



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2008년06월16일
(11) 등록번호 10-0838438
(24) 등록일자 2008년06월09일

(51) Int. Cl.
G06F 9/46 (2006.01) G06F 9/48 (2006.01)
G06F 1/00 (2006.01)
(21) 출원번호 10-2006-7011159
(22) 출원일자 2006년06월07일
심사청구일자 2006년08월22일
번역문제출일자 2006년06월07일
(65) 공개번호 10-2007-0022198
(43) 공개일자 2007년02월26일
(86) 국제출원번호 PCT/JP2005/006966
국제출원일자 2005년04월08일
(87) 국제공개번호 WO 2005/119447
국제공개일자 2005년12월15일
(30) 우선권주장
JP-P-2004-00163649 2004년06월01일 일본(JP)
(56) 선행기술조사문헌
JP10171668 A*
JP15108260 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
가부시킴가이샤 소니 컴퓨터 엔터테인먼트
일본국 도쿄도 107-0062 미나토구 미나미-아오야
마 2-6-21
(72) 발명자
아다치 켄이치
일본국 도쿄도 미나토구 미나미-아오야마 2-6-21
가부시킴가이샤소니 컴퓨터 엔터테인먼트 내
야자와 카즈야키
일본국 도쿄도 미나토구 미나미-아오야마 2-6-21
가부시킴가이샤소니 컴퓨터 엔터테인먼트 내
(뒷면에 계속)
(74) 대리인
윤동열

전체 청구항 수 : 총 29 항

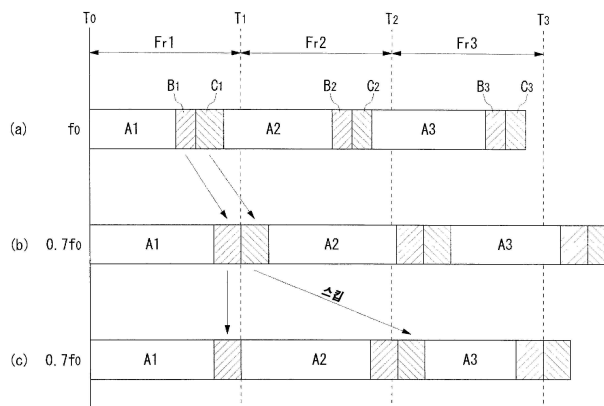
심사관 : 노영철

(54) 태스크 관리방법, 태스크 관리장치, 반도체 집적회로, 전자장치, 태스크 관리 시스템, 및 프로그램을 격납한 기록매체

(57) 요약

이 태스크 관리방법은, 처리의 단위시간을 실시간성을 보증하는 보호 대역과 실시간성을 보증하지 않는 비보호 대역으로 분할하여, 프로세서의 처리능력이 저하했을 때, 비보호영역에 있어서 실행해야 할 태스크의 실행을 적의 스킵하는 것이다. 즉, 프로세서의 발열을 억제할 목적으로 동작주파수를 낮게 한 경우, 비보호 대역에 있어서 실행해야 할 태스크를 베스트 에포트에서 처리하는 대신에 보호 대역에서 실행해야 할 태스크의 실시간성을 보증하는 것이다. 이에 따라, 동작주파수가 변동할 경우에도 적절하게 태스크의 관리가 행해져 프로세서의 처리능력이 충분히 발휘된다.

대표도 - 도2



(72) 발명자

타키구치 이와오

일본국 도쿄도 미나토구 미나미-아오야마 2-6-21
가부시키가이샤소니 컴퓨터 엔터테인먼트 내

이마이 아츠히코

일본국 도쿄도 미나토구 미나미-아오야마 2-6-21
가부시키가이샤소니 컴퓨터 엔터테인먼트 내

타무라 테츠지

일본국 도쿄도 미나토구 미나미-아오야마 2-6-21
가부시키가이샤소니 컴퓨터 엔터테인먼트 내

특허청구의 범위

청구항 1

프로세서에 의해 태스크를 실행할 때, 처리의 단위시간을, 실시간성을 보증하기 위한 보호 대역과 실시간성을 보증하지 않는 비보호 대역으로 분할하여, 프로세서의 처리능력이 저하했을 때, 상기 비보호 대역에 있어서 실행해야 할 태스크의 실행을 스킵하는 것을 특징으로 하는 태스크 관리방법.

청구항 2

제1항에 있어서, 프로세서 또는 그 주변회로의 온도가 소정의 역치값을 초과했을 때 프로세서의 동작주파수가 저감되는 것을 특징으로 하는 태스크 관리방법.

청구항 3

제1항에 있어서, 상기 프로세서의 소비전력 값이 소정의 역치값을 초과했을 때 프로세서의 동작주파수가 저감되는 것을 특징으로 하는 태스크 관리방법.

청구항 4

프로세서에 의해 태스크를 실행할 때, 처리의 단위시간을, 실시간성을 보증하기 위한 보호대역과 실시간성을 보증하지 않는 비보호대역으로 분할하여, 상기 프로세서의 처리능력을 저하시킬 필요가 있는 경우, 상기 비보호대역에 있어서의 처리능력을 저하시키는 것을 특징으로 하는 태스크 관리방법.

청구항 5

프로세서에 의해 태스크를 실행할 때, 처리의 단위시간을, 실시간성을 보증하기 위한 보호대역과 실시간성을 보증하지 않는 비보호대역으로 분할하여, 상기 프로세서의 처리능력이 저하되는 경우, 상기 비보호대역에 있어서의 프로세서의 점유시간을 단축시키는 것을 특징으로 하는 태스크 관리방법.

청구항 6

주처리부에 실행시킬 복수의 태스크의 전환지시를 행하는 전환지시부와,

상기 주처리부의 처리능력을 검출하는 검출부를 구비하며,

상기 전환지시부는, 처리의 단위시간을, 실시간성을 보증하기 위한 보호 대역과 실시간성을 보증하지 않는 비보호 대역으로 분할하여, 상기 주처리부의 처리능력이 저하했을 때, 비보호 대역에 있어서 실행해야 할 태스크의 실행을 스킵시키는 것을 특징으로 하는 태스크 관리장치.

청구항 7

제6항에 있어서, 상기 검출부는, 상기 주처리부에 있어서의 동작주파수를 검출하는 것을 특징으로 하는 태스크 관리장치.

청구항 8

제6항 또는 제7항에 있어서, 각 태스크로 실행되는 프로그램 내에 기술된 실시간성에 대한 요청을 해석하는 해석부를 더 구비하며,

상기 전환지시부는, 그 해석에 기초하여 각 태스크를 상기 보호 대역 또는 상기 비보호 대역 중 어느 하나에 할당하는 것을 특징으로 하는 태스크 관리장치.

청구항 9

제6항 또는 제7항에 있어서, 각 태스크로 실행되는 프로그램에서 호출되는 명령, 프로세서 점유율, 프로세서 점유시간 및 태스크의 실행에 따라 얻어지는 특성 중 하나 이상을 판단하는 판단부를 더 구비하며,

상기 전환지시부는, 그 판단에 기초하여 각 태스크를 상기 보호 대역 또는 상기 비보호 대역 중 어느 하나에 할당하는 것을 특징으로 하는 태스크 관리장치.

청구항 10

제6항 또는 제7항에 있어서, 상기 단위시간은 표시에 관련된 단위시간인 것을 특징으로 하는 태스크 관리장치.

청구항 11

제6항 또는 제7항에 있어서, 상기 주처리부의 사용률을 검출하는 제2검출부를 더 구비하며,

상기 전환지시부는, 상기 사용률에 따라 비보호 대역에 있어서 실행해야 할 태스크의 실행률을 변경하는 것을 특징으로 하는 태스크 관리장치.

청구항 12

제11항에 있어서, 상기 주처리부의 처리능력에 관한 정보와, 그 처리능력에서의 비보호 대역에 있어서 실행해야 할 태스크의 실행률을 대응시켜서 보유하는 테이블을 더 구비하며,

상기 전환지시부는, 상기 주처리부의 사용률이 소정의 역치값보다 낮은 경우, 비보호 대역에 있어서 실행해야 할 태스크의 실행률을, 상기 테이블에서 설정된 실행률보다 높이는 것을 특징으로 하는 태스크 관리장치.

청구항 13

주처리부에 실행시킬 복수의 태스크의 전환지시를 행하는 전환지시부와,

상기 주처리부의 처리능력을 검출하는 검출부를 구비하며,

상기 전환지시부는, 처리의 단위시간을, 실시간성을 보증하기 위한 보호대역과 실시간성을 보증하지 않는 비보호대역으로 분할하여, 상기 주처리부의 처리능력을 저하시킬 필요가 있는 경우, 상기 비보호대역에 있어서의 처리능력을 저하시키는 것을 특징으로 하는 태스크 관리장치.

청구항 14

소정의 태스크를 실행하는 주처리부; 및

처리 단위시간을, 실시간성을 보증하기 위한 보호 대역과 실시간성을 보증하지 않는 비보호 대역으로 분할하여, 상기 주처리부의 처리능력이 저하했을 때, 비보호 대역에 있어서 실행해야 할 태스크의 실행을 스킵시키는 태스크 관리부;를 구비하는 것을 특징으로 하는 반도체 집적회로.

청구항 15

제14항에 있어서, 상기 주처리부에 소정의 동작주파수의 클럭을 공급하는 클럭 생성부를 더 구비하며,

상기 태스크 관리부는, 상기 동작주파수가 저하했을 때, 비보호 대역에 있어서 실행해야 할 태스크의 실행을 스킵시키는 것을 특징으로 하는 반도체 집적회로.

청구항 16

제15항에 있어서, 상기 주처리부 또는 그 주변회로의 온도가 소정의 역치값을 초과했을 때, 상기 클럭 생성부는 상기 동작주파수를 저감하는 것을 특징으로 하는 반도체 집적회로.

청구항 17

제16항에 있어서, 상기 클럭 생성부는, 상기 주처리부의 소비전력 값이 소정의 역치값을 초과했을 때 상기 동작주파수를 저감하는 것을 특징으로 하는 반도체 집적회로.

청구항 18

제15항에 있어서, 상기 태스크 관리부는, 상기 주처리부 또는 그 주변회로의 온도가 소정의 역치값을 초과했을 때, 비보호 대역에 있어서 실행해야 할 태스크의 실행을 스킵시키는 것을 특징으로 하는 반도체 집적회로.

청구항 19

제16항에 있어서, 상기 태스크 관리부는, 상기 주처리부의 소비전력 값이 소정의 역치값을 초과했을 때 비보호

대역에 있어서 실행해야 할 태스크의 실행을 스킵시키는 것을 특징으로 하는 반도체 집적회로.

청구항 20

소정의 동작주파수에서 태스크를 실행하는 주처리부와,

상기 동작주파수의 클럭을 상기 주처리부에 공급하는 클럭 생성부와,

태스크 관리기능을 실현하기 위한 프로그램을 외부로부터 읽어들이으로써 동적으로 태스크 관리기능을 실현하기 위한 회로를 구비하며,

상기 태스크 관리기능은, 처리의 단위시간을, 실시간성을 보증하기 위한 보호 대역과 실시간성을 보증하지 않는 비보호 대역으로 분할하여, 상기 동작주파수가 저하했을 때, 비보호 대역에 있어서 실행해야 할 태스크의 실행을 스킵시키는 것을 특징으로 하는 반도체 집적회로.

청구항 21

소정의 동작주파수에서 태스크를 실행하는 프로세서와,

상기 프로세서에 실행시킬 프로그램을 보유하는 기억부를 구비하며,

상기 프로그램은, 처리의 단위시간을, 실시간성을 보증하기 위한 보호 대역과 실시간성을 보증하지 않는 비보호 대역으로 분할하여, 상기 동작주파수가 저하했을 때, 비보호 대역에 있어서 실행해야 할 태스크의 실행을 스킵하도록 태스크를 스케줄링하는 기능을 상기 프로세서에 실행시키는 것을 특징으로 하는 전자장치.

청구항 22

제21항에 있어서, 상기 프로세서 또는 그 주변회로의 온도가 소정의 역치값을 초과했을 때 상기 동작주파수를 저감하는 주파수 제어부를 더 구비하는 것을 특징으로 하는 전자장치.

청구항 23

제21항에 있어서, 상기 프로세서의 소비전력 값이 소정의 역치값을 초과했을 때 상기 동작주파수를 저감하는 주파수 제어부를 더 구비하는 것을 특징으로 하는 전자장치.

청구항 24

프로세서에 의해 태스크를 실행할 때, 처리의 단위시간을, 실시간성을 보증하기 위한 보호 대역과 실시간성을 보증하지 않는 비보호 대역으로 분할하여, 프로세서의 처리능력이 저하했을 때, 비보호 대역에 있어서 실행해야 할 태스크의 실행을 스킵하는 단계를 컴퓨터가 실행할 수 있도록 하는 프로그램을 격납한 기록매체.

청구항 25

프로세서에 의해 실행해야 할 태스크를 그 성질에 따라 제1타입과 제2타입으로 분류하고, 소정의 요인으로 처리의 실시간성이 저해될 가능성이 있을 때, 제1타입의 태스크를 실행하면서, 제1타입의 태스크 사이에 실행되어야 할 제2타입의 태스크의 실행을 스킵하는 단계를 컴퓨터가 실행할 수 있도록 하는 프로그램을 격납한 기록매체.

청구항 26

소정의 동작주파수에서 태스크를 실행하는 프로세서와,

상기 동작주파수의 클럭을 상기 프로세서에 공급하는 클럭 생성부와,

상기 프로세서에 실행시킬 복수의 태스크의 전환지시를 행하는 전환지시부를 구비하며,

상기 전환지시부는, 처리의 단위시간을, 실시간성을 보증하기 위한 보호 대역과 실시간성을 보증하지 않는 비보호 대역으로 분할하여, 상기 프로세서의 동작주파수가 저하했을 때, 비보호 대역에 있어서 실행해야 할 태스크의 실행을 스킵시키는 것을 특징으로 하는 태스크 관리 시스템.

청구항 27

제26항에 있어서, 상기 클럭 생성부는, 상기 프로세서 또는 그 주변회로의 온도가 소정의 역치값을 초과했을 때

상기 동작주파수를 저감하는 것을 특징으로 하는 태스크 관리 시스템.

청구항 28

제26항에 있어서, 상기 클럭 생성부는, 상기 프로세서의 소비전력에 따라 상기 동작주파수를 저감하는 것을 특징으로 하는 태스크 관리 시스템.

청구항 29

주처리부에 실행시킬 복수의 태스크의 전환지시를 행하는 전환지시부와,

상기 주처리부의 처리능력을 검출하는 검출부를 구비하며,

상기 전환지시부는, 처리의 단위시간을, 실시간성을 보증하기 위한 보호대역과 실시간성을 보증하지 않는 비보호대역으로 분할하여, 상기 주처리부의 처리능력이 저하되는 경우, 상기 비보호대역에 있어서의 프로세서의 점유시간을 단축시키는 것을 특징으로 하는 태스크 관리장치.

명세서

기술분야

<1> 이 발명은 프로세서에 실행되는 태스크를 관리하는 방법, 그 방법을 이용하여 태스크를 관리하는 장치, 그 장치의 실체로서의 반도체 집적회로, 및 그 반도체 집적회로를 구비하는 전자장치에 관한 것이다.

배경기술

<2> LSI 설계에 있어서 제조 프로세스의 미세화와 소자의 고집적화가 한층 진행되어, 칩의 성능한계로서 발열량을 고려하는 것이 설계상 매우 중요하게 되었다. 칩이 고온이 되면 동작불량을 일으키거나, 장기 신뢰성이 저하하거나 하기 때문에 여러 가지 발열대책이 취해지고 있다. 예를 들면, 칩의 상부에 방열 핀을 설치하여 칩으로부터 발생하는 열을 내보내는 방법이 이용된다.

<3> 또한 칩의 소비전력 분포에 기초하여, 프로세서의 태스크를 스케줄링하는 것도 검토되고 있다. 또한 칩이 고온이 되는 것을 회피하는 방법으로서, 프로세서에 있어서의 동작주파수를 낮게 하는 것도 검토되고 있다(예를 들면, 특허문헌 1 참조).

<4> 특허문헌 1: 미국 특허출원 공개 제2002/0065049호 명세서

발명의 상세한 설명

<5> 동작주파수를 낮게 함으로써 칩의 발열량을 낮출 수는 있지만, 단위시간에 있어서의 처리단계 수가 감소해 버리기 때문에, 단위시간 내에 완료해야 할 태스크가 단위시간 내에 완료되지 못하는 경우가 있다. 이 때문에, 미리 최저의 동작주파수 베이스에서 프로그램의 설계를 행할 필요가 있다. 그러나 이것으로는 프로세서의 처리능력이 충분히 발휘되지 않는다.

<6> 본 발명은 이러한 과제에 비추어 이루어진 것으로, 그 목적은, 이러한 이율 배반되는 목표를 실현하여 효과적인 태스크의 관리방법, 그 방법에 기초한 태스크 관리장치, 그 장치의 실체로서의 반도체 집적회로, 및 그 반도체 집적회로를 구비하는 전자장치를 제공하는 데 있다.

<7> 본 발명의 어떤 형태는 태스크의 실행방법에 관한다. 이 방법은 프로세서에 의해 태스크를 실행할 때, 처리의 단위시간을, 실시간성을 보증하기 위한 보호 대역과 실시간성을 보증하지 않는 비보호 대역으로 분할하여, 프로세서의 처리능력이 저하했을 때, 비보호 대역에 있어서 실행해야 할 태스크의 실행을 적의 스킵한다. 엄밀하게는, 태스크를 실행한다거나 또는 태스크를 스킵한다고 하는 것은 그 태스크에 관한 프로세스, 즉 프로그램을 실행 또는 스킵하는 것인데, 이하 간결하게 하기 위해 간단히 "태스크를 실행한다" 또는 "태스크를 스킵한다" 등과 같이 표현한다. "태스크"는 프로그램상의 블록에 관계없이 하나의 기능의 블록을 가리키는 것으로 한다. 따라서 실제로 보급하고 있는 OS가 파악하는 태스크보다도 큰 기능단위를 가리키는 것도 있다.

<8> "프로세서의 처리능력의 저하"는 프로세서의 동작주파수의 저감에 기인하는 것이어도 된다. 프로세서의 동작주파수는 프로세서 또는 그 주변회로의 온도가 소정의 역치값을 초과했을 때에 저감되어도 된다. 또한 프로세서의 소비전력에 응하여 저감되어도 된다.

- <9> 이 형태에 따르면, 프로세서의 처리능력이 저하했을 때, 비보호 대역에 있어서 실행해야 할 태스크의 실행이 적의 스킵되기 때문에, 보호 대역에 있어서 실행되어야 할 태스크의 실시간성이 보증되거나, 적어도 그 정확도가 대폭으로 증가한다(단, 이후 간단히 "보증된다"라고 함). 바꿔 말하면, 실시간성을 보증해야 할 태스크를 보호 대역에 포함되도록 설계해 두면, 보호 대역을 포함하지 않는 범위에서 프로세서의 처리능력을 변경시킬 수 있고, 프로세서의 열 제어를 유연하게 행할 수 있다.
- <10> 본 발명의 다른 형태는 태스크 관리장치이다. 이 장치는, 프로세서에 실행시키는 복수의 태스크의 전환지시를 행하는 전환지시부와 프로세서의 처리능력을 검출하는 검출부를 구비하고, 전환지시부는, 처리의 단위시간을, 실시간성을 보증하기 위한 보호 대역과 실시간성을 보증하지 않는 비보호 대역으로 분할하여, 프로세서의 처리능력이 저하했을 때, 비보호 대역에 있어서 실행해야 할 태스크의 실행을 적의 스킵시킨다.
- <11> 이 장치는, 각 태스크로 실행되는 프로그램 내에 기술(記述)된 실시간성에 대한 요청을 해석하는 해석부를 더 구비해도 된다. 그 경우, 전환지시부는 그 해석에 기초하여 각 태스크를 보호 대역 또는 비보호 대역의 어느 쪽에 할당해도 된다. "실시간성에 대한 요청"은, 예를 들면 태스크를 보호 대역에서 실행해야 하는지의 여부의 판정에 이용가능한 정보여도 되고, 프로그램 내에 기술된 속성을 나타내는 정보여도 되며, 태스크의 중요도나 우선도를 나타내는 정보여도 된다.
- <12> 이 장치는, 각 태스크로 실행되는 프로그램의 성질을 프로세서에서 판단하는 판단부를 더 구비해도 된다. 그 경우, 전환지시부는 그 판단에 기초하여 각 태스크를 보호 대역 또는 비보호 대역의 어느 쪽에 할당해도 된다. "프로그램의 성질"은 그 프로그램에서 호출되는 명령이어도 되고, 그 프로그램에 의한 프로세서의 점유율이나 점유시간이어도 되며, 태스크를 실행함으로써 간접적으로 얻어지는 특징이어도 된다.
- <13> 본 발명의 다른 형태는 태스크 관리시스템이다. 이 시스템은, 소정의 동작주파수에서 태스크를 실행하는 프로세서와, 그 동작주파수의 클럭(clock)을 프로세서에 공급하는 클럭생성부와, 프로세서에 실행시키는 복수의 태스크의 전환지시를 행하는 전환지시부를 구비하고, 전환지시부는, 처리의 단위시간을, 실시간성을 보증하기 위한 보호 대역과 실시간성을 보증하지 않는 비보호 대역으로 분할하여, 프로세서의 동작주파수가 저하했을 때, 비보호 대역에 있어서 실행해야 할 태스크의 실행을 적의 스킵시킨다.
- <14> 한편, 이상의 구성요소의 임의의 조합, 본 발명의 표현을 방법, 장치, 시스템, 컴퓨터 프로그램 등의 사이에서 변환한 것도 또한 본 발명의 형태로서 유효하다.
- <15> [발명의 효과]
- <16> 본 발명에 따르면, 실시간 태스크와 비 실시간 태스크를 멀티 태스크로 실행하고 있을 때의 프로세서의 처리능력이 저하한 경우에도 실시간 태스크의 실시간성을 보증할 수 있다.

실시 예

- <48> 실시의 형태를 설명하기 전에 과제를 명확히 한다. 예를 들면 게임의 경우, 프레임(frame) 기간에 CG의 렌더링(rendering) 등의 묘화처리가 완료하도록 설계된다. 이렇게 소정의 시간 내에 처리가 완료하지 않으면 안 되는 태스크를, 이하 "실시간 태스크"라고 부른다. 실시간 태스크와 같이 시간제한이 설정되어 있지 않은 태스크를, 이하 "비 실시간 태스크"라고 부른다. 이러한 시간에 대한 설계사상이 다른 2종류의 태스크를 병행하여 실행하는 경우, 실시간 태스크의 실시간성은 비 실시간 태스크의 처리시간에 크게 영향을 받는다.
- <49> 도 1의 (a)는 통상시의 동작주파수 f_0 에서 작동하는 프로세서에 있어서의 태스크의 처리상태를 시계열적으로 나타내는 도이다. 시각 T_0 에서 T_1 , 시각 T_1 에서 T_2 , 및 시각 T_2 에서 T_3 는 각각 1프레임 기간이다. 본 도면에서는, 프레임의 묘화를 행하기 위한 실시간 태스크(RT), 및 2개의 비 실시간 태스크(NRT)가 순서대로 실행되고 있다. 시각 T_1 , T_2 , T_3 에서 프레임의 표시가 행해지게 된다. 시각 T_0 에서 T_1 까지의 사이에 실시간 태스크(RT), 제1 비 실시간 태스크(NRT1), 및 제2 비 실시간 태스크(NRT2)가 순차 실행된다. 그 후, 시각 T_2 에서 표시하는 프레임을 묘화하는 실시간 태스크(RT)가 실행되고, 이어서 제1 비 실시간 태스크(NRT1), 제2 비 실시간 태스크(NRT2)가 순차 실행된다. 시각 T_1 , T_2 , T_3 에 프레임을 표시하기 위한 실시간 태스크(RT)는, 각각 시각 T_1 , T_2 , T_3 보다 앞에서 처리가 완료하고 있기 때문에, 드롭(dropping)되지 않고 프레임이 표시된다.
- <50> 도 1의 (b)는 동작주파수 $0.8f_0$ 인 경우의 프로세서에 있어서의 태스크의 처리상태를 시계열적으로 나타내는 도이다. 프로세서의 동작주파수는, 예를 들면 프로세서에 있어서의 열 제어를 위해 조정되고, 프로세서의 온도가 높아지면 동작주파수가 낮게 조정된다. 동작주파수가 낮아지면, 프레임 기간에 있어서의 프로세서의 실행 사이클

수가 적어진다. 이 때문에, 각 태스크를 완료하기까지의 시간이 길어져 프레임 기간마다 실행되어야 할 실시간 태스크(RT), 제1 비 실시간 태스크(NRT1) 및 제2 비 실시간 태스크(NRT2)가, 본래 실행되어야 할 프레임 기간을 초과해 다음 프레임 기간에서 실행되게 된다. 본 도면에서는, 실시간 태스크(RT)의 실시간성은 거의 유지되지만, 장기적으로는 프레임의 드롭이 생겨 버린다.

<51> 도 1의 (c)는 동작주파수 $0.7f_0$ 인 경우의 프로세서에 있어서의 태스크의 처리상태를 시계열적으로 나타내는 도이다. 본 도면에서는, 곧바로 프레임의 드롭이 생겨 버려 게임으로서의 기능을 달성할 수 없게 된다. 이와 같이, 동작주파수가 저하한 상태에서 통상시의 동작주파수와 동일하게 태스크를 실행하면, 실시간 태스크의 실시간성을 보장하는 것이 곤란해진다.

<52> 종래, 컴퓨터의 일반적인 사용방법에서는 워드프로세서 소프트나 메일 소프트가 병행하여 동작하고 있는 정도여서, 그다지 태스크의 실시간성을 의식할 필요가 없었다. 그러나 범용적인 기능을 함께 가진 게임기가 보급되기 시작하면서, 게다가 프로세서의 열의 문제로부터 프로세서의 동작주파수를 떨어뜨리는 스로틀(throttle) 기술이 필요해졌다는 점에서, 본 발명자는 전술한 CG영상의 프레임 드롭이라는, 과거에는 존재하지 않았던 문제를 인식하기에 이르렀다.

<53> (실시의 형태 1)

<54> 실시의 형태 1 및 실시의 형태 2에서는, 비선점형(non-preemptive)의 태스크 관리가 행해지는 경우, 즉 태스크의 전환이 태스크의 자주성에 맡겨져 있는 경우에 있어서의 태스크 관리방법에 대해서 설명한다. 실시의 형태 1에 관련된 태스크 관리방법은 처리의 단위시간을 실시간성을 보장하는 보호 대역과 실시간성을 보장하지 않는 비보호 대역으로 분할하여, 프로세서의 처리능력이 저하했을 때, 비보호 영역에 있어서 실행해야 할 태스크의 실행을 적의 스킵하는 것이다. 즉, 프로세서의 발열을 억제할 목적으로 동작주파수를 낮게 했을 경우, 비보호 대역에 있어서 실행해야 할 태스크를 베스트 에포트(best-efforts)에서 처리하는 대신에, 보호 대역에서 실행해야 할 태스크의 실시간성을 보장하는 것이다.

<55> "보호 대역"은 단위시간에 있어서 보장해야 할 프로세서의 실행 사이클 수로서 고정값이다. "비보호 대역"은 단위시간에 있어서 보장하지 않는 프로세서의 실행 사이클 수이며, 0부터 소정의 값까지 소정의 범위를 갖는 변동값이다. 프로세서의 처리능력이 최대일 때, 그 실행 사이클 수는 보호 대역에 있어서의 실행 사이클 수와 비보호 대역에 있어서의 최대의 실행 사이클 수와의 합계값이다. 예를 들면, 프로세서의 열 제어를 위해 동작주파수를 낮게 하는 경우, 실행 사이클 수의 변동이 비보호 대역에 있어서 취할 수 있는 실행 사이클 수의 범위에 포함되도록 동작주파수를 낮게 한다. 역으로 말하면, 이 범위에서 동작주파수를 제어하면, 보호 대역에서 실행해야 할 태스크의 실시간성이 보장된다.

<56> 실시의 형태 1에 관련된 태스크 관리방법은 실시간 태스크와 비 실시간 태스크가 혼재하는 때에 실시간 태스크를 보호 대역에 할당하고, 비 실시간 태스크를 비보호 대역에 할당하여 태스크를 실행하는 것이다. 그리고 프로세서의 열 제어를 위해 행하는 동작주파수의 제어는, 동작주파수의 제어에 수반하는 실행 사이클 수의 변동이 비보호 대역에 포함되는 범위에서 행해진다. 이에 따라, 프로세서의 열 제어를 위해 행하는 동작주파수의 제어를 프로그램 설계와 분리해서 행할 수 있다. 즉, 프로그램 설계의 자유도를 제한하는 일 없이 유연하게 열 제어를 행할 수 있다. 따라서, 프로세서의 처리능력을 충분히 발휘시킬 수 있고, 또한 동작주파수를 바꾸는데 따르는 열 제어도 효율적으로 행할 수 있다.

<57> 도 2의 (a)는 동작주파수가 f_0 인 경우, 즉 프로세서의 발열을 억제할 필요가 없는 통상의 동작주파수 때의 태스크의 실행상태를 나타내는 도이다. 동작주파수가 f_0 인 경우, 각각의 프레임 기간에 있어서 A, B, C의 3개의 태스크가 실행된다. 태스크 A는 보호 대역에 할당되어야 할 실시간 태스크이다. 태스크 B 및 C는 비보호 대역에 할당되어야 할 비 실시간 태스크이다.

<58> 도 2의 (b)는 동작주파수가 $0.7f_0$ 인 때의 태스크의 실행상태를 나타내는 도이다. 실시간 태스크 A2 및 A3는 각각 T2 및 T3를 지나서 처리가 완료되어 있다. 이러면 프레임의 드롭이 발생해 버린다.

<59> 도 2의 (c)는 본 실시의 형태에 관련된 태스크 관리방법에 기초하여 태스크를 관리한 경우의, 동작주파수가 $0.7f_0$ 인 때의 태스크의 실행상태를 나타내는 도이다. 본 도면에서는 비 실시간 태스크(C1)가 스킵되어 있다. 이처럼 비 실시간 태스크를 실행하지 않음으로써 후속의 실시간 태스크를 실행하는 타이밍이 빨라진다. 이에 따라, 실시간 태스크의 실시간성을 보장할 수 있다. 즉 본 실시의 형태에서는 보호 대역이 $0.7f_0$ 로 설정되어 있

기 때문에, 비보호 대역이 0이 된다. 이 때문에, $0.7f_0$ 를 경계로 태스크 관리의 제어가 전환되어 있다.

- <60> 도 3은 도 2의 (c)를 사용하여 설명한 태스크 관리방법을 이용하여 태스크를 실행하는 프로세서 시스템(10)의 구성도이다. 이 프로세서 시스템(10)이 게임기에 탑재된다. 프로세서 시스템(10)은 프로세서인 반도체 집적회로(100)와 메인 메모리(14)를 포함하며, 이들은 버스(12)에 접속되어 있다. 버스(12)는 어드레스 버스, 데이터 버스 및 컨트롤 버스를 포함한다. 반도체 집적회로(100)는 주처리부(120), 내부 클럭 생성부(130), 태스크 관리부(150), 및 주파수제어부(110)를 갖는다. 메인 메모리(14)는 반도체 집적회로(100)에서 실행되는 프로그램(16)과, 각 프로그램(16)을 실행함으로써 얻어진 연산결과(18)를 보유한다. 프로그램(16)이 반도체 집적회로(100)에 읽어 들여짐으로써 태스크가 실행된다.
- <61> 반도체 집적회로(100)는, 예를 들면 도시하지 않은 캐쉬 메모리, 명령 레지스터, 연산 레지스터, 디코더, 제어부, 연산기 등의 회로를 포함하며, 그들의 회로를 이용하여 태스크를 실행한다. 캐쉬 메모리 또는 메인 메모리(14)에 기억되어 있는 명령은 인출되어 명령 레지스터에 받아들여진다. 디코더는 명령 레지스터에 보유된 명령을 디코드하여 오퍼레이션 코드에 대응하는 제어신호를 제어부에 공급한다. 제어부는, 예를 들면 제어신호에 기초하여, 그 오퍼레이션 코드에 대한 처리를 실행하기 위한 연산기를 선택하고, 피연산자로 지정된 어드레스로부터 연산에 필요한 데이터를 받아들여 연산 레지스터에 기록한다. 연산기는, 예를 들면 연산 레지스터에 보유되어 있는 데이터를 이용하여 연산처리를 행하고, 피연산자로 지정된 어드레스에 기록한다.
- <62> 본 도면 및 이로부터 설명하는 각 도면은 하드웨어 단위의 구성이 아니라, 기능 단위의 구성을 나타내고 있다. 또한 각 기능 블록은 동시에 반도체 집적회로(100)의 내부에 정적으로 형성되어 있을 필요는 없고, 일정한 기간에 동적으로 형성되면 된다. 본 실시의 형태에서는, 태스크 관리부(150) 및 주파수제어부(110)는, 예를 들면 오퍼레이팅 시스템에 내장된 프로그램이 실행됨으로써 반도체 집적회로(100)에 동적으로 형성되는 것으로 한다. 또한 주처리부(120)는, 본래는 반도체 집적회로(100) 전체로 생각할 수도 있으나, 여기서는 이해를 돕기 위해 주파수제어부(110), 내부 클럭 생성부(130), 태스크 관리부(150)를 제외한 기능부분을 가리키는 것으로 한다.
- <63> 베이스 클럭 공급부(30)는 반도체 집적회로(100)에 대해 베이스 클럭을 공급한다. 내부 클럭 생성부(130)는, 예를 들면 PLL(Phase Locked Loop)을 포함하고, 베이스 클럭의 주파수의 정수배의 클럭을 생성한다. 내부 클럭 생성부(130)가 생성하는 클럭을 동작 클럭이라고 부르고, 동작 클럭의 주파수를 동작주파수라고 부른다. 반도체 집적회로(100)에 포함되는 각 회로는, 동작 클럭의 상승 또는 하강의 타이밍에서 동작한다. 내부 클럭 생성부(130)는, PLL을 구성하는 회로에 포함되는 카운터의 카운터값을 조정함으로써 동작주파수를 바꿀 수 있다.
- <64> 온도센서(102)는 주처리부(120) 또는 그 주변회로의 온도를 측정하고, 측정된 온도를 주파수제어부(110)에 출력한다. 온도센서(102)는 반도체 집적회로(100)의 외부에 설치되어도 되며, 반도체 집적회로(100)의 내부, 즉 다이(die) 상에 설치되어도 된다.
- <65> 주파수제어부(110)는 반도체 집적회로(100)의 온도에 따라 열 제어를 위해 필요한 동작주파수를 산출하고, 그 주파수에서 동작 클럭을 생성하도록 내부 클럭 생성부(130)를 제어한다. 주파수제어부(110)는 온도가 소정의 역치값보다 높아진 경우에, 동작주파수를 낮게 하도록 내부 클럭 생성부(130)를 제어한다. 동작주파수를 낮게 함으로써, 반도체 집적회로(100)에 있어서의 발열량을 억제할 수 있으며, 나아가 히트싱크 등의 방열(放熱)기구가 작용하여 반도체 집적회로(100)의 온도를 낮출 수 있다.
- <66> 열 제어를 위해 조정하는 동작주파수의 범위는 보호 대역보다 낮아지지 않는 것이 바람직하고, 반도체 집적회로(100)의 온도에 따라 단계적으로 동작주파수를 조정해도 된다. 실행해야 할 태스크가 적을 때나 반도체 집적회로(100)의 발열량이 많아서 비보호 대역만으로는 온도를 충분히 낮출 수 없는 비상사태라고 판단한 경우, 주파수제어부(110)는 동작주파수를 보호 대역보다 낮게 해도 된다. 이러한 발열량에 대한 동작주파수의 제어방법은 여러 가지가 있으며, 주파수제어부(110)는 임의의 방법에 기초하여 동작주파수를 산출해도 된다.
- <67> 예를 들면, 주파수제어부(110)는 반도체 집적회로(100)의 온도와 동작주파수를 대응시킨 테이블을 참조하여 동작주파수를 산출한다. 그리고 주파수제어부(110)는 내부 클럭 생성부(130)에 그 동작주파수의 동작 클럭을 생성시킨다.
- <68> 주처리부(120)는 태스크 관리부(150)에 지시된 태스크에 대응하는 프로그램(16)을 메인 메모리(14)로부터 읽어 들여 태스크를 실행한다. 그리고 주처리부(120)는 그 태스크를 실행함으로써 얻은 연산결과(18)를 메인 메모리(14)에 기록한다.
- <69> 태스크 관리부(150)는 주파수제어부(110)로부터 주처리부(120)의 동작주파수를 특정하는 주파수 정보를 접수하

고, 그 정보에 기초하여 전술한 태스크 관리방법에 따라 태스크의 스케줄링을 행한다.

- <70> 도 4는 도 3의 태스크 관리부(150)의 내부 구성도이다. 주과수검출부(152)는 동작주과수를 검출한다. 본 실시의 형태에서는, 주과수검출부(152)는, 주과수제어부(110)로부터 주과수 정보를 접수함으로써 동작주과수를 파악한다. 주과수검출부(152)는 주과수 정보를 전환지시부(154)에 출력한다.
- <71> 전환지시부(154)는 스케줄 작성부(156) 및 지시부(158)를 가지며, 주처리부(120)에 실행시키는 태스크의 전환지시를 행한다. 상세하게는 후술하겠으나, 스케줄 작성부(156)는, 제어목표 테이블(160)과 태스크 테이블(164)을 참조하여 동작주과수에 따른 태스크의 스케줄링을 행한다. 지시부(158)는 스케줄 작성부(156)에서 생성된 스케줄에 기초하여, 각각의 태스크의 실행지시를 주처리부(120)에 대해 행한다.
- <72> 도 5는 도 3의 제어목표 테이블(160)에 보유되어 있는 데이터를 그래프 표시한 도이다. 제어목표 테이블(160)은 동작주과수와 비 실시간 태스크를 어느 정도 실행할지를 결정하는 제어목표를 대응시켜서 보유한다. 이 제어목표는 실시간 태스크의 실행횟수에 대한 비 실시간 태스크의 실행횟수의 비율이다. 이 비율을, 이하 "실행률"이라고 부른다. 제어목표 테이블(160)에 보유되어 있는 이들 데이터를, 가로축에 동작주과수를 취하고 세로축에 비 실시간 태스크의 실행률을 취하여 그래프로 하면, 본 도면과 같이 0.7f₀를 경계로 하여 다른 양상이 된다. 본 실시의 형태에서는 보호 대역이 0.7f₀로 설정되어 있다. 비보호 대역인 f₀에서 0.9f₀인 때는 비 실시간 태스크의 실행률이 100%이며, 0.9f₀에서 0.7f₀인 때는, 100%에서 10%까지 비 실시간 태스크의 실행률이 선형으로 감소된다. 또한 동작주과수가 보호 대역인 0.7f₀보다 낮아지면 비 실시간 태스크의 실행률이 10%가 된다. 0.7f₀보다 낮아진 경우, 실시간 태스크의 실시간성을 보증한다고 하는 의미에서 비 실시간 태스크의 실행률을 0%로 해도 되지만, 본 실시의 형태에서는 비 실시간 태스크가 전혀 실행되지 않는다고 하는 상황을 피하기 위해 실행률이 10%로 되어 있다. 예를 들면 실행률 "30%"는, 실시간 태스크를 10회 실행하는 데 대해 비 실시간 태스크를 3회 실행하는 것을 나타낸다. 도 3의 스케줄 작성부(156)는 이 제어목표에 기초하여 비 실시간 태스크를 실행하는 타이밍을 조정한다.
- <73> 도 6은 도 4의 태스크 테이블(164)의 데이터 구조의 일례를 나타내는 도이다. 태스크 테이블(164)은, 속성 필드(184), 태스크 필드(186) 및 우선도 필드(188)를 대응시켜서 갖는다. 속성 필드(184)는 태스크의 속성(屬性)을 나타내는 속성정보를 보유한다. 본 실시의 형태에서는, 속성으로서 실시간 태스크와 비 실시간 태스크가 있다. 본 도면에서 "RT"는 실시간 태스크를 나타내며, "NRT"는 비 실시간 태스크를 나타낸다. 태스크 필드(186)는, 예를 들면 프로그램의 파일명이나 태스크의 식별정보 등 태스크를 특정하기 위한 정보를 보유한다. 본 도면에서, 예를 들면 "랜더링" 태스크는 실시간 태스크로서 등록되며, "브라우저" 태스크는 비 실시간 태스크로서 등록되어 있다. 우선도 필드(188)는 태스크의 우선도를 나타내는 정보를 보유한다. 본 도면에서는, 우선도는 10단계로 나타나 있으며, "10"이 가장 우선도가 높고, "1"이 가장 우선도가 낮다.
- <74> 도 4로 돌아가서, 등록부(162)는, 예를 들면 사용자가 프로그램의 실행지시를 한 경우에, 프로그램 코드에 포함되는 속성정보를 해석하고, 그 프로그램을 특정하는 정보에 대응시켜서 태스크 테이블(164)에 등록한다. 속성정보가 프로그램 코드에 포함되어 있지 않은 경우, 등록부(162)는 그 프로그램을 실시간 태스크로서 태스크 테이블(164)에 등록한다. 이에 따라, 예를 들면 속성정보를 포함하지 않는 오래된 프로그램도 실행할 수 있다.
- <75> 또한 다른 예에서는, 등록부(162)는 각 태스크로 실행되는 프로그램의 성질을 판단하고, 그 판단에 기초하여 그 프로그램의 속성을 추정하여 태스크 테이블(164)에 등록해도 된다. 등록부(162)는 그 프로그램에 포함되는 명령, 그 프로그램에 의한 프로세서의 점유율이나 점유시간 등의, 태스크를 실행함으로써 간접적으로 얻어지는 특징에 기초하여 속성을 추정한다. 예를 들면 등록부(162)는, 비선점형의 태스크 관리가 행해지는 경우는, 일정 기간에 있어서의 프로세서의 점유시간을 태스크마다 측정해 점유시간이 긴 태스크를 실시간 태스크로서 추정하고, 점유시간이 짧은 태스크를 비 실시간 태스크로서 추정해도 된다. 이에 따라, 프로그램 코드에 속성정보가 포함되어 있지 않은 프로그램에 대해서도 적절한 태스크 관리를 행할 수 있다.
- <76> 도 7은 도 4의 스케줄 작성부(156)의 내부 구성도이다. 제1계획부(170)는 태스크 테이블(164)을 참조하여 실시간 태스크의 실행순서를 결정한다. 실시간 태스크가 복수 있는 경우, 제1계획부(170)는 도 6의 우선도 필드(188)를 참조하여 우선도가 높은 실시간 태스크가 우선이 되도록 실행순서를 결정한다. 그리고 제1계획부(170)는 실시간 태스크의 실행순서를 통합부(178)에 출력한다.
- <77> 제2계획부(172)는 태스크 테이블(164)을 참조하여 비 실시간 태스크의 실행순서를 결정하고, 실행순서를 통합부(178)에 출력한다. 제2계획부(172)는 설정부(174), 제1카운터(176a), 및 제2카운터(176b)를 갖는다. 제1카운터

(176a) 및 제2카운터(176b)를 총칭하여 카운터(176)로 부른다. 카운터(176)는 비 실시간 태스크의 스케줄링 처리를 행할 때마다 계수를 행하고, 설정된 비율로 비 실시간 태스크의 실행을 허가한다.

- <78> 예를 들면, 제1카운터(176a)는 25%의 비율로 비 실시간 태스크의 실행을 허가하는 카운터이며, 실시간 태스크를 3회 실행하는 데 대해 비 실시간 태스크를 1회 실행하는 것을 허가한다. 또한 허가의 타이밍은 임의로 결정할 수 있다. 본 도면에서는, 제1카운터(176a)의 허가 타이밍은 "○×××"로서 도시되어 있다. "○"가 허가하는 것을 나타내며, "×"가 허가하지 않는 것을 나타내므로, 제1카운터(176a)는 최초로 비 실시간 태스크의 실행을 허가한 후, 3회 스킵하는 것을 나타낸다.
- <79> 설정부(174)는 주과수 검출부(152)로부터 현재의 동작주과수를 접수하고, 그 동작주과수에 따른 비 실시간 태스크의 제어목표를 제어목표 테이블(160)로부터 읽어들인다. 그리고 설정부(174)는 제어목표에 따라 카운터(176)가 비 실시간 태스크의 실행을 허가하는 비율을 설정한다. 예를 들면, 25%로 비 실시간 태스크를 실행하는 경우, 설정부(174)는 카운터(176)에 4회에 1회의 비율로 비 실시간 태스크의 실행을 허가하도록 설정한다.
- <80> 비 실시간 태스크가 복수 있는 경우, 설정부(174)는 비 실시간 태스크마다 카운터(176)를 설정하여, 복수의 비 실시간 태스크의 실행횟수를 합계한 값이 제어목표가 되도록 각각의 카운터(176)의 설정을 행한다. 예를 들면, 비 실시간 태스크가 2개 있고, 25%의 비율로 비 실시간 태스크를 실행하는 경우, 설정부(174)는 제1카운터(176a) 및 제2카운터(176b)의 각각에 실시간 태스크를 8회 실행하는 데 대해 비 실시간 태스크의 실행을 1회 허가하도록 설정한다. 더욱이 설정부(174)는 2개의 비 실시간 태스크를 실행하는 타이밍이 겹치지 않도록 허가 타이밍을 설정한다. 즉 제1카운터(176a)의 허가 타이밍을 "×○××××××"라고 하고, 제2카운터(176b)의 허가 타이밍을 "××××××○×"라고 한다.
- <81> 또한, 예를 들면 50%의 비율로 비 실시간 태스크를 실행하는 경우, 설정부(174)는 허가 타이밍을 "○○××"가 아니라, "○×○×"와 같이 "○"와 "×"를 분리하여, 즉 비 실시간 태스크의 실행을 허가하는 타이밍을 분산하도록 허가 타이밍을 설정한다. 이와 같이 비 실시간 태스크의 실행을 허가하는 타이밍을 분산함으로써, 비 실시간 태스크가 보다 스무스하게 채감된다.
- <82> 통합부(178)는 제1계획부(170)로부터 공급되는 실시간 태스크의 실행순서와, 제2계획부(172)로부터 공급되는 비 실시간 태스크의 실행순서를, 실시간 태스크의 실행순서가 먼저가 되도록 나열하여 스케줄을 작성한다. 그리고 통합부(178)는 작성한 스케줄을 지시부(158)에 출력한다.
- <83> 도 8은 도 4의 태스크 관리부(150)에 있어서의 태스크를 스케줄링하는 처리의 플로우 차트이다. 도 4의 주과수 검출부(152)는 동작주과수를 검출한다(S10). 도 4의 스케줄 작성부(156)는 동작주과수에 기초하여 도 4의 제어 목표 테이블(160)을 참조하고 실시간 태스크와 비 실시간 태스크의 실행순서를 스케줄링한다(S12). 도 4의 지시부(158)는 스케줄 작성부(156)에 스케줄링된 순서로 태스크의 실행지시를 행한다(S14).
- <84> (실시의 형태 2)
- <85> 도 9는 실시의 형태 2에 관련된 태스크 관리부(150)의 내부 구성도이다. 실시의 형태 2는 반도체 집적회로(100)의 사용률에 따라 제어목표 테이블(160)을 최적화하는 형태이다. 이미 설명한 구성과 동일한 부호를 붙인 본 도면의 구성은, 기능 및 동작이 이미 설명한 구성과 거의 동일하다. 이하, 이미 설명한 구성에 있어서의 기능과는 다른 점을 중심으로 설명한다.
- <86> 프로세서 사용률 검출부(190)는, 도 3의 반도체 집적회로(100)의 사용률을, 예를 들면 프레임 기간마다 검출한다. 그리고 프로세서 사용률 검출부(190)는 사용률을 갱신부(192)에 출력한다. 갱신부(192)는 소정의 기간에 있어서의 사용률의 평균값을 산출하고, 그 평균값에 기초하여 비 실시간 태스크의 실행률을 증대하도록 제어목표 테이블(160)을 최적화한다. 예를 들면 갱신부(192)는 0.5초간, 즉 30프레임 분의 사용률의 평균값을 산출한다. 그리고 평균값이 역치값보다 낮을 경우, 즉 부하가 비교적 적은 경우, 갱신부(192)는 제어목표 테이블(160)에 있어서의 비 실시간 태스크의 실행률을 높인다. 평균값이 역치값보다 높을 경우, 갱신부(192)는 제어목표 테이블을 디폴트(default)값으로 되돌린다. 역치값이나 비 실시간 태스크의 실행률의 높임 폭은 실험에 의해 적절한 값이 설정되어도 되며, 실행에 수반하여 서서히 반영되어도 된다.
- <87> 도 10은 도 9의 갱신부(192)에 의해 최적화된 제어목표 테이블(160)에 보유되어 있는 테이터를 그래프 표시한 도이다. 본 도면에서는, 보호 대역이 0.5f₀까지 시프트되고, 0.9f₀에서 0.5f₀인 때는 100%에서 10%까지 비 실시간 태스크의 실행률이 선형으로 감소된다. 이와 같이, 프로세서의 사용률에 따라 보호 대역을 조정함으로써 프로세서의 처리능력을 유효하게 이용할 수 있다.

- <88> 도 11은 도 9의 태스크 관리부(150)에 있어서의 태스크를 스케줄링하는 처리의 플로우 차트이다. 도 9의 태스크 관리부(150)에 있어서의 스케줄링 처리는, 도 8을 이용하여 설명한 도 4의 태스크 관리부(150)에 있어서의 스케줄링 처리에, 제어목표 테이블을 최적화하는 처리(S20)를 부가함으로써 실현된다.
- <89> 도 12는 도 11의 제어목표 테이블을 최적화하는 처리(S20)의 상세한 플로우 차트이다. 도 9의 프로세서 사용률 검출부(190)는 도 3의 반도체 집적회로(100)의 사용률을 검출한다(S22). 도 9의 갱신부(192)는 소정의 기간이 경과했는지 여부를 판정한다(S24). 소정의 기간이 경과한 경우(S24의 Y), 갱신부(192)는 그 기간에 있어서의 사용률의 평균값을 산출한다(S26). 그리고 사용률이 소정의 역치값보다 높을 경우(S28의 Y), 디폴트의 제어목표 테이블로 되돌린다(S34). 스텝 28에서 사용률이 소정의 역치값보다 낮을 경우(S28의 N), 갱신부(192)는 제어목표 테이블을 비 실시간 태스크의 실행률을 높이도록 변경한다(S30). 스텝 24에서 소정의 기간이 경과하고 있지 않은 경우(S24의 N), 도 11의 스텝 10으로 진행한다. 이와 같이 제어목표 테이블(160)을 주처리부(120)의 사용률에 따라 최적화함으로써 주처리부(120)의 처리능력을 끌어낼 수 있다.
- <90> (실시의 형태 3)
- <91> 실시의 형태 3 및 실시의 형태 4에서는, 선점형(preemptive)의 태스크 관리가 행해져 있는 경우, 즉 타이머 인터럽트에 의해 강제적으로 태스크의 전환이 행해지는 경우에 있어서의 태스크 관리방법에 대해서 설명한다. 이하, 프레임 기간을 "t"로 한다. 실시의 형태 3에 관련된 태스크 관리방법은, 비 실시간 태스크를 비보호 대역으로서 할당된 기간에 실행하는 것이다. 이하, 보호 대역으로서 할당된 기간을 "보호기간(tr)"이라고 하고, 비보호 대역으로서 할당된 기간을 "비보호기간(tn)"이라고 한다. 복수의 비 실시간 태스크가 있는 경우에는, 비보호기간(tn)을 등분하여 각각의 비 실시간 태스크가 실행된다.
- <92> 도 13의 (a)는 동작주파수가 f_0 인 경우의 태스크의 실행상태를 나타내는 도이다. 동작주파수가 f_0 인 때, 보호기간(tr)은 $0.7t$ 이고 비보호기간(tn)은 $0.3t$ 이다. 비 실시간 태스크가 2개 있으므로, 각각의 비 실시간 태스크는 $0.15t$ 의 기간씩 순서대로 실행된다.
- <93> 도 13의 (b)는 동작주파수가 $0.8f_0$ 인 경우의 태스크의 실행상태를 나타내는 도이다. 동작주파수가 $0.8f_0$ 인 때, 동작주파수의 주기는 $t/(0.8 \cdot f_0)$ 이므로 보호기간(tr)은 $0.7f_0 \cdot t/(0.8 \cdot f_0) = 7/8 \cdot t$ 이고, 비보호기간(tn)은 $1/8 \cdot t$ 이다. 비 실시간 태스크가 2개 있으므로, 각각의 비 실시간 태스크는 $1/16 \cdot t$ 의 기간씩 순서대로 실행된다. 본 도면에서는, 보호기간(tr) 사이를 실시간 태스크(RT)가 점유하고 있으나, 실시간 태스크(RT)의 프로그램은 보호기간(tr)보다 짧은 기간에서 처리가 완료하도록 설계되므로, 일반적으로는 보호기간(tr)의 전부를 점유하는 일은 없다.
- <94> 도 14는 실시의 형태 3에 관련된 태스크 관리부(150)의 내부 구성도의 일례를 나타내는 도이다. 이미 설명한 구성과 동일한 부호를 붙인 본 도면의 구성은, 기능 및 동작이 이미 설명한 구성과 거의 동일하다. 이하, 이미 설명한 구성에 있어서의 기능과는 다른 점을 중심으로 설명한다. 스케줄 작성부(156)는 도 13을 이용하여 설명한 바와 같이 태스크의 스케줄링을 행한다. 스케줄 작성부(156)는 태스크 테이블(164)로부터 실시간 태스크와 비 실시간 태스크를 읽어들이, 실시간 태스크에 이어서 비 실시간 태스크를 실행하도록 스케줄을 작성한다. 또한, 스케줄 작성부(156)는 스케줄링을 위해 제어목표 테이블(160)을 참조하여, 동작주파수에 따른 비 실시간 태스크의 실행시간을 산출한다. 그리고 스케줄 작성부(156)는, 예를 들면 비 실시간 태스크마다 실행시간을 대응시켜서 스케줄을 작성한다. 지시부(158)는 스케줄에 포함되는 실행시간에 기초하여 비 실시간 태스크를 전환하기 위한 인터럽트 타이머를 설정한다.
- <95> 도 15는 도 14의 제어목표 테이블(160)에 보유되어 있는 데이터를 그래프 표시한 도이다. 제어목표 테이블(160)은 동작주파수와, 비 실시간 태스크를 실행하기 위해 프로세서를 점유할 수 있는 시간, 즉 비보호기간(tn)을 대응시켜서 보유한다. 제어목표 테이블(160)에 보유되어 있는 이들 데이터를, 가로축에 동작주파수를 취하고 세로축에 비보호기간(tn)을 취하여 그래프로 하면, 본 도면과 같이 $0.7f_0$ 를 경계로 하여 다른 양상이 된다. 본 실시의 형태에서는 보호 대역이 $0.7f_0$ 로 설정되어 있다. 비보호 대역인 f_0 에서 $0.9f_0$ 인 때는 비보호기간(tn)이 $0.3t$ 이고, $0.9f_0$ 에서 $0.7f_0$ 인 때는 $0.3t$ 에서 $1.0t$ 까지 선형으로 비보호기간(tn)이 짧아져 있다. 또한 동작주파수가 보호 대역인 $0.7f_0$ 보다 낮아지면 보호기간(tr)이 $0.01t$ 이 된다. $0.7f_0$ 보다 낮아진 경우, 실시간 태스크의 실시간성을 보증한다는 의미로서 비보호기간(tn)을 0으로 해도 되지만, 본 실시의 형태에서는 비 실시간 태스크가 전혀 실행되지 않는다고 하는 상황을 피하기 위해 비보호기간(tn)이 $0.01t$ 로 되어 있다. 예를 들면, 비보호기간(tn)이 "0.1t"이고 비 실시간 태스크가 2개 있는 경우, 각각의 비 실시간 태스크의 실행시간은 $0.05t$ 가 된

다. 도 14의 스케줄 작성부(156)는 이 제어목표에 기초하여 비 실시간 태스크의 실행시간을 조정한다.

- <96> (실시의 형태 4)
- <97> 도 16은 실시의 형태 4에 관련된 태스크 관리부(150)의 내부 구성도이다. 실시의 형태 4는, 반도체 집적회로(100)의 사용률에 따라 제어목표 테이블(160)을 최적화하는 형태이다. 이미 설명한 구성과 동일한 부호를 붙인 본 도면의 구성은, 기능 및 동작이 이미 설명한 구성과 거의 동일하다. 이하, 이미 설명한 구성에 있어서의 기능과는 다른 점을 중심으로 설명한다.
- <98> 갱신부(192)는 소정의 기간에 있어서의 반도체 집적회로(100)의 사용률의 평균값을 산출하고, 그 평균값에 기초하여 비보호기간(t_n)을 늘리도록 제어목표 테이블(160)을 최적화한다. 평균값이 역치값보다 작은 경우, 즉 부하가 비교적 적은 경우, 갱신부(192)는 제어목표 테이블(160)에 있어서의 비보호기간(t_n)을 길게 한다. 평균값이 역치값보다 높을 경우, 갱신부(192)는 제어목표 테이블을 디폴트값으로 되돌린다. 갱신의 판단에 이용하는 역치값이나 비보호기간(t_n)을 연장하는 길이는 실험에 의해 적절한 값이 설정되어도 되고, 실행에 수반하여 서서히 반영되어도 된다.
- <99> 도 17은 도 16의 갱신부(192)에 의해 최적화된 제어목표 테이블(160)에 보유되어 있는 데이터를 그래프 표시한 도이다. 본 도면에서는, 보호 대역이 $0.5f_0$ 까지 시프트되고, $0.9f_0$ 에서 $0.5f_0$ 인 때는 $0.3t$ 에서 $0.01t$ 까지 선형으로 비보호기간(t_n)이 짧아져 있다. 이와 같이, 프로세서의 사용률에 따라 보호 대역을 조정함으로써 프로세서의 처리능력을 유효하게 이용할 수 있다.
- <100> 이상, 본 발명을 실시의 형태를 바탕으로 설명했다. 이들의 실시의 형태는 예시이며, 그들의 각 구성요소나 각 처리 프로세스의 조합에 여러 가지 변형예가 가능하다는 것, 또 그러한 변형예도 본 발명의 범위에 있다는 것은 당업자에게 이해되는 부분이다. 이러한 변형예로서, 실시의 형태 1에서 실시의 형태 4에서는, 1프레임 기간에 묘화처리를 완료해야 할 시간적 제약이 있는 태스크를 보호 대역에 할당하고, 시간적 제약이 없는 태스크를 비보호 대역에 할당하는 것으로서 설명했다, 보호 대역 또는 비보호 대역에 할당하는 조건은 이것에 한정되지 않는다.
- <101> 시간적 제약과는 다른 관점에서 태스크를 보호 대역 또는 비보호 대역에 할당해도 되는데, 예를 들면 데이터를 확실하게 기록할 필요가 있는지 여부의 관점에서 태스크를 보호 대역 또는 비보호 대역의 어느 쪽에 할당해도 된다. 이 경우, 예를 들면 프로그램 방송을 녹화하는 태스크와 워드프로세서의 태스크가 있는 경우, 녹화의 태스크를, 데이터를 확실하게 기록한다는 관점에서 보호 대역에 할당하고, 워드프로세서의 태스크를 비보호 대역에 할당해도 된다. 이와 같이, 보호 대역과 비보호 대역에 분배하는 조건을 바꿈으로써, 실시의 형태에서 설명한 태스크 관리방법을, 예를 들면 항공기의 제어용 컴퓨터나 자동차의 제어용 컴퓨터 등 실시간 OS를 탑재하는 전자장치에 채용할 수 있다.
- <102> 다른 변형예로서, 실시의 형태에서는 열 제어라는 관점에서 반도체 집적회로(100)의 동작주파수가 제어되는 것으로 했으나, 예를 들면, 전력소비라는 관점에서 동작주파수가 제어되어도 된다. 소비전력은 클럭의 동작주파수, 회로의 트랜지스터 수, 및 전원전압의 2승에 비례한다. 역으로 말하면, 동작주파수, 부하가 되는 트랜지스터 수, 및 전원전압을 알면 소비전력을 산출할 수 있다. 예를 들면, 도 3의 주파수제어부(110)는 레귤레이터의 출력 전압값을 측정하는 센서로부터 전압값을 취득하여, 그 전압값으로부터 산출되는 소비전력이 소정의 역치값보다 높아진 경우, 열 폭주를 방지하기 위해 동작주파수를 낮게 해도 된다. 또한, 전원 모드가 배터리 모드인지 AC전원 모드인지를 판단하여 배터리 모드의 경우면 절전을 위해 동작주파수를 낮게 해도 된다. 이러한 경우에도, 실시의 형태에서 설명한 태스크 관리방법에 따르면 실시간 태스크의 실시간성이 보증된다.
- <103> 다른 변형예로서, 실시의 형태에서는 주처리부(120)에 있어서의 동작주파수에 따라 비 실시간 태스크의 실행률이 조정되는 것으로 했으나, 주처리부(120) 또는 그 주변회로의 온도에 따라 비 실시간 태스크의 실행률이 조정되어도 된다. 예를 들면, 주처리부(120)에 공급하는 클럭의 동작주파수를 주처리부(120) 또는 그 주변회로의 온도에 따라 소프트웨어와는 독립적으로 제어하는 회로가 설치되어 있는 경우, 동작주파수는 주처리부(120) 또는 그 주변회로의 온도에 따라 하드웨어적으로 제어된다. 이러한 회로에 있어서, 소프트웨어적으로 취득할 수 있는 정보가 주파수 정보가 아니라 온도 정보인 경우, 즉 도 3의 태스크 관리부(150)가 주파수 정보가 아니라 온도 정보를 취득할 수 있는 경우, 온도 정보에 기초하여 태스크의 실행률이 조정되어도 된다. 이 경우, 예를 들면 도 4의 제어목표 테이블(160)은 주처리부(120) 또는 그 주변회로의 온도와 실행률을 대응시켜서 보유하고, 도 4의 스케줄 작성부(156)는 이 테이블에 기초하여 비 실시간 태스크를 실행하는 타이밍을 조정한다.
- <104> 또한, 주처리부(120) 또는 반도체 집적회로(100)의 소비전력에 따라 비 실시간 태스크의 실행률이 조정되어도

된다. 각종 전원관리 소프트웨어 등으로부터 도 3의 태스크 관리부(150)가 주파수 정보가 아니라 소비전력 정보를 취득할 수 있는 경우, 소비전력 정보에 기초하여 태스크의 실행률이 조정되어도 된다. 이 경우, 예를 들면 도 4의 제어목표 테이블(160)은 소비전력 정보와 실행률을 대응시켜서 보유하고, 도 4의 스케줄 작성부(156)는 이 테이블에 기초하여 비 실시간 태스크를 실행하는 타이밍을 조정한다.

산업상 이용 가능성

<105> 본 발명은 프로세서의 태스크 관리의 분야에 적용할 수 있다.

도면의 간단한 설명

<17> 도 1의 (a)는 통상시의 동작주파수에서 동작하는 프로세서에 있어서의 태스크의 처리상태를 시계열적으로 나타내는 도이고, (b)는 동작주파수가 저하한 경우의 프로세서에 있어서의 태스크의 처리상태를 시계열적으로 나타내는 도이며, (c)는 동작주파수가 더 저하한 경우의 프로세서에 있어서의 태스크의 처리상태를 시계열적으로 나타내는 도이다.

<18> 도 2의 (a)는 동작주파수가 f_0 인 경우, 즉 프로세서의 발열을 억제할 필요가 없는 통상의 동작주파수 때의 태스크의 실행상태를 나타내는 도이고, (b)는 동작주파수가 $0.7f_0$ 인 때의 태스크의 실행상태를 나타내는 도이며, (c)는 본 실시의 형태에 관련된 태스크 관리방법에 기초하여 태스크를 관리한 경우의, 동작주파수가 $0.7f_0$ 인 때의 태스크의 실행상태를 나타내는 도이다.

<19> 도 3은 도 2를 사용하여 설명한 태스크 관리방법을 이용하여 태스크를 실행하는 프로세서 시스템의 구성도이다.

<20> 도 4는 도 3의 태스크 관리부의 내부 구성도이다.

<21> 도 5는 도 3의 제어목표 테이블에 보유되어 있는 데이터를 그래프 표시한 도이다.

<22> 도 6은 도 4의 태스크 테이블의 데이터 구조의 일례를 나타내는 도이다.

<23> 도 7은 도 4의 스케줄 작성부(156)의 내부 구성도이다.

<24> 도 8은 도 4의 태스크 관리부에 있어서의 태스크를 관리하는 처리의 플로우 차트이다.

<25> 도 9는 실시의 형태 2에 관련된 태스크 관리부의 내부 구성도이다.

<26> 도 10은 도 9의 갱신부에 의해 최적화된 제어목표 테이블에 보유되어 있는 데이터를 그래프 표시한 도이다.

<27> 도 11은 도 9의 태스크 관리부에 있어서의 태스크 관리처리의 플로우 차트이다.

<28> 도 12는 도 9의 제어목표 테이블 갱신처리의 상세한 플로우 차트이다.

<29> 도 13의 (a)는 동작주파수가 f_0 인 경우의 태스크의 실행상태를 나타내는 도이며, (b)는 동작주파수가 $0.8f_0$ 인 경우의 태스크의 실행상태를 나타내는 도이다.

<30> 도 14는 실시의 형태 3에 관련된 태스크 관리부의 내부 구성도의 일례를 나타내는 도이다.

<31> 도 15는 도 14의 제어목표 테이블에 보유되어 있는 데이터를 그래프 표시한 도이다.

<32> 도 16은 실시의 형태 4에 관련된 태스크 관리부의 내부 구성도이다.

<33> 도 17은 도 16의 갱신부에 의해 최적화된 제어목표 테이블에 보유되어 있는 데이터를 그래프 표시한 도이다.

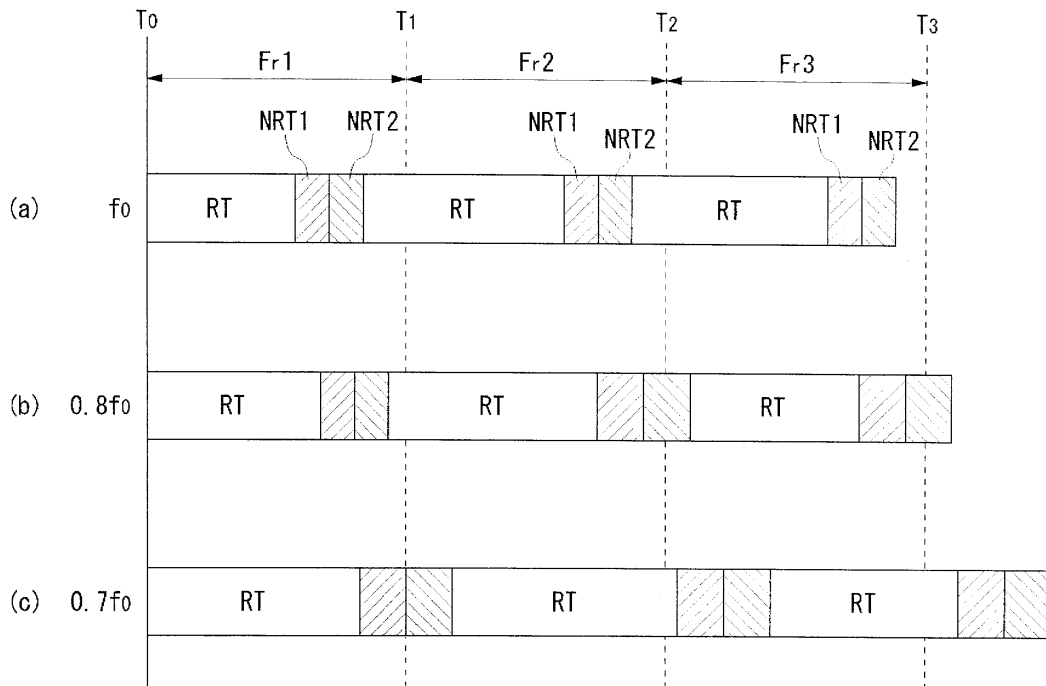
<34> [부호의 설명]

<35>	10	프로세서 시스템	12	버스
<36>	14	메인 메모리	16	프로그램
<37>	18	연산결과	30	베이스 클럭 공급부
<38>	100	반도체 집적회로	102	온도센서
<39>	110	주파수 제어부	120	주처리부

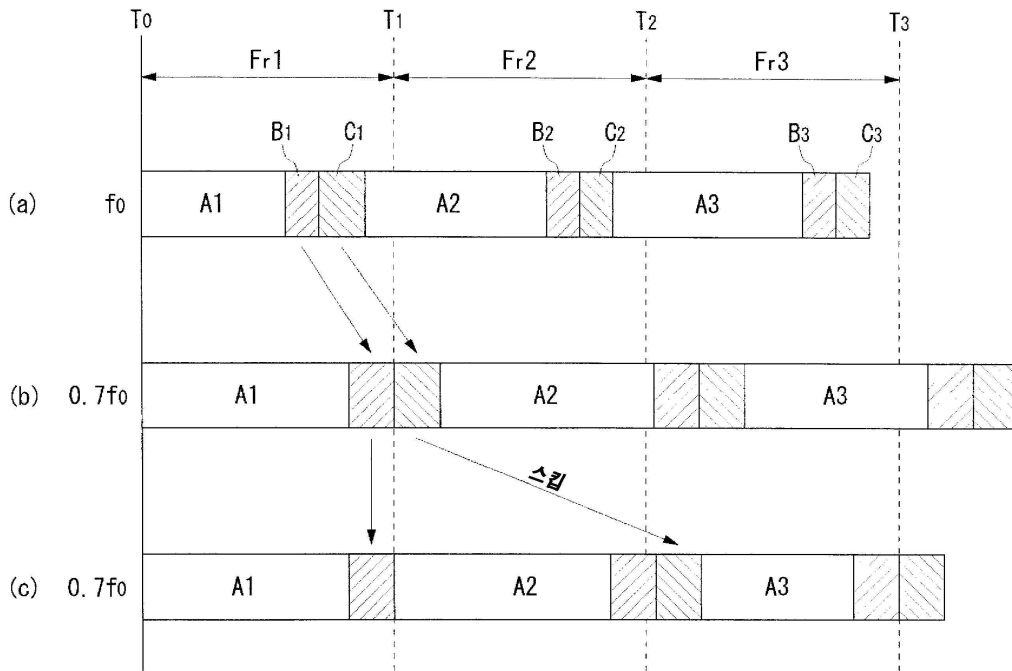
<40>	130	내부 클럭 생성부	150	태스크 관리부
<41>	152	주파수 검출부	154	전환지시부
<42>	156	스케줄 작성부	158	지시부
<43>	160	제어목표 테이블	162	등록부
<44>	164	태스크 테이블	170	제1계획부
<45>	172	제2계획부	174	설정부
<46>	176	카운터	190	프로세서 사용률 검출부
<47>	192	갱신부		

도면

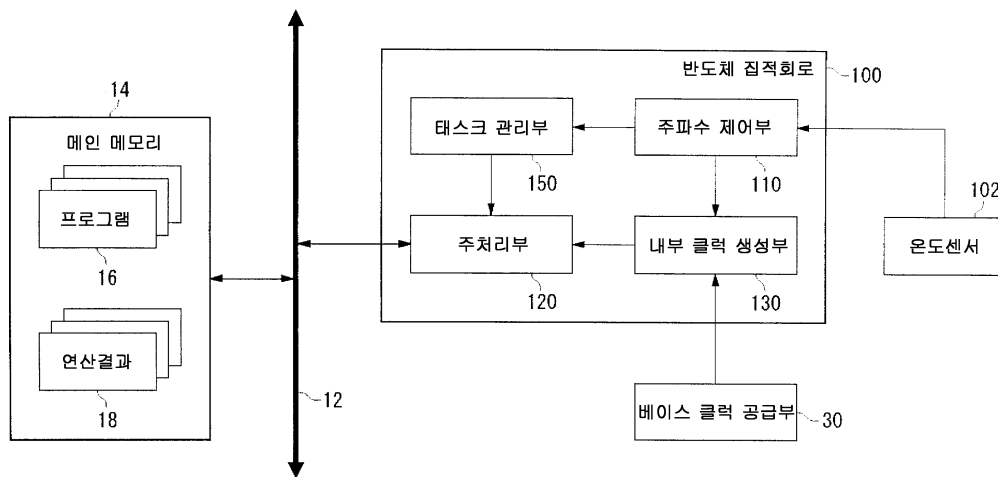
도면1



도면2

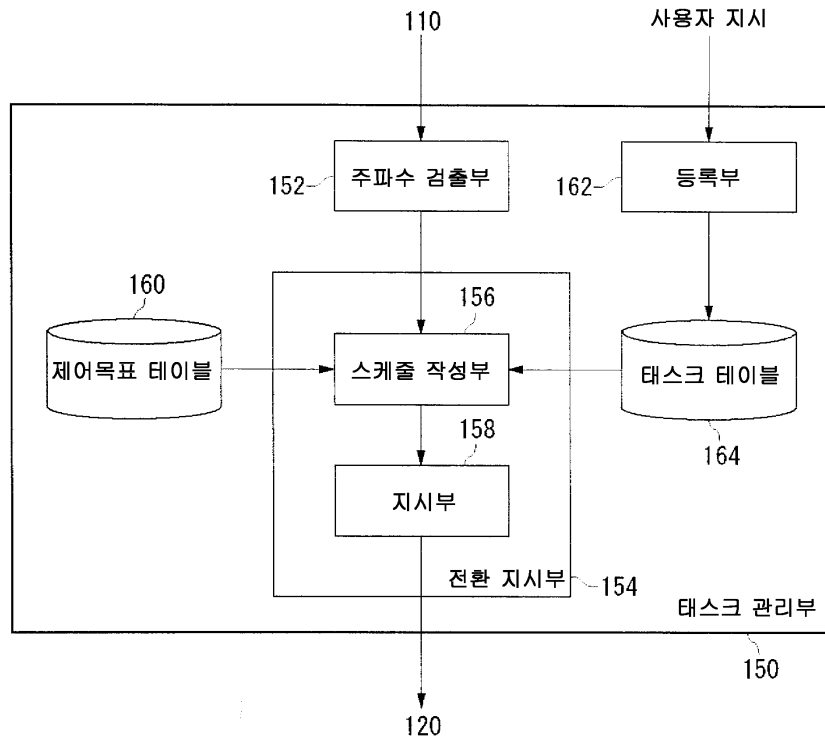


도면3

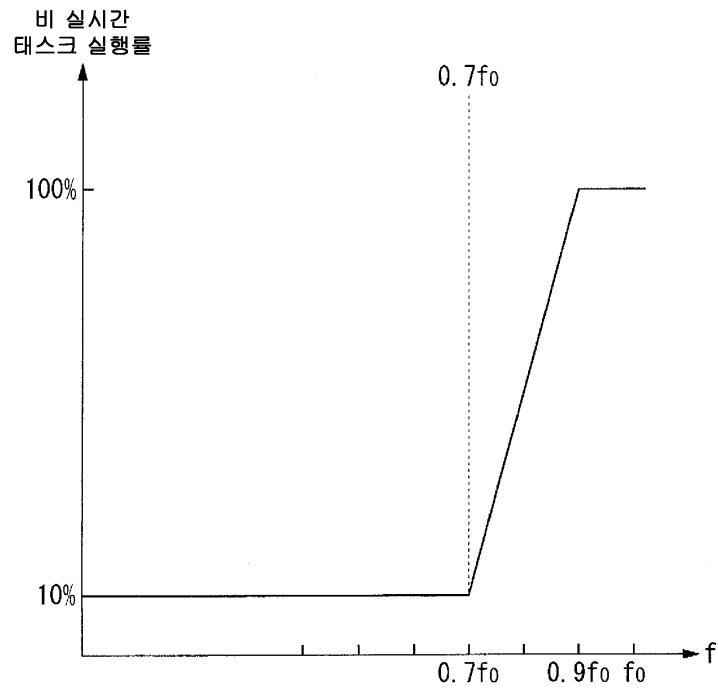


10

도면4



도면5

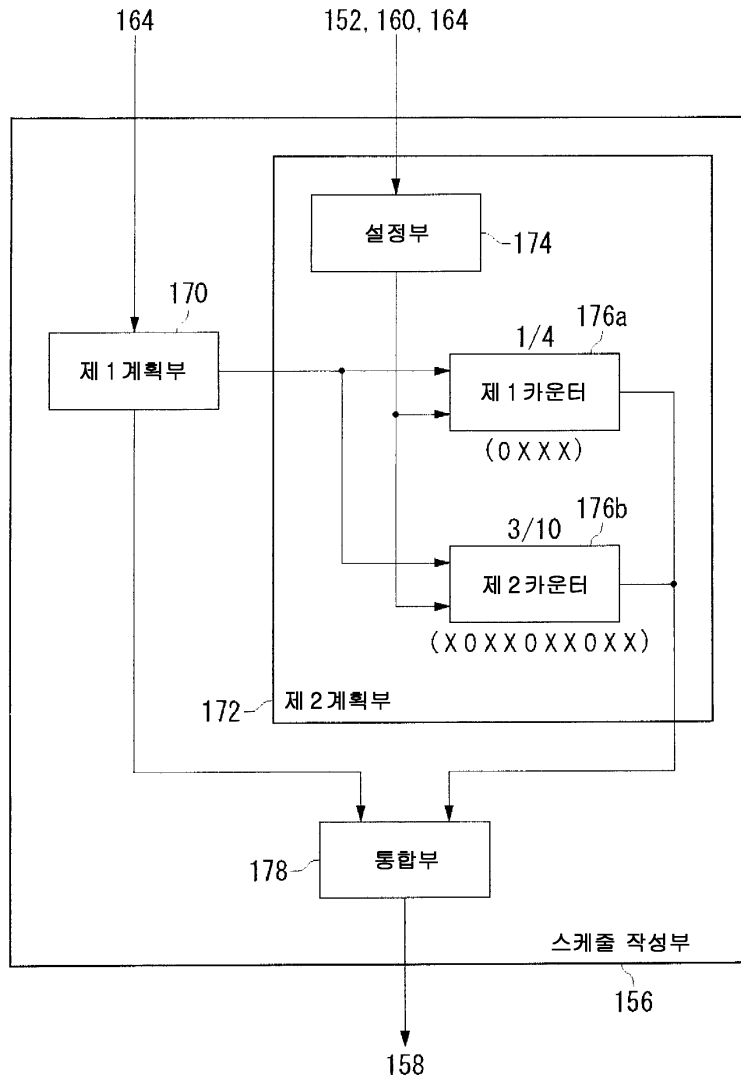


도면6

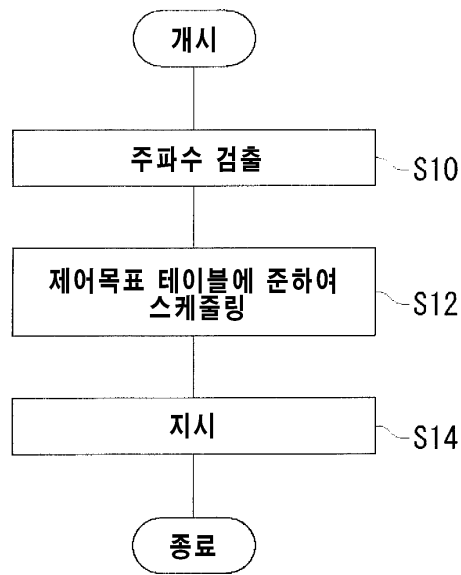
184 속성	186 태스크	188 우선도
⋮	⋮	⋮
RT	랜더링	9
RT	녹화	10
NRT	브라우저	5
⋮ ⋮ ⋮	⋮ ⋮ ⋮	⋮ ⋮ ⋮

164

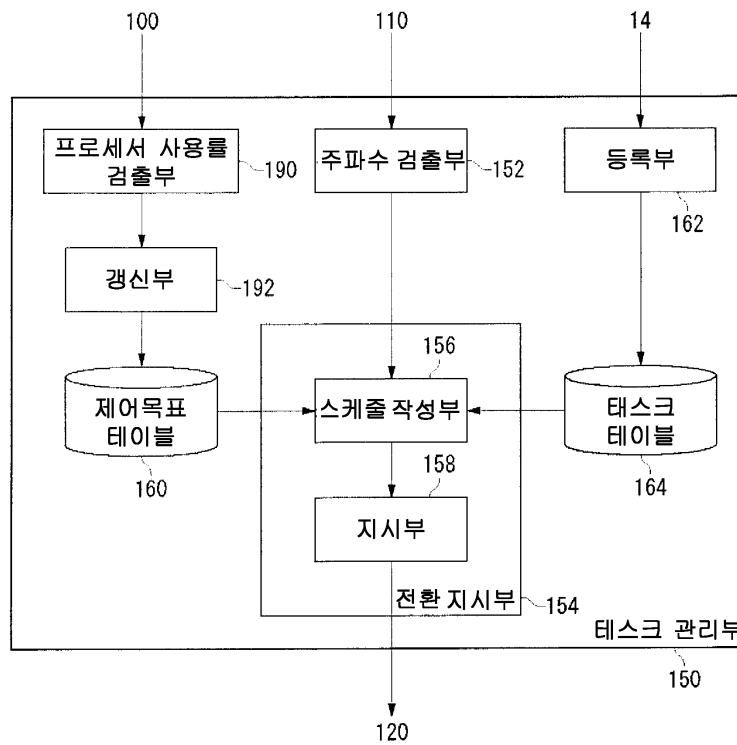
도면7



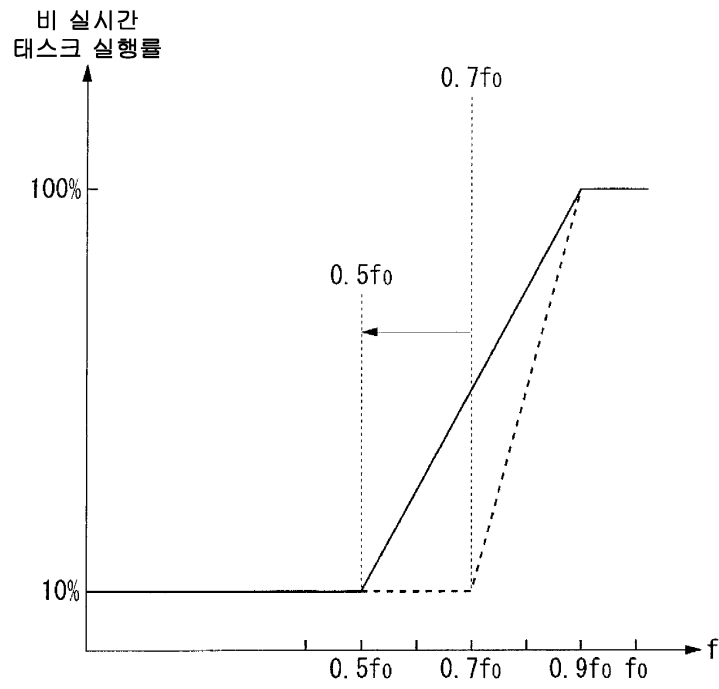
도면8



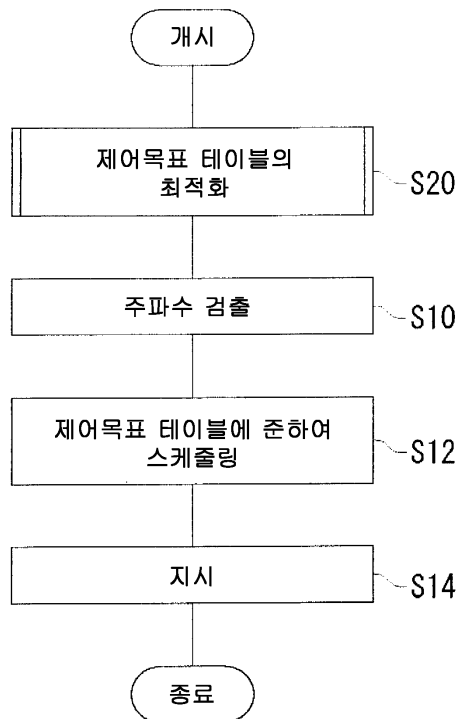
도면9



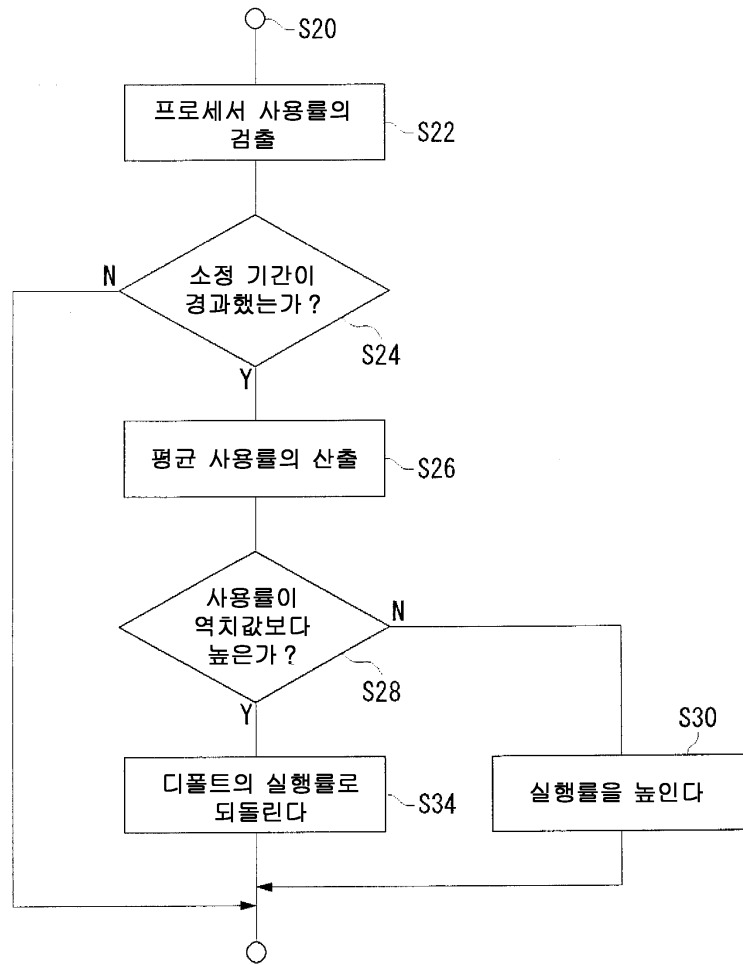
도면10



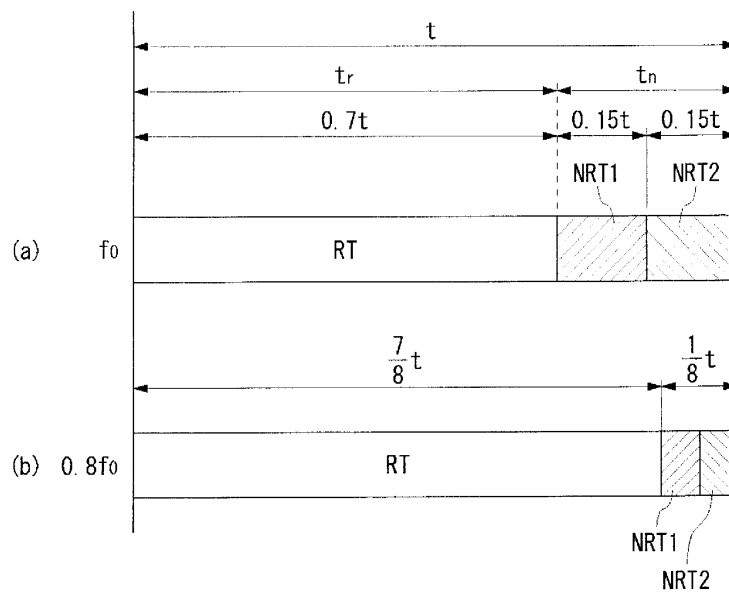
도면11



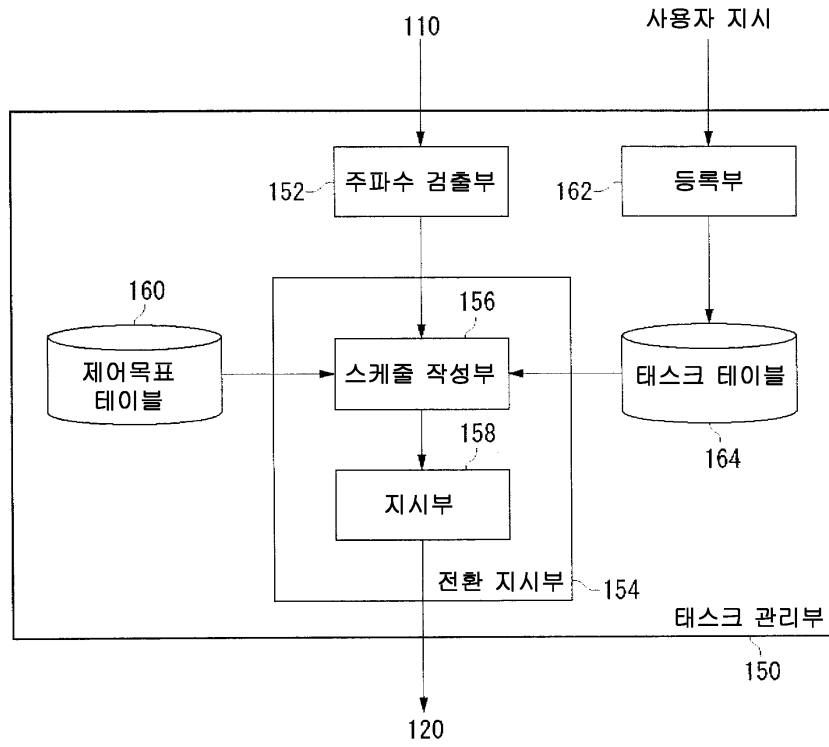
도면12



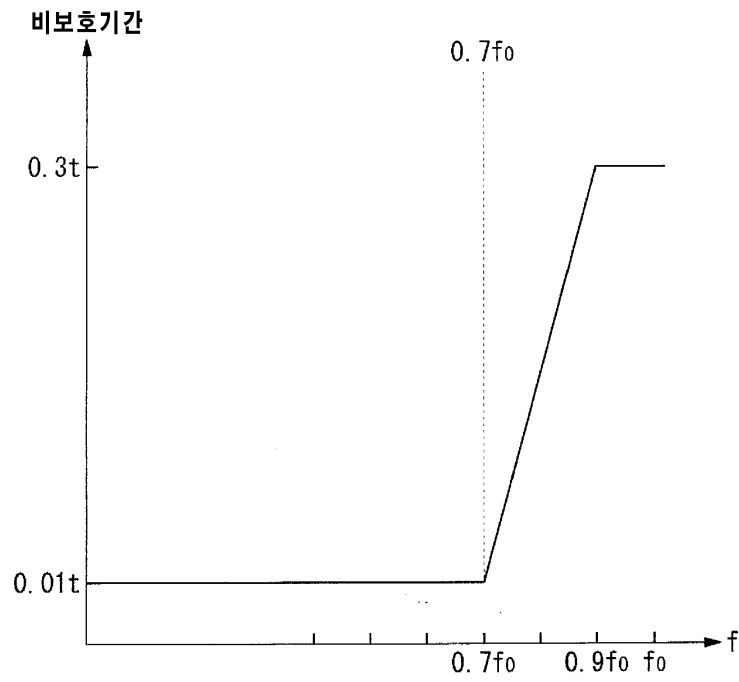
도면13



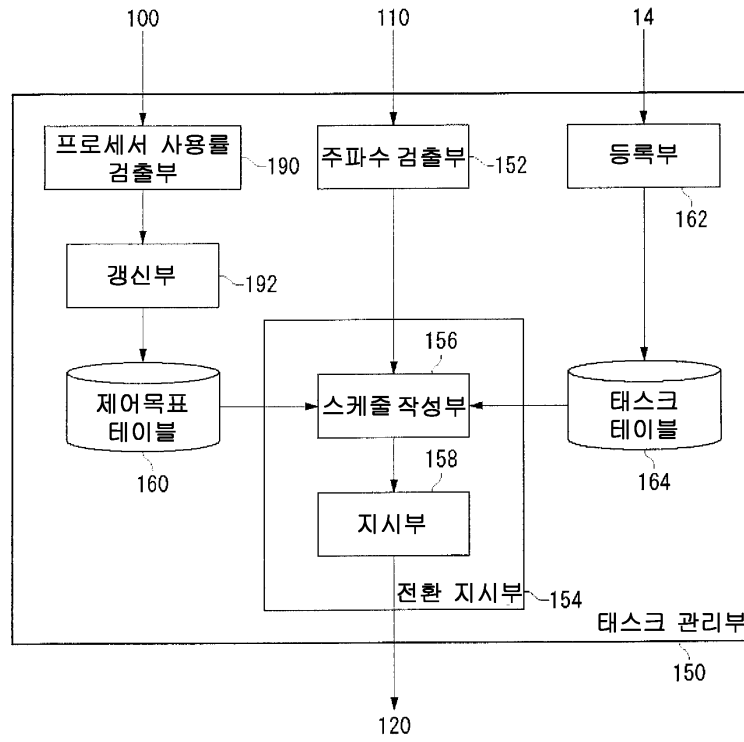
도면14



도면15



도면16



도면17

