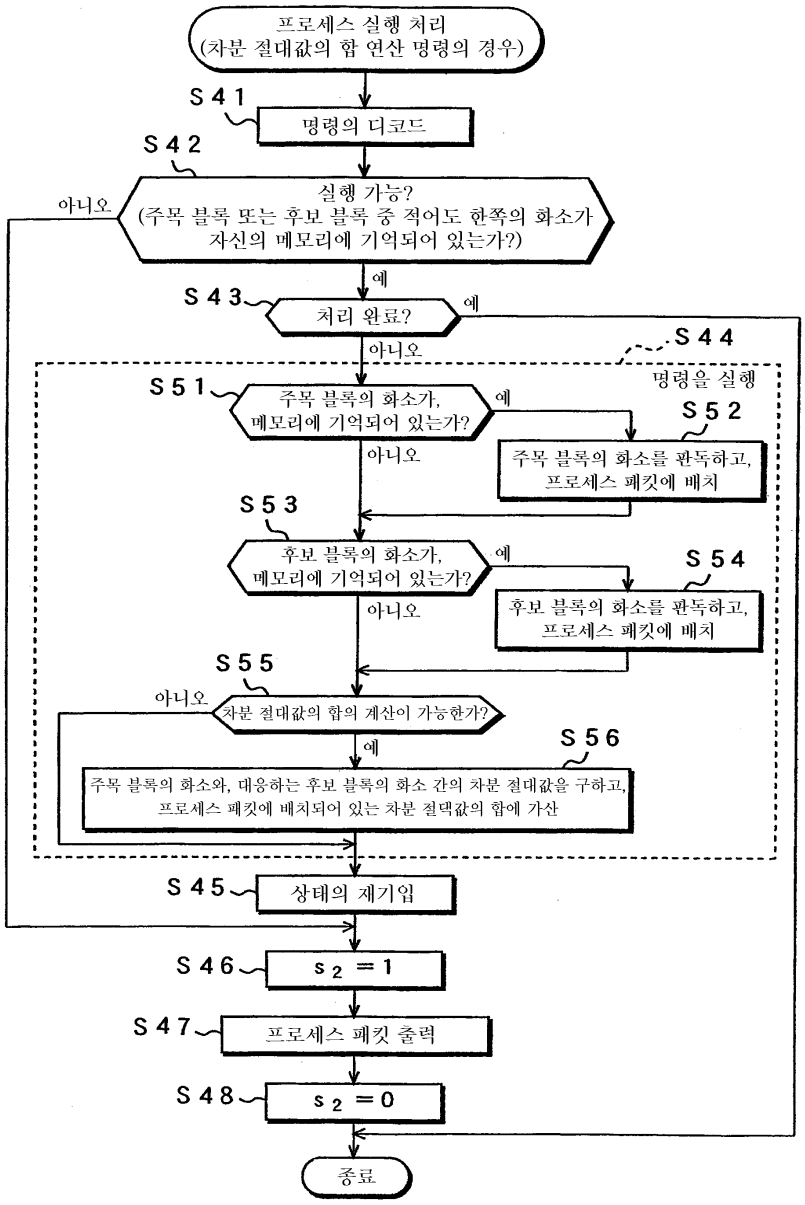
도면13b


1202
1203
1201
1204
1206
1205
1208
1207
1209
12012
12010
12011

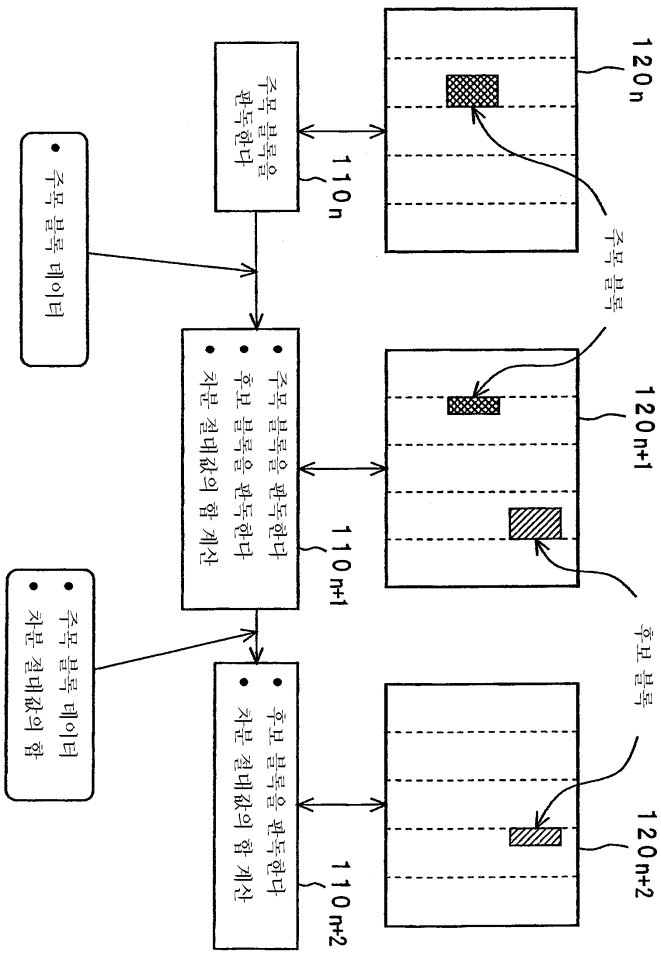
도면14


프로세스 실행 처리
(판독 명령의 경우)
S31
명령의 디코드
S32
아니오
실행 가능?
(판독 어드레스가, 자신이 갖는 메모리의 어드레스인가?)
예
S33
처리 완료?
예
S34
아니오
명령을 실행
(화상 데이터의 판독)
S35
상태의 재기입
S36
s2 = 1
S37
프로세스 패킷 출력
S38
s2 = 0
종료

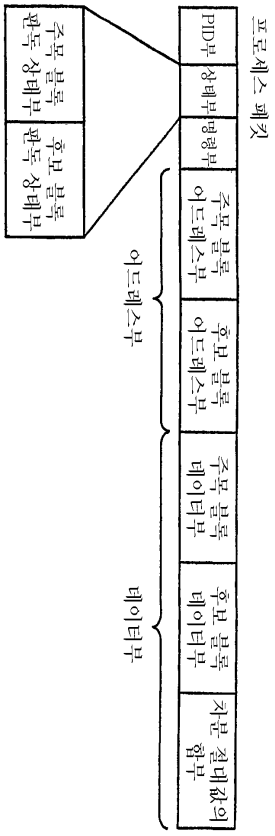
도면15


프로세스 실행 처리
(차분 절대값의 합 연산 명령의 경우)
S41
명령의 디코드
S42
실행 가능?
(주목 블록 또는 후보 블록 중 적어도 한쪽의 화소가 자신의 메모리에 기억되어 있는가?)
아니오
예
S43
처리 완료?
예
아니오
S44
명령을 실행
S51
주목 블록의 화소가, 메모리에 기억되어 있는가?
예
S52
주목 블록의 화소를 판독하고, 프로세스 페킷에 배치
아니오
S53
후보 블록의 화소가, 메모리에 기억되어 있는가?
예
S54
후보 블록의 화소를 판독하고, 프로세스 페킷에 배치
아니오
S55
아니오
차분 절대값의 합의 계산이 가능한가?
예
S56
주목 블록의 화소와, 대응하는 후보 블록의 화소 간의 차분 절대값을 구하고, 프로세스 페킷에 배치되어 있는 차분 절댁값의 합에 가산
S45
상태의 재기입
S46
s2 = 1
S47
프로세스 페킷 출력
S48
s2 = 0
종료

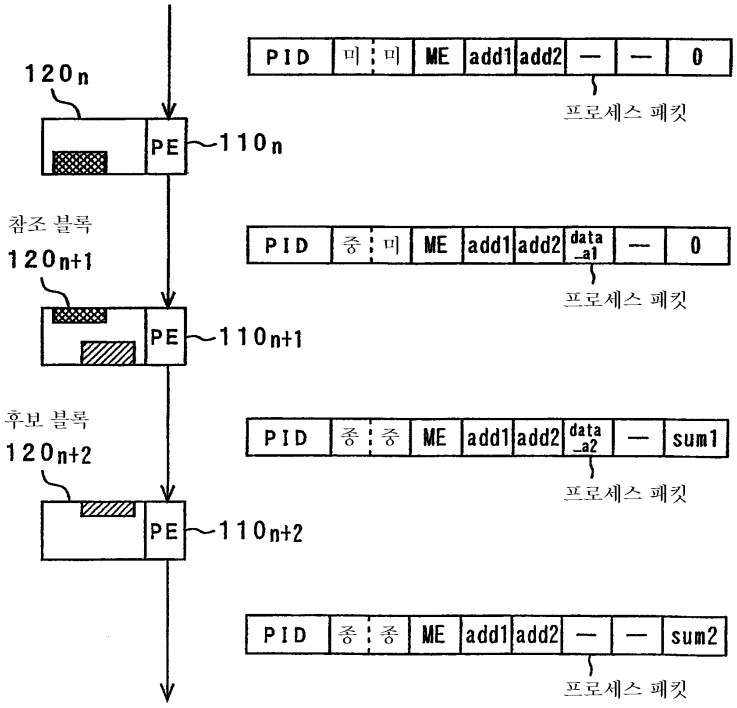
도면16


120n
주목 블록
120n+1
후보 블록
120n+2
주목 블록을 판독한다
110n
• 주목 블록 데이터
• 주목 블록을 판독한다
• 후보 블록을 판독한다
• 차분 절대값의 합 계산
110n+1
• 주목 블록 데이터
• 차분 절대값의 합
• 후보 블록을 판독한다
• 차분 절대값의 합 계산
110n+2

도면17a


프로세스 패킷
PID부
상태부
명령부
주목 블록 어드레스부
후보 블록 어드레스부
주목 블록 데이터부
후보 블록 데이터부
차분 절대값의 합부
어드레스부
데이터부
주목 블록 판독 상태부
후보 블록 판독 상태부

도면17b


120n
PE
110n
PID
미
미
ME
add1
add2
—
—
0
프로세스 패킷
참조 블록
120n+1
PE
110n+1
PID
종
미
ME
add1
add2
data_a1
—
0
프로세스 패킷
후보 블록
120n+2
PE
110n+2
PID
종
종
ME
add1
add2
data_a2
—
sum1
프로세스 패킷
PID
종
종
ME
add1
add2
—
—
sum2
프로세스 패킷