(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0032985
(43) 공개일자 2019년03월28일

(51) 국제특허분류(Int. Cl.)
G06F 1/04 (2006.01)
(52) CPC특허분류
G06F 1/04 (2013.01)
(21) 출원번호 10-2018-0017398
(22) 출원일자 2018년02월13일
심사청구일자 2018년02월13일
(30) 우선권주장
JP-P-2017-179976 2017년09월20일 일본(JP)

(71) 출원인
가부시끼가이샤 도시바
일본국 도꾜도 미나또꾸 시바우라 1쪼메 1방 1고
도시바 디바이스 앤 스토리지 가부시키가이샤
일본국 도쿄도 미나토쿠 시바우라 1쵸메 1반 1고우
(72) 발명자
가미야 사토시
일본 도쿄도 미나토쿠 시바우라 1쵸메 1반 1고우 도시바 디바이스 앤 스토리지 가부시키가이샤 지적재산실 내
(74) 대리인
장수길, 김승식, 박충범

전체 청구항 수 : 총 14 항

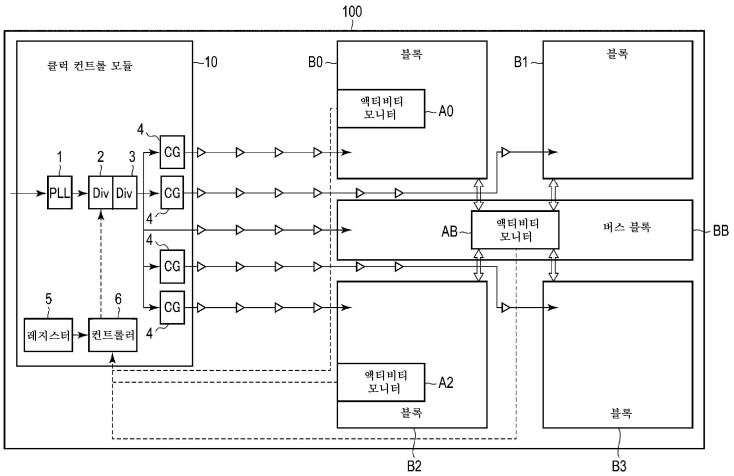
(54) 발명의 명칭 **클럭 생성 회로 및 클럭 생성 방법**

(57) 요 약

실시 형태는, 원하는 주파수의 클럭을 즉시 생성하는 것이 가능한 클럭 생성 회로 및 클럭 생성 방법을 제공한다.

실시 형태의 클럭 생성 회로는, 소스 클럭의 펄스 열의 일부를 마스크하는 처리를 행함으로써 당해 소스 클럭보다 주파수가 낮은 분주 클럭을 생성하는 것이 가능한 분주기를 구비한다.

대 표 도



명 세 서

청구범위

청구항 1

소스 클럭의 펄스 열의 일부를 마스크하는 처리를 행함으로써 당해 소스 클럭보다 주파수가 낮은 분주 클럭을 생성하는 것이 가능한 분주기를 구비하는, 클럭 생성 회로.

청구항 2

제1항에 있어서,

상기 분주기는, 상기 소스 클럭의 펄스 열의 일부를 마스크하는 비트 열의 패턴을 사용함으로써, 당해 소스 클럭보다 주파수가 낮은 분주 클럭을 생성하는, 클럭 생성 회로.

청구항 3

제2항에 있어서,

상기 분주기는, 상기 비트 열의 패턴을 전환함으로써, 생성되는 분주 클럭의 주파수를 전환하는, 클럭 생성 회로.

청구항 4

제2항에 있어서,

상기 분주기는, 소정의 하드웨어로부터 제어됨으로써, 상기 비트 열의 패턴을 전환하고, 생성되는 분주 클럭의 주파수를 전환하는, 클럭 생성 회로.

청구항 5

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 분주기는, 상기 소스 클럭의 펄스 열의 일부를 마스크하는 비트 열의 패턴으로서 복수 종의 비트 열의 패턴을 사용하고, 당해 복수 종의 비트 열의 패턴 중 하나를 선택적으로 사용함으로써, 주파수가 상이한 복수 종의 분주 클럭 중 하나를 선택적으로 생성하는, 클럭 생성 회로.

청구항 6

제1항 내지 제4항 중 어느 한 항에 있어서,

어느 대상물의 상태 변화에 따라서, 상기 분주기가 생성하는 분주 클럭의 주파수를 변화시키는 하드웨어로서의 컨트롤러를 더 구비하는, 클럭 생성 회로.

청구항 7

제6항에 있어서,

상기 컨트롤러는, 하나 또는 복수의 트랜잭션의 발생의 유무에 따라서, 상기 분주기가 생성하는 분주 클럭의 주파수를 변화시키는, 클럭 생성 회로.

청구항 8

제6항에 있어서,

상기 컨트롤러는, 소정의 장치에 데이터를 보내는 기억 수단의 빈 용량에 따라서, 상기 분주기가 생성하는 분주 클럭의 주파수를 변화시키는, 클럭 생성 회로.

청구항 9

제6항에 있어서,

상기 컨트롤러는, 소정의 장소의 온도에 따라서, 상기 분주기가 생성하는 분주 클럭의 주파수를 변화시키는, 클럭 생성 회로.

청구항 10

제6항에 있어서,

상기 컨트롤러는, 동작 모드의 천이에 따라서, 상기 분주기가 생성하는 분주 클럭의 주파수를 변화시키는, 클럭 생성 회로.

청구항 11

소스 클럭을 분주기에 입력하고,

상기 분주기에 의해, 상기 소스 클럭의 펄스 열의 일부를 마스크하는 처리를 행함으로써 당해 소스 클럭보다 주파수가 낮은 분주 클럭을 생성하는

것을 포함하는, 클럭 생성 방법.

청구항 12

제11항에 있어서,

상기 분주기에 의해, 상기 소스 클럭의 펄스 열의 일부를 마스크하는 비트 열의 패턴을 사용함으로써, 당해 소스 클럭보다 주파수가 낮은 분주 클럭을 생성하는

것을 포함하는, 클럭 생성 방법.

청구항 13

제12항에 있어서,

상기 분주기에 의해, 상기 비트 열의 패턴을 전환함으로써, 생성되는 분주 클럭의 주파수를 전환하는

것을 포함하는, 클럭 생성 방법.

청구항 14

제11항 내지 제13항 중 어느 한 항에 있어서,

상기 분주기에 의해, 상기 소스 클럭의 펄스 열의 일부를 마스크하는 복수 종의 비트 열의 패턴을 사용하고, 당해 복수 종의 비트 열의 패턴 중 하나를 선택적으로 사용함으로써, 주파수가 상이한 복수 종의 분주 클럭 중 하나를 선택적으로 생성하는 것을 포함하는, 클럭 생성 방법.

발명의 설명

기 술 분 야

[0001] 본 발명의 실시 형태는, 클럭 생성 회로 및 클럭 생성 방법에 관한 것이다.

배 경 기 술

[0002] 컴퓨터나 반도체 등의 회로의 동작 시의 전력을 삭감하기 위해, 클럭을 필요 충분한 주파수로 제어하는 방법은 알려져 있다. 일정 이상의 성능을 필요로 하는 처리나 부하가 높은 처리에 있어서는, 기능을 저하시키지 않도록 클럭 주파수를 높인다. 반대로, 성능을 필요로 하지 않는 처리나 부하가 낮은 처리의 경우에는, 클럭 주파수를 낮추어 동작 시의 전력을 삭감한다.

[0003] 또한, 회로가 허용하는 온도나 최대 전류량 등의 동작 조건이 있는 경우, 그 동작 조건의 범위 내에서 동작시키기 위해, 클럭 주파수를 낮추는 제어가 행해진다. 그러나, 이 경우에는 성능은 희생되어, 원래 제공해야 할 기능을 만족시키기가 어려워진다.

[0004] 일반적으로, 클럭 주파수의 제어에는, 미리 복수의 클럭을 생성해 두고, 그 클럭을 전환하는 방법이나, PLL(Phase Locked Loop) 등의 클럭 발생 회로나 복수의 분주비를 취할 수 있는 분주 회로의 설정 변경 제어를 행하는 방법이 취해진다. 이 경우, 클럭의 전환 시간이나 회로의 면적 오버헤드 등의 문제가 발생한다.

[0005] 클럭 전환에 시간이 걸려 버리면, 최적의 클럭 주파수에 의한 동작을 행할 수 없다. 예를 들어, 저주파수 동작이 가능한 기간에 주파수가 높은 클럭에 의해 동작하여, 불필요한 전력을 소비해 버린다. 또한, 반대로, 높은 주파수에서 동작할 것이 요구되는 기간에 주파수가 낮은 클럭이 공급되어, 성능의 저하를 초래해 버린다.

[0006] 클럭 생성 회로는, 다른 것에 비해 동작률이 높은 블록이며, 면적 오버헤드가 발생한 경우, 본래, 목적으로 하는 전력의 삭감을 달성할 수 없는 요인이 될 수 있다.

발명의 내용

해결하려는 과제

[0007] 실시 형태는, 원하는 주파수의 클럭을 즉시 생성하는 것이 가능한 클럭 생성 회로 및 클럭 생성 방법을 제공한다.

과제의 해결 수단

[0008] 실시 형태의 클럭 생성 회로는, 소스 클럭의 펄스 열의 일부를 마스크하는 처리를 행함으로써 당해 소스 클럭보다 주파수가 낮은 분주 클럭을 생성하는 것이 가능한 분주기를 구비한다.

도면의 간단한 설명

[0009] 도 1은 각 실시 형태에 관한 반도체 장치의 기본 구성의 일례를 나타내는 개념도.

도 2는 펄스 마스크형 분주기의 회로 구성의 일례를 나타내는 회로도.

도 3은 도 2 중에 나타내어진 클럭 펄스 패턴 생성부가 소정의 기억 수단에 기억하고 있는, 복수 종의 클럭 펄스 패턴이 정의된 테이블의 일례를 나타내는 도면.

도 4는 각종 분주 클럭과 소스 클럭의 관계를 나타내는 타이밍 차트.

도 5는 소스 클럭에 대한 마스크 수의 설정 및 인에이블 신호의 온/오프에 따라서 마스크 처리된 분주 클럭이 형성되는 동작의 일례를 나타내는 타이밍 차트.

도 6은 제1 실시 형태에 관한 반도체 장치의 구성의 일례를 나타내는 개념도.

도 7은 각 버스 마스터의 트랜잭션의 상태에 따라서 클럭의 주파수가 변화되는 동작의 일례를 나타내는 타임차트.

도 8은 제2 실시 형태에 있어서 감시의 대상이 되는 FIFO 등을 나타내는 구성의 일례를 나타내는 개념도.

도 9a는 FIFO의 큐잉 상태에 따라서 클럭의 주파수가 변화되는 동작의 일례를 나타내는 타임차트.

도 9b는 FIFO의 큐잉 상태에 따라서 클럭의 주파수가 변화되는 동작의 다른 구체예를 나타내는 타임차트.

도 10은 제3 실시 형태의 제어에 의한 동작의 일례를 나타내는 흐름도.

도 11의 (a) 및 (b)는 제3 실시 형태에 의한 제어와 일반적인 기술에 의한 제어의 차이에 의해 실현되는 온도 변화 및 성능 변화를 대비시켜 나타내는 개념도.

도 12는 제4 실시 형태에 있어서 감시의 대상이 되는 동작 모드의 천이의 일례를 나타내는 개념도.

발명을 실시하기 위한 구체적인 내용

[0010] 이하, 도면을 참조하여 실시 형태에 대해 설명한다.

[0011] (각 실시 형태에 공통)

[0012] 먼저, 각 실시 형태에 공통되는 사항에 대해 설명한다.

[0013] 각 실시 형태에 나타내는 클럭 생성 회로는, 컴퓨터나 반도체를 비롯한 클럭에 의해 동작하는 회로에 적용되는

것이며, 예를 들어 다이내믹 전력 삭감을 위한 기술인 DFS(Dynamic Frequency Scaling) 제어나, 각종 조건하에서의 동작을 실현하는 데 있어서, 각각의 시스템, 사용예에 맞춘 최적의 클럭 펄스를 발생시키는 것을 가능하게 한다. 구체적으로는, 특정 기능을 갖는 분주기(후술하는 펄스 마스크형 분주기)를 사용하여, 소스 클럭의 펄스열의 일부를 마스크하는 처리를 행함으로써, 당해 소스 클럭보다 주파수가 낮은 분주 클럭을 생성하는 것을 가능하게 한다. 당해 마스크 처리는, 최소 1 클럭 사이클 단위(1 펄스 단위)부터 행하는 것이 가능하여, 보다 세밀한 주파수 제어의 실현을 가능하게 한다.

[0014] 도 1은, 각 실시 형태에 관한 반도체 장치의 기본 구성의 일례를 나타내는 개념도이다.

[0015] 도 1에 나타낸 바와 같이, 반도체 장치(100) 중에는, 클럭 컨트롤 모듈(10), 버스 블록(BB), 블록(B0, B1, B2, B3)을 포함하는 복수의 기능 블록 등이 배치되어 있다.

[0016] 클럭 컨트롤 모듈(10)은, 각 부에 공급해야 할 클럭을 생성하는 클럭 생성 회로를 갖는 것이다. 블록(B0, B1, B2, B3)은, 각각, 클럭 컨트롤 모듈(10)로부터 공급되는 클럭을 사용하여 동작하는 기능 블록이다. 버스 블록(BB)은, 클럭 컨트롤 모듈(10)로부터 공급되는 클럭을 사용하여 동작하는 버스를 포함한다.

[0017] 이 반도체 장치(100)에는, 필요에 따라서, 어느 대상물의 어떠한 상태를 검지하는 액티비티 모니터(상태 감시 회로)가 구비된다. 예를 들어, 도 1에 나타낸 바와 같이, 버스 블록(BB)에 액티비티 모니터(AB)가 구비되어 있어도 되고, 블록(B0, B1, B2, B3) 전부, 혹은 그 일부에(예를 들어, 블록(B0)이나 블록(B2)에), 액티비티 모니터(예를 들어, 액티비티 모니터(A0)나 액티비티 모니터(A2))가 구비되어 있어도 된다.

[0018] 액티비티 모니터(AB)는, 예를 들어 버스 블록(BB)에 대한 블록(B0, B1, B2, B3) 각각의 버스 액세스(트랜잭션)의 발생의 유무 등을 감시하도록 구성되어 있어도 된다.

[0019] 액티비티 모니터(A0) 또는 액티비티 모니터(A2)는, 예를 들어 소정의 장치로 데이터를 보내는 FIFO 등의 기억 수단의 빈 용량을 감시하도록 구성되어 있어도 되고, 소정의 장소의 온도를 감시하도록 구성되어 있어도 되고, 소비 전력이 상이한 복수의 동작 모드간의 상태 천이를 감시하도록 구성되어 있어도 된다.

[0020] 클럭 컨트롤 모듈(10)에는, PLL 회로(1), 분주기(2, 3), 클럭 게이팅 회로(4), 레지스터(5), 컨트롤러(제어 회로)(6) 등이 구비되어 있다. 단, 클럭 컨트롤 모듈(10)의 구성 요소는, 이들에 한정되는 것은 아니다. 일부의 구성 요소, 예를 들어 분주기(3)의 설치를 생략하는 등, 적절하게 구성을 변경해도 된다.

[0021] PLL 회로(1)는, 위상이 조정된 클럭 신호를 출력한다. 분주기(2, 3)는, PLL 회로(1)로부터 공급되는 클럭 신호(소스 클럭)에 대해 주파수를 저하시킨 클럭 신호를 출력한다. 클럭 게이팅 회로(4)는, 분주기(2, 3)로부터 공급되는 클럭 신호에 대해 클럭 게이팅을 실시한다. 클럭 게이팅이 실시된 클럭 신호는, 블록(B0, B1, B2, B3)에 각각 공급된다.

[0022] 분주기(2, 3) 중 적어도 분주기(2)는, 소스 클럭의 펄스 열의 일부를 마스크하는 처리를 행함으로써, 당해 소스 클럭보다 주파수가 낮은 분주 클럭을 생성하는 기능을 갖는 펄스 마스크형 분주기이다.

[0023] 구체적으로는, 펄스 마스크형 분주기(2)는, 소스 클럭의 펄스 열의 일부를 마스크하는 비트 열의 패턴(이하, 「클럭 펄스 패턴」이라고 칭함)을 사용함으로써, 당해 소스 클럭보다 주파수가 낮은 분주 클럭을 생성하는 기능을 갖는다. 클럭 펄스 패턴은, 소스 클럭의 펄스 열을 어느 타이밍에 마스크하여, 어느 타이밍에 출력하느냐를 정의하고 있다.

[0024] 또한, 펄스 마스크형 분주기(2)는, 소정의 하드웨어(예를 들어, 컨트롤러(6))로부터 제어됨으로써, 클럭 펄스 패턴을 전환하고, 생성되는 분주 클럭의 주파수를 전환하는 기능을 갖는다. 보다 구체적으로는, 펄스 마스크형 분주기(2)는, 미리 준비되어 있는 복수 종의 클럭 펄스 패턴 중 하나를 선택적으로 사용함으로써, 주파수가 상이한 복수 종의 분주 클럭 중 하나를 선택적으로 생성하는 기능을 갖는다.

[0025] 레지스터(5)는, 컨트롤러(6)가 펄스 마스크형 분주기(2)를 제어하기 위해 사용하는 각종 정보를 기억한다.

[0026] 컨트롤러(6)는, 펄스 마스크형 분주기(2)에 제어용 신호를 보냄으로써, 펄스 마스크형 분주기(2)를 제어한다. 이 컨트롤러(6)는, 예를 들어 반도체 장치(100)에 구비되는 액티비티 모니터(예를 들어, 액티비티 모니터(AB, A0, A2) 등)로부터 얻어지는 소정의 대상물의 상태에 따라서, 펄스 마스크형 분주기(2)가 생성하는 분주 클럭의 주파수를 변화시킬 수 있다.

[0027] 펄스 마스크형 분주기(2)로부터 출력된 클럭은, 필요에 따라서 분주기(3)나 클럭 게이팅 회로(4) 등을 경유한 후, 클럭 컨트롤 모듈(10)로부터 출력되어, 개개의 블록으로 공급된다.

[0028] 도 2는, 펄스 마스크형 분주기(2)의 회로 구성의 일례를 나타내는 회로도이다.

[0029] 펄스 마스크형 분주기(2)에는, 클럭 펄스 패턴 생성부(11), 카운터(12), 클럭 게이팅부(13) 등이 구비되어 있다. 또한, 펄스 마스크형 분주기(2)에는, PLL(1)측으로부터 공급되는 소스 클럭을 입력하는 단자, 컨트롤러(6)로부터 공급되는 제어용 각 신호(후술하는 펄스 마스크 수의 설정을 지시하는 신호나 인에이블 신호 등)를 입력하는 단자, 생성되는 분주 클럭을 출력하는 단자 등이 구비되어 있다.

[0030] 클럭 펄스 패턴 생성부(11)는, 미리 소정의 기억 수단에 기억되어 있는 복수 종의 클럭 펄스 패턴 중 해당되는 클럭 펄스 패턴을, 컨트롤러(6)에 의해 지정되는 펄스 마스크 수에 따라서 선택하고, 선택한 클럭 펄스 패턴의 비트 열을 출력한다. 클럭 펄스 패턴의 비트 열은, 예를 들어 「0」에 의해 소정의 펄스를 마스크하는 것을 나타내고, 「1」에 의해 소정의 펄스를 마스크하지 않는 것을 나타낸다.

[0031] 카운터(12)는, 소스 클럭의 펄스를 카운트함과 함께, 인에이블 신호의 온/오프를 검지한다. 카운터(12)에 의해 얻어진 정보는 클럭 펄스 패턴 생성부(11)에 전달된다. 이에 의해, 클럭 펄스 패턴 생성부(11)는, 인에이블 신호에 따라서 클럭 펄스 패턴의 비트 열 출력을 개시하거나, 소스 클럭에 동기하여 클럭 펄스 패턴의 비트 열을 출력하거나 할 수 있다. 즉, 카운터(12)는, 클럭 펄스 패턴 생성부(11)로부터 하나의 클럭 펄스 패턴의 비트 열을 일정한 사이클로 클럭 게이팅부(13)에 송출시킨다.

[0032] 클럭 게이팅부(13)는, 소스 클럭을 입력함과 함께, 클럭 펄스 패턴 생성부(11)로부터 보내져 온 클럭 펄스 패턴의 비트 열을 입력하고, 클럭 펄스 패턴의 비트 열에 따라서 소스 클럭의 펄스 열의 일부를 마스크하여 출력한다. 즉, 클럭 게이팅부(13)는, 클럭 펄스 패턴의 비트 열에 따라서, 소스 클럭을 펄스마다 마스크하는 처리와 마스크하지 않고 그대로 출력시키는 처리를 실시한다. 이 결과, 클럭 게이팅부(13)로부터는, 소스 클럭에 마스크 처리가 실시된 분주 클럭 혹은 마스크 처리가 실시되지 않은 클럭이 출력된다.

[0033] 도 2에 나타내어진 펄스 마스크형 분주기(2)는, 기존의 클럭 라인에, 소정의 게이팅 기구를 넣기만 하면 실현할 수 있기 때문에, 레이아웃·타이밍 설계나 회로 실장이 용이하다.

[0034] 또한, 도 2에 나타낸 펄스 마스크형 분주기(2)의 구성은, 이것에 한정되는 것은 아니다. 동일한 기능을 실현할 수 있는 것이면, 다른 구성을 채용해도 된다.

[0035] 도 3은, 도 2 중에 나타내어진 클럭 펄스 패턴 생성부(11)가 소정의 기억 수단에 기억하고 있는, 복수 종의 클럭 펄스 패턴이 정의된 테이블의 일례를 나타내는 도면이다.

[0036] 도 3의 예에서는, 소스 클럭에 있어서의 32개의 클럭 사이클(32개의 펄스)을 하나의 단위로 하고 있다. 단, 클럭 펄스 패턴의 단위는 이 예에 한정되는 것은 아니다. 단위가 되는 클럭 사이클(펄스)의 수로서는, 32개가 아닌, 16개나 64개 등, 시스템이나 사용예에 따라서 임의의 수를 채용해도 된다.

[0037] 도 3에는, 펄스 마스크 수에 따른 32가지의 클럭 펄스 패턴의 예가 나타나 있다.

[0038] 각 클럭 펄스 패턴은, 32개의 클럭 사이클 중 어느 펄스를 마스크할지, 또한 어느 펄스를 마스크하지 않고 출력할지를 정의하고 있다. 「0」은, 펄스를 마스크하는 것을 나타내고, 「1」은, 펄스를 마스크하지 않고 출력하는 것을 나타내고 있다.

[0039] 도 3에 나타내어진 테이블은, 제품의 사양이나 제약에 맞추어 원하는 복수의 마스크 패턴을 정의하는 것이 가능하다.

[0040] 도 4는 각종 분주 클럭과 소스 클럭의 관계를 나타내는 타이밍 차트이다.

[0041] 도 4에는, 도 3의 테이블에 나타내어진 복수 종의 클럭 펄스 패턴에 각각 대응하는 클럭의 예가 나타나 있다. 구체적으로는, 펄스 마스크 수가 0인 소스 클럭의 예와, 펄스 마스크 수에 따른 31가지의 분주 클럭의 예가 나타나 있다.

[0042] 도 4의 분주 클럭의 예로 나타내어진 바와 같이, 도 3의 클럭 펄스 패턴의 「0」으로 되어 있는 타이밍에는, 소스 클럭의 펄스가 마스크되고, 도 3의 클럭 펄스 패턴의 「1」로 되어 있는 타이밍에는, 소스 클럭의 펄스가 마스크되지 않고 출력되고 있다.

[0043] 도 4의 예로부터 알 수 있는 바와 같이, 컨트롤러(6)에 의해 지정되는 마스크 수가 증가할수록, 씨닝되는 펄스의 수가 증가하여, 클럭 주파수가 저하되게 된다.

[0044] 도 5는 소스 클럭에 대한 마스크 수의 설정 및 인에이블 신호의 온/오프에 따라서 마스크 처리된 분주 클럭이

형성되는 동작의 일례를 나타내는 타이밍 차트이다.

[0045] 도 5의 예에서는, 마스크 수가 24인 경우를 나타낸다. 마스크 수 24가 설정된 후, 인에이블 신호가 온 상태로 되면, 즉시(예를 들어, 수 사이클 후에), 설정된 마스크 수에 대응하는 클럭 펄스 패턴의 비트 열에 의해, 일정 기간마다, 소스 클럭의 펄스 열이 부분적으로 마스크된다. 이 결과, 소스 클럭과 동일한 주파수였던 분주 클럭은, 주파수가 낮추어진 상태로 된다. 그 후, 인에이블 신호가 오프 상태로 되면, 즉시(예를 들어, 수 사이클 후에), 당해 마스크 처리가 해제되어, 분주 클럭의 주파수가 원래의 상태로 되돌아간다.

[0046] 또한, 분주 클럭을 형성하는 동작은, 도 5의 예에 한정되는 것은 아니다. 예를 들어, 인에이블 신호는 반드시 필요한 것은 아니며, 인에이블 신호를 사용하지 않고 마찬가지의 동작을 실현하는 것도 가능하다. 이 경우, 인에이블 신호가 온 상태로 된 시점을 기준으로 마스크 수의 전환을 행하는 대신, 마스크 수가 설정된 시점을 기준으로 마스크 수가 전환되도록 해도 된다.

[0047] 상술한 펄스 마스크형 분주기(2)는, 도 1의 반도체 장치(100)의 예에 한정되지 않고, 모든 종류의 장치나 시스템에의 내장을 용이하게 행할 수 있는 것이다.

[0048] 또한, 하드웨어로부터 펄스 마스크형 분주기(2)를 제어하는 구성을 채용하고 있기 때문에, 펄스 마스크형 분주기(2)는 새로운 클럭 주파수를 지정하는 신호를 받으면 즉시, 지정된 주파수를 생성할 수 있다. 주파수의 신속한 전환이 가능해지기 때문에, 주파수의 전환 기간에 있어서의 소비 전력의 손실을 저감시킬 수 있다. 또한, 새로운 클럭 주파수를 지정하는 신호를 받은 직후부터, 지정된 주파수를 즉시 생성할 수 있기 때문에, 원하는 성능을 바로 발휘시킬 수 있다.

[0049] 또한, 클럭 게이팅 등의 다이내믹 제어는, 1 사이클에 제어 신호가 유효해질 필요가 있지만, 펄스 마스크형 분주기(2)를 사용한 제어에서는, 필요한 성능에 따른 클럭 펄스 패턴을 임의로 설정할 수 있고, 반드시 1 사이클 제어를 필요로 하지는 않는다.

[0050] 또한, 각각의 시스템, 사용예에 맞추어, 상기 하드웨어에 의한 제어 외에도, 소프트웨어에 의한 제어를 추가하는 것도 가능하다.

[0051] 또한, 클럭 펄스 패턴은 임의로 선택할 수 있고, 각각의 사용예에 따라서 어느 클럭 펄스 패턴을 선택할지를 미리 설정해 둠으로써, 제어를 더 용이하게 할 수 있다.

[0052] (제1 실시 형태)

[0053] 다음으로, 제1 실시 형태에 대해 설명한다.

[0054] 제1 실시 형태에서는, 도 6 및 도 7을 참조하여 설명한다.

[0055] 도 6은, 제1 실시 형태에 관한 반도체 장치의 구성의 일례를 나타내는 개념도이다. 또한, 도 1과 공통되는 요소에는 동일한 부호를 붙이고, 중복되는 설명을 생략한다.

[0056] 도 6에 나타내어진 바와 같이, 버스 블록(BB)에는 액티비티 모니터(AB)가 구비되어 있다. 액티비티 모니터(AB)는, 예를 들어 버스 블록(BB)에 대한 블록(B0, B1, B2, B3)으로부터의 각 트랜잭션의 발생의 유무를 감시한다. 또한, 블록(B0, B1, B2, B3)은 각각 버스 마스터(마스터 모듈)로서 동작하고, 버스 블록(BB)은 상기 버스 마스터와 슬레이브(도시하지 않음) 사이에서 교환되는 신호를 전달하는 버스 본체로서 동작한다. 즉, 액티비티 모니터(AB)는, 버스 마스터와 슬레이브 사이에서 교환되는 신호를 감시하여 각 트랜잭션의 발생 유무를 판정한다. 이하에서는, 블록(B0, B1, B2, B3)을 각각 Master0, Master1, Master2, Master3이라고 칭하는 경우가 있다. 액티비티 모니터(AB)에 의해 감시되는 정보는, 상시 컨트롤러(6)에 전달된다.

[0057] 컨트롤러(6)는, 액티비티 모니터(AB)로부터 전달되는 정보에 나타내어진 각 트랜잭션의 발생 유무에 따라서, 펄스 마스크형 분주기(2)가 마스크하는 펄스의 마스크 수를 변화시키고, 펄스 마스크형 분주기(2)가 생성하는 분주 클럭의 주파수를 변화시킨다. 여기서는, 컨트롤러(6)는, 예를 들어 Master0, Master1, Master2를 감시의 대상으로 하고 있는 것으로 한다.

[0058] 레지스터(5)에는, 어느 블록의 트랜잭션의 상태를 DFS 제어에 사용할지를 지정하는 정보 등이 기억된다. 여기서는, 예를 들어 Master0, Master1, Master2의 각 트랜잭션의 상태를 DFS 제어에 사용할 것을 지정하는 정보가 기억되는 것으로 한다. 이 레지스터(5)에 기억된 정보는, 컨트롤러(6)에 의해 참조된다.

[0059] 도 7은, 각 버스 마스터의 트랜잭션의 상태에 따라서 클럭의 주파수가 변화되는 동작의 일례를 나타내는 타임차

트이다.

[0060] 여기서는, Master0, Master1, Master2의 각 트랜잭션의 상태에 따라서 클럭의 주파수가 변화되는 예를 나타낸다.

[0061] 도 7에 나타내어진 바와 같이, 예를 들어 Master0, Master1, Master2 중 어느 쪽에도 트랜잭션이 발생하지 않은 아이들 상태의 기간(IDLE 기간)에 있어서는, 데이터 전송을 서두를 필요가 없다는 점에서, 통상의 동작보다 저속인 저속 동작이 실시된다. 이에 의해 전력 소비가 억제된다.

[0062] 컨트롤러(6)는, 이러한 동작을 실현하기 위해, 미리 정해진 펄스 마스크 수의 설정을 펄스 마스크형 분주기(2)에 대해 지시하는 신호를 보냄과 함께, 인에이블 신호를 펄스 마스크형 분주기(2)로 보낸다. 이에 의해, 펄스 마스크형 분주기(2)로부터는, 씨닝된 펄스의 수가 많은 클럭이 생성된다.

[0063] 한편, Master0, Master1, Master2 중 적어도 어느 하나에 트랜잭션이 발생하고 있는 비지 상태의 기간(버스 액세스 기간)에 있어서는, 데이터 전송을 서두를 필요가 있다는 점에서, 고속 동작 혹은 통상 동작이 실시된다. 또한, 트랜잭션이 발생한 버스 마스터의 수에 따라서 단계적으로 동작 속도(주파수)가 바뀌도록 해도 된다.

[0064] 컨트롤러(6)는, 이러한 동작을 실현하기 위해, 펄스 마스크 수를 0으로 하는 설정, 혹은 IDLE 기간 시보다 적은 펄스 마스크 수의 설정을 펄스 마스크형 분주기(2)에 대해 지시하는 신호를 보냄과 함께, 인에이블 신호를 펄스 마스크형 분주기(2)에 보낸다. 이에 의해, 펄스 마스크형 분주기(2)로부터는, 펄스가 씨닝되지 않은 클럭, 혹은 씨닝된 펄스의 수가 적은 클럭이 생성된다.

[0065] 이러한 제어에 의해, 소프트웨어에서는 용이하게 실현할 수 없는 미세한 입도인 트랜잭션 단위에서의 DFS 제어를 실현하는 것이 가능해진다. 또한, 트랜잭션이 발생하지 않은 IDLE 기간이 있을 때마다, 저속 동작으로의 전환을 신속하게 행할 수 있기 때문에, 전력 삭감 효과가 크다. 또한, 버스의 스테이터스를 나타내는 신호를 이용하여 제어를 용이하게 실현할 수 있기 때문에, 실장이 용이하다.

[0066] 다음으로, 몇 가지 구체예를 들어 설명한다.

[0067] · 제어예 (1a)

[0068] 아이들(IDLE) 상태의 버스 마스터의 수에 따라서 펄스 마스크 수를 바꾸는 제어예에 대해 설명한다.

[0069] 또한, 이 제어를 실시하기 위해 필요한 정보(각 상태에 대응하는 펄스 마스크 수를 나타내는 정보 등)는, 컨트롤러(6)가 참조할 수 있도록 레지스터(5) 내에 기억된다.

[0070] 여기서는, IDLE 상태의 버스 마스터의 수가 많을수록, 펄스 마스크 수를 많게 하여, 클럭 주파수를 낮추는 것으로 한다.

[0071] Master0, Master1, Master2 모두가 IDLE 상태에 있는 기간에 있어서는, 컨트롤러(6)는, 32개의 펄스 중 예를 들어 24개의 펄스를 마스크하기 위한 제어를 실시한다.

[0072] 또한, Master0과 Master1이 IDLE 상태에 있고, Master2의 트랜잭션이 발생하고 있는 기간에 있어서는, 컨트롤러(6)는 32개의 펄스 중 예를 들어 12개의 펄스를 마스크하기 위한 제어를 실시한다.

[0073] 또한, Master0과 Master1의 트랜잭션이 발생하고 있고, Master2가 IDLE 상태에 있는 기간에 있어서는, 컨트롤러(6)는 32개의 펄스 중의 예를 들어 4개의 펄스를 마스크하기 위한 제어를 실시한다.

[0074] · 제어예 (1b)

[0075] 버스 액세스 중인 각 버스 마스터에 대해, 예를 들어 필요한 성능의 정도 등에 따라서 가중치 부여를 행하고, 그들의 합계에 따라서 펄스 마스크 수를 바꾸는 제어예에 대해 설명한다.

[0076] 또한, 이 제어를 실시하기 위해 필요한 정보(각 버스 마스터의 가중치를 나타내는 정보나, 합계값에 따른 펄스 마스크 수를 나타내는 정보 등)는, 컨트롤러(6)가 참조할 수 있도록 레지스터(5) 내에 기억된다.

[0077] 여기서는, Master0의 가중치를 2, Mater1의 가중치를 1, Master2의 가중치를6으로 하고, 「(버스 액세스 중인 버스 마스터)×(가중치)」의 합계가 많을수록, 펄스 마스크 수를 적게 하고, 클럭 주파수를 높이는 것으로 한다. 또한, (버스 액세스 중인 버스 마스터)의 값은, 예를 들어 버스 마스터마다 1로 한다.

[0078] Master0, Master1, Master2 모두가 IDLE 상태에 있는 기간에 있어서는, 합계가 0이 된다. 이 경우, 컨트롤러(6)는 32개의 펄스 중, 예를 들어 24개의 펄스를 마스크하기 위한 제어를 실시한다.

[0079] 또한, Master0과 Master1이 IDLE 상태에 있고, Master2의 트랜잭션이 발생하고 있는 기간에 있어서는, 합계가 6이 된다. 이 경우, 컨트롤러(6)는, 32개의 펄스 중, 4개의 펄스를 마스크하기 위한 제어를 실시한다.

[0080] 또한, Master0과 Master1의 트랜잭션이 발생하고 있고, Master2가 IDLE 상태에 있는 기간에 있어서는, 합계가 3이 된다. 이 경우, 컨트롤러(6)는, 32개의 펄스 중, 12개의 펄스를 마스크하기 위한 제어를 실시한다.

[0081] · 제어예 (1c)

[0082] 상술한 제어예(1b)에 있어서, 특정한 버스 마스터의 트랜잭션, 예를 들어 Master2의 트랜잭션을 무시하는 제어예에 대해 설명한다.

[0083] 또한, 이 제어를 실시하기 위해 필요한 정보(Master2의 트랜잭션을 무시하는 것을 나타내는 정보 등)는, 컨트롤러(6)가 참조할 수 있도록 레지스터(5) 내에 기억된다.

[0084] Master0, Master1, Master2 모두가 IDLE 상태에 있는 기간에 있어서는, 합계가 0이 된다. 이 경우, 컨트롤러(6)는 32개의 펄스 중, 예를 들어 24개의 펄스를 마스크하기 위한 제어를 실시한다.

[0085] Master0과 Master1이 IDLE 상태에 있고, Master2의 트랜잭션이 발생하고 있는 기간에 있어서는, 합계가 0이 된다. 이 경우, 컨트롤러(6)는, 32개의 펄스 중, 24개의 펄스를 마스크하기 위한 제어를 실시한다.

[0086] 이 제1 실시 형태에 의하면, 트랜잭션의 발생 상황에 따라서, 하드웨어에 의해 펄스 마스크형 분주기(2)를 제어하는 구성에 의해, 버스 트랜잭션 단위로의 주파수의 전환을 행할 수 있으므로, 보다 입도가 미세한 제어를 실현할 수 있고, 전력 절약 효과를 높일 수도 있다.

[0087] 또한, 버스 트랜잭션의 감시는, 기존의 액티비티 모니터 회로에 있어서의 기존의 신호를 이용하여 행할 수 있으므로, 범용성이 높고, 복잡한 회로를 다시 실장할 필요가 없이, 간단한 회로 실장이면 되고, 또한 기존의 회로에의 내장이 용이하다. 또한, 제품마다의 아키텍처 통일이 용이하다.

[0088] 또한, 소프트웨어를 통하지 않고 하드웨어에 의해 자동 제어를 행하기 때문에, 소프트웨어 개발을 위한 공정 수를 삭감할 수 있다. 또한, 소프트웨어에 의한 제어에 비해, 보다 미세한 입도로 제어를 행할 수 있으므로, 저소비 전력 동작이 가능해진다.

[0089] 또한, 레이아웃 공정에 있어서의 타이밍 설계가 용이하고, 면적 증대나 성능 저하 등의 리스크가 낮다.

[0090] 또한, 역치나 가중치, 전환하는 주파수가 가변이기 때문에, 다종의 제품에 적용하는 것이 용이하다.

[0091] (제2 실시 형태)

[0092] 다음으로, 제2 실시 형태에 대해 설명한다.

[0093] 제2 실시 형태에서는, 전술한 도 1을 참조함과 함께, 도 8, 도 9a 및 도 9b를 참조하여 설명한다.

[0094] 제2 실시 형태에 있어서 감시의 대상이 되는 것은, DRAM 등으로부터 디스플레이 등으로 데이터가 공급되는 경로의 도중에 있는 FIFO 등의 기억 수단(이하 「FIFO」라고 약칭함)이다.

[0095] 도 8은, 제2 실시 형태에 있어서 감시의 대상이 되는 FIFO 등을 나타내는 구성의 일례를 나타내는 개념도이다.

[0096] 또한, 도 8에 나타낸 구성은, 반도체 장치(100) 내에 설치되어 있어도 되고, 그 일부가 반도체 장치(100) 외부에 설치되어 있어도 된다.

[0097] 도 8에 나타내어진 바와 같이, DRAM 등으로부터 디스플레이로 데이터가 공급되는 경로의 도중에는, 디스플레이 인터페이스 컨트롤러(21), FIFO(22), 디스플레이 인터페이스(23)가 있다. FIFO(22)는, 디스플레이 인터페이스 컨트롤러(21)로부터 공급되는 데이터를 디스플레이 인터페이스(23)로 순차 송출한다.

[0098] FIFO(22)는, 상황에 따라 큐잉에 빈 공간이 많을 때도 있는가 하면 빈 공간이 적을 때도 있다. 이러한 FIFO(22)의 큐잉 상태를 나타내는 정보는, 예를 들어 반도체 장치(100) 내의 소정의 블록의 액티비티 모니터를 통해 클럭 컨트롤 모듈(10)의 컨트롤러(6)로 전달된다. 단, 정보를 전송하는 경로는, 이 예에 한정되는 것은 아니다. 예를 들어, 다른 수단을 통해, FIFO(22)의 큐잉 상태를 나타내는 정보가 컨트롤러(6)로 전달되도록 해도 된다.

[0099] 여기서는, 예를 들어 FIFO(22)의 큐잉의 빈 용량이, 반도체 장치(100) 내의 블록(B0)의 액티비티 모니터(A0)에 의해 감시되고, 그 감시 결과가 컨트롤러(6)로 전달되는 것으로 한다.

[0100] 컨트롤러(6)는, 액티비티 모니터(A0)로부터 전달되는 정보에 나타나는 FIFO(22)의 큐잉의 빈 용량에 따라서, 펄스 마스크형 분주기(2)가 마스크하는 펄스의 마스크 수를 변화시켜, 펄스 마스크형 분주기(2)가 생성하는 분주 클럭의 주파수를 변화시킨다.

[0101] 레지스터(5)에는, 저속 동작으로 이행하는 타이밍이나 고속 동작(통상 동작)으로 이행하는 타이밍을 정하는 역치 등의 정보가 기억된다. 이 레지스터(5)에 기억된 정보는, 컨트롤러(6)에 의해 참조된다.

[0102] 도 9a는, FIFO(22)의 큐잉 상태에 따라서 클럭의 주파수가 변화되는 동작의 일례를 나타내는 타임차트이다.

[0103] 여기서는, FIFO(22)의 큐잉의 빈 용량에 따라서 클럭의 주파수가 변화되는 예를 나타낸다.

[0104] 도 9a에 나타내어진 바와 같이, 예를 들어 FIFO(22)의 큐잉의 빈 용량이 제1 역치(UPPER)를 상회하는 경우, 큐잉의 빈 공간에 여유가 있어, 데이터 전송을 서두를 필요가 없다는 점에서, 통상의 동작보다 저속인 저속 동작이 개시된다. 이에 의해 전력 소비가 억제된다.

[0105] 컨트롤러(6)는, 이러한 동작을 실현하기 위해, FIFO(22)의 큐잉의 빈 용량이 제1 역치(UPPER)를 상회하는 것을 검지하면, 미리 정해진 펄스 마스크 수의 설정을 펄스 마스크형 분주기(2)에 대해 지시하는 신호를 보냄과 함께, 인에이블 신호를 펄스 마스크형 분주기(2)로 보낸다. 이에 의해, 펄스 마스크형 분주기(2)로부터는, 씨닝된 펄스의 수가 많은 클럭이 생성된다. 또한, 이 제어를 실시하기 위해 필요한 정보(제1 역치(UPPER)를 나타내는 정보 등)는, 컨트롤러(6)가 참조할 수 있도록 레지스터(5) 내에 기억된다.

[0106] 한편, FIFO(22)의 큐잉의 빈 용량이 제2 역치(LOWER)를 하회하는 경우, 큐잉의 빈 공간에 여유가 없어, 데이터 전송을 서두를 필요가 있다는 점에서, 고속 동작 또는 통상 동작이 개시된다. 또한, 큐잉의 빈 용량에 따라서 단계적으로 동작 속도(주파수)가 바뀌도록 해도 된다.

[0107] 컨트롤러(6)는, 이러한 동작을 실현하기 위해, FIFO(22)의 큐잉의 빈 용량이 제2 역치(LOWER)를 하회하는 것을 검지하면, 펄스 마스크 수를 0으로 하는 설정 혹은 IDLE 기간 시보다 적은 펄스 마스크 수의 설정을 펄스 마스크형 분주기(2)에 대해 지시하는 신호를 보냄과 함께, 인에이블 신호를 펄스 마스크형 분주기(2)로 보낸다. 이에 의해, 펄스 마스크형 분주기(2)로부터는, 펄스가 씨닝되지 않은 클럭 또는 씨닝된 펄스의 수가 적은 클럭이 생성된다. 또한, 이 제어를 실시하기 위해 필요한 정보(제2 역치(LOWER)를 나타내는 정보 등)는 컨트롤러(6)가 참조할 수 있도록 레지스터(5) 내에 기억된다.

[0108] 다음으로, 그 밖의 구체예를 들어 설명한다.

[0109] · 제어예 (2a)

[0110] 상기 2개의 역치를 사용하여 펄스 마스크 수를 3단계로 변화시키는 제어예에 대해, 도 9b를 참조하여 설명한다.

[0111] 또한, 이 제어를 실시하기 위해 필요한 정보(2개의 역치를 나타내는 정보 등)는 컨트롤러(6)가 참조할 수 있도록 레지스터(5) 내에 기억된다.

[0112] 여기서는, FIFO(22)의 큐잉의 빈 공간에 여유가 있을수록, 펄스 마스크 수를 많게 하고, 클럭 주파수를 낮추는 것으로 한다.

[0113] FIFO(22)의 큐잉의 빈 용량이 제1 역치(UPPER)를 상회하는 기간에 있어서는, 컨트롤러(6)는 32개의 펄스 중 예를 들어 28개의 펄스를 마스크하기 위한 제어를 실시한다.

[0114] 또한, FIFO(22)의 큐잉의 빈 용량이 제1 역치(UPPER)와 제2 역치(LOWER) 사이에 있는 기간에 있어서는, 컨트롤러(6)는 32개의 펄스 중 예를 들어 14개의 펄스를 마스크하기 위한 제어를 실시한다.

[0115] 또한, FIFO(22)의 큐잉의 빈 용량이 제2 역치(LOWER)를 하회하는 기간에 있어서는, 컨트롤러(6)는 32개의 펄스 중 예를 들어 0개의 펄스를 마스크하기(펄스 마스크를 무효로 하기) 위한 제어를 실시한다.

[0116] 이에 의해, 보다 세밀한 제어를 실현할 수 있다. 또한, 역치를 2개가 아닌, 3개 이상으로 함으로써 보다 다단계로 펄스 마스크 수를 변화시키도록 제어해도 된다.

[0117] 이 제2 실시 형태에 의하면, FIFO(22)의 큐잉 상태에 따라서 컨트롤러(6)가 펄스 마스크형 분주기(2)를 제어하는 구성에 있어서도, 제1 실시 형태의 경우와 마찬가지의 효과를 얻을 수 있다.

[0118] (제3 실시 형태)

[0119] 다음으로, 제3 실시 형태에 대해 설명한다.

[0120] 제3 실시 형태에서는, 전술한 도 1을 참조함과 함께, 도 10 및 도 11을 참조하여 설명한다.

[0121] 제3 실시 형태에 있어서 감시의 대상이 되는 것은, 반도체 장치(100)의 소정의 장소의 온도이다.

[0122] 도 1에 나타내어진 반도체 장치(100)의 소정의 장소에는, 온도 센서(도시하지 않음)가 설치된다. 당해 온도 센서는, 반도체 장치(100)의 소정의 장소의 온도를 검지하는 것이다. 또한, 이 온도 센서는, 반도체 장치(100)의 내측에 설치되어 있어도 되고, 반도체 장치(100)의 외측에 설치되어 있어도 된다. 온도 센서에 의해 검지된 온도를 나타내는 정보는, 직접, 컨트롤러(6)에 전달되도록 해도 되고, 혹은 다른 수단을 통해 컨트롤러(6)에 전달되도록 해도 된다. 예를 들어, 온도 센서에 의해 검지되는 온도가 반도체 장치(100) 내의 블록(B0)의 액티비티 모니터(A0)에 의해 감시되고, 그 감시 결과가 컨트롤러(6)에 전달되도록 해도 된다.

[0123] 여기서는, 온도 센서에 의해 검지된 온도를 나타내는 정보가, 직접 컨트롤러(6)에 전달되는 것으로 한다.

[0124] 컨트롤러(6)는, 온도 센서로부터 전달되는 정보에 나타나는 온도의 값에 따라서, 펄스 마스크형 분주기(2)가 마스크하는 펄스의 마스크 수를 변화시키고, 펄스 마스크형 분주기(2)가 생성하는 분주 클럭의 주파수를 변화시킨다.

[0125] 레지스터(5)에는, 저속 동작으로 이행하는 타이밍이나 고속 동작(통상 동작)으로 이행하는 타이밍을 정하는 온도의 역치 등의 정보가 기억된다. 이 레지스터(5)에 기억된 정보는, 컨트롤러(6)에 의해 참조된다.

[0126] 도 10은, 제3 실시 형태의 제어에 의한 동작의 일례를 나타내는 흐름도이다.

[0127] 여기서는, 단위 시간당 온도 변화량(전회 취득한 온도와 금회 취득한 온도의 차분)에 따라서, 펄스 마스크 수를 바꾸는 제어예에 대해 설명한다.

[0128] 먼저, 컨트롤러(6)는 펄스 마스크형 분주기(2)에 대한 인에이블 신호를 온으로 한 상태에서, 온도 센서로부터의 온도 측정의 완료를 나타내는 EOC(End Of Conversation) 신호가 어서트될 때까지 대기한다(스텝 S1).

[0129] 다음으로, 컨트롤러(6)는, 온도 측정의 완료를 나타내는 EOC 신호의 어서트에 따라서, 온도 정보를 취득하고, 취득한 온도에 기초하는 처리를 개시한다(스텝 S2).

[0130] 컨트롤러(6)는, 금회 취득한 온도와, 레지스터(5)에 미리 설정된 기준 온도를 비교하여, 금회 취득한 온도가 기준 온도 이상인지 여부를 판정한다(스텝 S3).

[0131] 금회 취득한 온도가 기준 온도 이상인 경우에는(스텝 S3의 "예"), 스텝 S4로 진행한다. 한편, 금회 취득한 온도가 기준 온도 미만인 경우에는(스텝 S3의 "아니오"), 펄스 마스크 수를 0으로 되돌리고(스텝 S5), 스텝 S6으로 진행한다.

[0132] 스텝 S4에서는, 컨트롤러(6)는, 전회 취득한 온도와 금회 취득한 온도의 비교를 행한다(스텝 S4).

[0133] 보다 구체적으로는, 컨트롤러(6)는, 금회 취득한 온도와 전회 취득한 온도의 차분과, 레지스터(5)에 미리 설정된 제1 역치(RANGE_UP)의 비교, 및 금회 취득한 온도와 전회 취득한 온도의 차분과, 레지스터(5)에 미리 설정된 제2 역치(RANGE_DN)의 비교를 행한다.

[0134] 금회 취득한 온도와 전회 취득한 온도의 차가, 레지스터(5)에 미리 설정된 제1 역치(RANGE_UP) 이상인 경우에는, 단위 시간 내에 일정 이상의 온도 상승이 있었다고 간주하고, 레지스터(5)에 미리 설정된 펄스 마스크 수 증가량(RAIO_UP)만큼, 현재의 펄스 마스크 수를 증가시키고(스텝 S5A), 클럭 주파수를 낮춘다. 이 후, 스텝 S6으로 진행한다.

[0135] 한편, 금회 취득한 온도와 전회 취득한 온도의 차가, 레지스터(5)에 미리 설정된 제2 역치(RANGE_DN) 이하인 경우에는, 단위 시간 내에 일정 이상의 온도 하강이 있었다고 간주하고, 레지스터(5)에 미리 설정된 펄스 마스크 수 증가량(RAIO_DN)만큼, 현재의 펄스 마스크 수를 감소시키고(스텝 S5B), 클럭 주파수를 높인다. 이 후, 스텝 S6으로 진행한다.

[0136] 또한, 금회 취득한 온도와 전회 취득한 온도의 차가, 제1 역치(RANGE_UP)와 제2 역치(RANGE_DN) 사이에 있는 경우에는, 펄스 마스크 수는 바꾸는 일 없이, 스텝 S6으로 진행한다.

[0137] 마지막으로, 컨트롤러(6)는, 인에이블 신호의 상태를 확인한다(스텝 S6). 컨트롤러(6)는, 인에이블 신호를 온으로 하고 있는 한은, 스텝 S1로부터의 처리를 반복한다. 한편, 인에이블 신호를 오프로 하였을 때에는, 당해

제어를 종료시킨다.

[0138] 또한, 상기 동작에 있어서는, 예를 들어 x℃라면 a개의 펄스 마스크를 지정하고, y℃라면 b개의 펄스 마스크를 지정하는 등을 정의한 테이블을 레지스터(5)에 미리 설정해 두고, 이 테이블의 정의에 따라서 제어를 행하도록 변형 실시하는 것도 가능하다.

[0139] 도 11은, 제3 실시 형태에 의한 제어와 일반적인 기술에 의한 제어에 의해 실현되는 온도 변화 및 성능 변화의 차이를 대비시켜 나타내는 개념도이다.

[0140] 도 11의 (a)는, 일반적인 기술에 의한 제어에 의해 실현되는 온도 변화 및 성능 변화의 예를 나타내고 있다. 한편, 도 11의 (b)는, 제3 실시 형태에 의한 제어에 의해 실현되는 온도 변화 및 성능 변화의 예를 나타내고 있다.

[0141] 또한, 도 11의 (a), (b) 중, 부호 T1은, 일반적인 기술에 의한 제어에 의해 실현되는 온도 변화를 나타내고, 부호 T2는, 본 실시 형태에 의한 제어에 의해 실현되는 온도 변화를 나타낸다. 또한, 부호 P1은, 일반적인 기술에 의한 제어에 의해 실현되는 성능 변화를 나타내고, 부호 P2는, 본 실시 형태에 의한 제어에 의해 실현되는 성능 변화를 나타낸다.

[0142] 도 11의 (a)에 나타낸 일반적인 기술에 의한 제어에서는, 온도가 상승하는 경우, 일반적인 분주기를 사용하여 클럭을 분주시킴으로써 클럭 주파수를 단계적으로 크게 하강시킨다. 이것에 따라서, 장치의 성능도 단계적으로 크게 저하된다.

[0143] 일반적인 분주기에서는, 온도의 상승에 따라서 세밀한 주파수 제어를 행할 수 없어, 대략적인 주파수 제어를 행하게 된다. 그 때문에, 온도 변화 T1에 나타낸 바와 같이 온도의 상승을 억제하는 효과가 작은 데다가 성능 변화 P1로 나타내어진 바와 같이 성능의 저하가 커진다.

[0144] 이에 비해, 도 11의 (b)에 나타낸 제3 실시 형태에 의한 제어에서는, 온도가 상승하는 경우, 예를 들어 단위 시간당 온도 변화량이 일정량을 초과할 때마다, 펄스 마스크형 분주기(2)의 펄스 마스크 수를 세밀하게 변화시킴으로써, 클럭 주파수를 조금씩 하강시킨다. 이것에 따라서, 장치의 성능도 조금씩 저하된다.

[0145] 이 제3 실시 형태에서는, 온도의 상승에 따라서, 펄스 마스크형 분주기(2)에 보다 세밀한 주파수 제어를 행할 수 있다. 그 때문에, 온도 변화 T2로 나타낸 바와 같이 온도의 상승을 효과적으로 억제할 수 있는 데다가 성능 변화 P2로 나타낸 바와 같이 성능의 저하를 억제할 수 있다.

[0146] (제4 실시 형태)

[0147] 다음으로, 제4 실시 형태에 대해 설명한다.

[0148] 제4 실시 형태에서는, 전술한 도 1을 참조함과 함께, 도 12를 참조하여 설명한다.

[0149] 제4 실시 형태에 있어서 감시의 대상이 되는 것은, 도 1에 나타내어진 반도체 장치(100)의 동작 모드이다.

[0150] 반도체 장치(100)의 동작 모드는, 예를 들어 반도체 장치(100) 내의 블록(B2)의 액티비티 모니터(A2)에 의해 감시되고, 그 감시 결과가 컨트롤러(6)로 전달되는 것으로 한다.

[0151] 컨트롤러(6)는, 액티비티 모니터(A2)로부터 전달되는 정보에 나타내어진 동작 모드의 천이에 따라서, 펄스 마스크형 분주기(2)가 마스크하는 펄스의 마스크 수를 변화시키고, 펄스 마스크형 분주기(2)가 생성하는 분주 클럭의 주파수를 변화시킨다.

[0152] 레지스터(5)에는, 개개의 동작 모드에 대응하는 펄스 마스크 수를 각각 정의한 테이블 등의 정보가 기억된다. 이 레지스터(5)에 기억된 정보는, 컨트롤러(6)에 의해 참조된다.

[0153] 도 12는, 제4 실시 형태에 있어서 감시의 대상이 되는 동작 모드의 천이의 일례를 나타내는 개념도이다.

[0154] 도 12에 나타낸 바와 같이, 반도체 장치(100)의 동작 모드에는, 예를 들어 스타트 업 모드(M1), 액티브 모드(M2)가 있는 것 외에, 웨이트 모드(M11), 리텐션 모드(M12) 및 RTC 모드(M13)나, 슬립 모드(M21)가 있다.

[0155] 스타트 업 모드(M1)는, 액티브 모드(M2)로 천이하는 것이 가능하다.

[0156] 액티브 모드(M2)는, 웨이트 모드(M11), 리텐션 모드(M12), RCT 모드(M13) 및 슬립 모드(M21) 중 어느 쪽으로도 천이하는 것이 가능하다.

[0157] 웨이트 모드(M11)는, 액티브 모드(M2)로 천이하는 것이 가능하다. 리텐션 모드(M12)는, 액티브 모드(M2)로 천이하는 것이 가능하다. RCT 모드(M13)는, 스타트 업 모드(M1)로 천이하는 것이 가능하다.

[0158] 슬립 모드(M21)는, 액티브 모드(M2)로 천이하는 것이 가능하다.

[0159] 웨이트 모드(M11), 리텐션 모드(M12), RTC 모드(M13)에 있어서는, CPU는 딥 슬립 상태로 된다. 이때, CPU의 클럭은 정지되어, 고속 복귀는 불필요하다. 그 때문에, 클럭은 공급원으로부터 정지시켜 두는 것이 가능하다(PLL 회로(1)도 정지시켜 두는 것이 가능함). 이 경우, 도 1 중의 컨트롤러(6)는, 펄스 마스크형 분주기(2)의 펄스 마스크 수의 설정 등의 제어는 불필요해진다.

[0160] 한편, 슬립 모드(M21)에 있어서는, CPU는 슬립 상태로 된다. 이때, CPU의 클럭은 정지되지만, 고속 복귀는 필요하다. 그 때문에, 복귀 시간을 고려하여 클럭의 공급원은 동작시켜 둘 필요가 있다(PLL 회로(1)도 동작시켜 둘 필요가 있음). 이 경우, 동작률을 100%로 할 필요는 없다. 따라서, 도 1 중의 컨트롤러(6)는 소비 전력을 삭감하기 위해, 펄스 마스크형 분주기(2)의 펄스 마스크 수를, 예를 들어 액티브 모드(M2) 시의 펄스 마스크 수보다 많은 값으로 설정함으로써, 클럭 주파수를 저하시킨다.

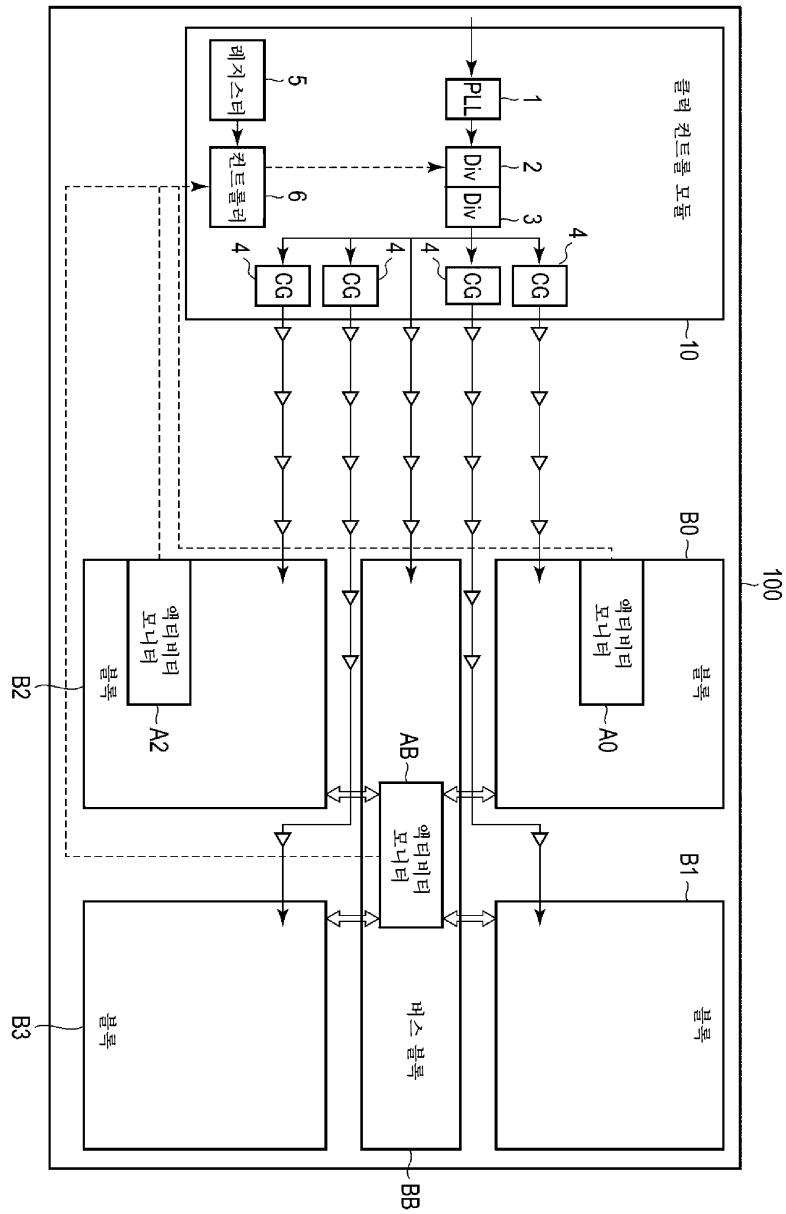
[0161] 이 제4 실시 형태에서는, CPU 슬립 모드 등의 동작률을 100%로 할 필요가 없는 모드에 있어서, 펄스 마스크형 분주기(2)에 의해 클럭 주파수를 원하는 값으로 저하시킴으로써, 소비 전력을 한층 더 삭감할 수 있다.

[0162] 이상에서 상세하게 설명한 바와 같이 실시 형태에 따르면, 원하는 주파수의 클럭을 즉시 생성하는 것이 가능해진다.

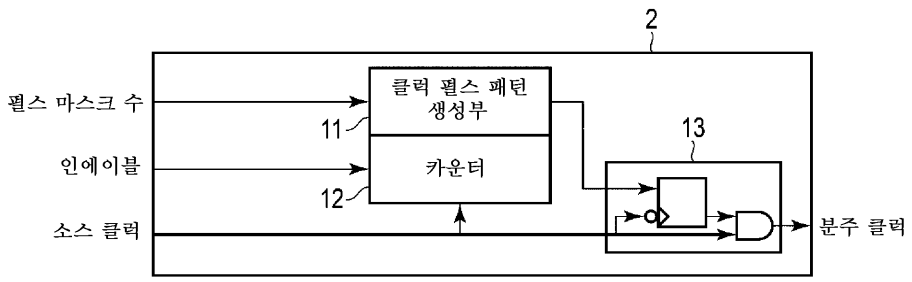
[0163] 본 발명의 몇 가지의 실시 형태를 설명하였지만, 이들 실시 형태는, 예로서 제시한 것이며, 발명의 범위를 한정하는 것은 의도하고 있지 않다. 이들 신규의 실시 형태는, 그 밖의 다양한 형태로 실시되는 것이 가능하고, 발명의 요지를 일탈하지 않는 범위에서, 다양한 생략, 치환, 변경을 행할 수 있다. 이들 실시 형태나 그 변형은, 발명의 범위나 요지에 포함됨과 함께, 청구범위에 기재된 발명과 그 균 등의 범위에 포함된다.

도면

도면1


클럭 컨트롤 모듈
레지스터
5
PLL
1
컨트롤러
6
Div
2
Div
3
CG
4
10
100
B0
B1
B2
B3
AB
BB
액티비티 모니터
A0
A2
블록
버스 블록

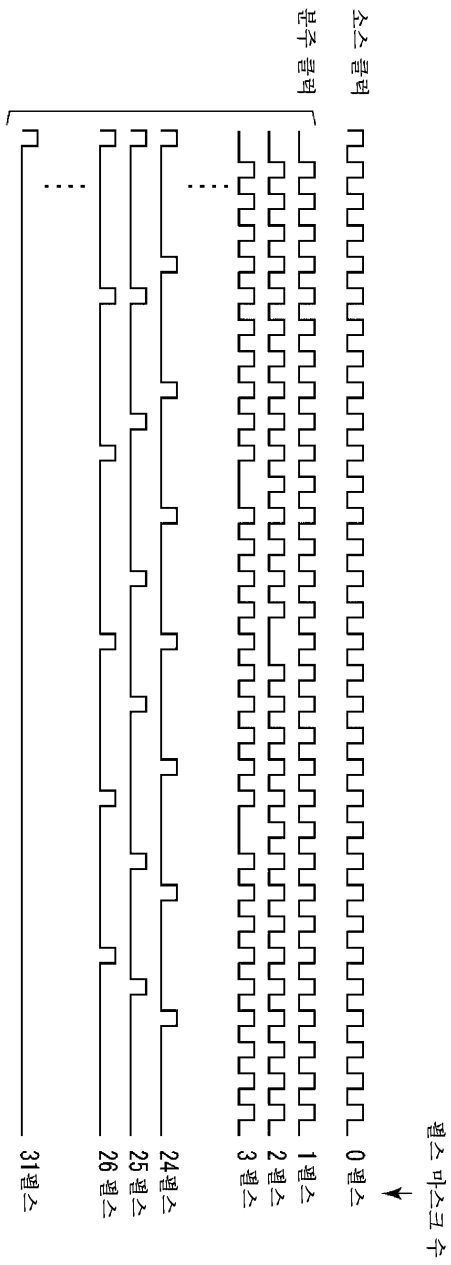
도면2


2
펄스 마스크 수
인에이블
소스 클럭
클럭 펄스 패턴 생성부
11
카운터
12
13
분주 클럭

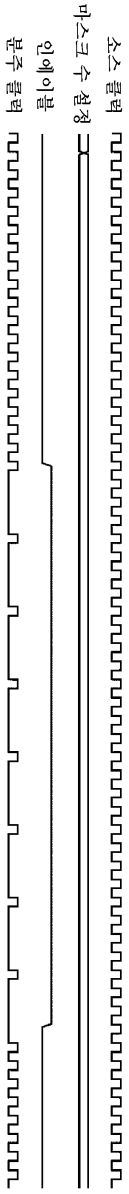
도면3

마스크 수	클럭 펄스 패턴(0: 마스크, 1: 펄스)
0×00	11111111111111111111111111111111
0×01	01111111111111111111111111111111
0×02	01111111111111110111111111111111
0×03	01111111111011111111110111111111
0×04	01111111011111110111111101111111
0×05	01111101111111011111011111011111
0×06	01111011110111111011110111011111
0×07	01111011101111011101111011101111
0×08	01110111011101110111011101110111
0×09	01101110111011011101110110111011
0×0a	01101101101101100110110110110111
0×0b	01101101101101101101101101101101
0×0c	01101101011011010110110101101101
0×0d	01101101011010101101101011010101
0×0e	01101010110101011010101101010101
0×0f	01010101011010101010110101010101
0×10	01010101010101010101010101010101
0×11	10101010100101010101001010101010
0×12	10010101001010100101010010101010
0×13	10010010100101010010010100101010
0×14	10010010100100101001001010010010
0×15	10010010010010010010010010010010
0×16	10010010010010001001001001001000
0×17	10010001000100100010001001000100
0×18	10001000100010001000100010001000
0×19	10000100010000100010000100010000
0×1a	10000100001000001000010000100000
0×1b	10000010000001000001000000100000
0×1c	10000000100000001000000010000000
0×1d	10000000000100000000001000000000
0×1e	10000000000000001000000000000000
0×1f	10000000000000000000000000000000

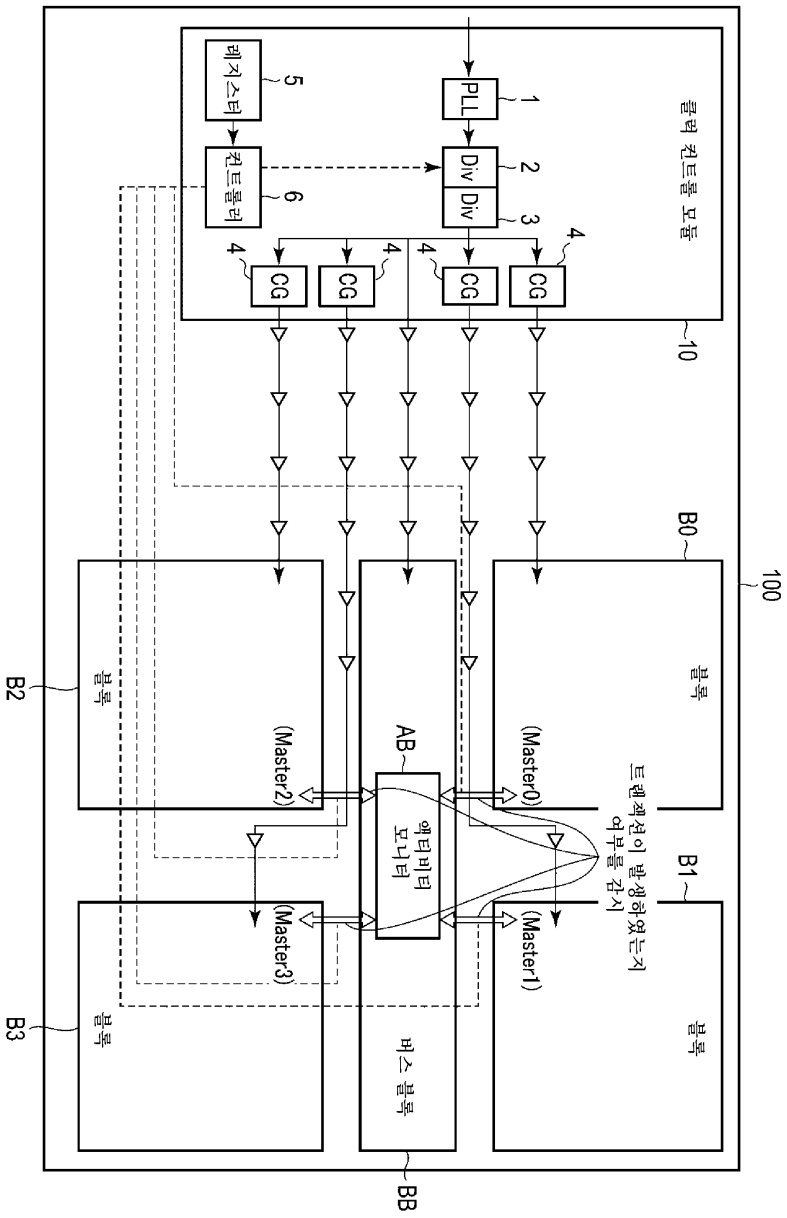
도면4



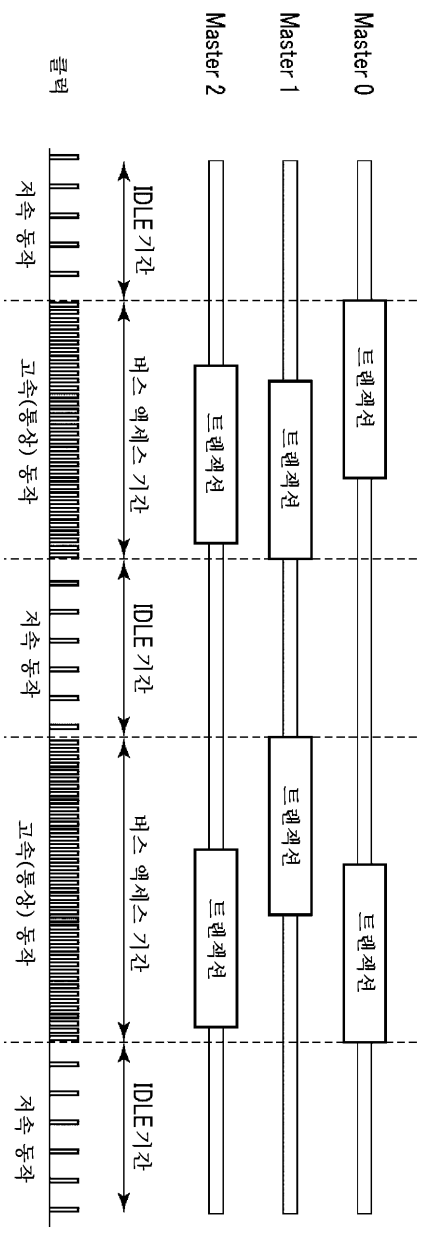
도면5



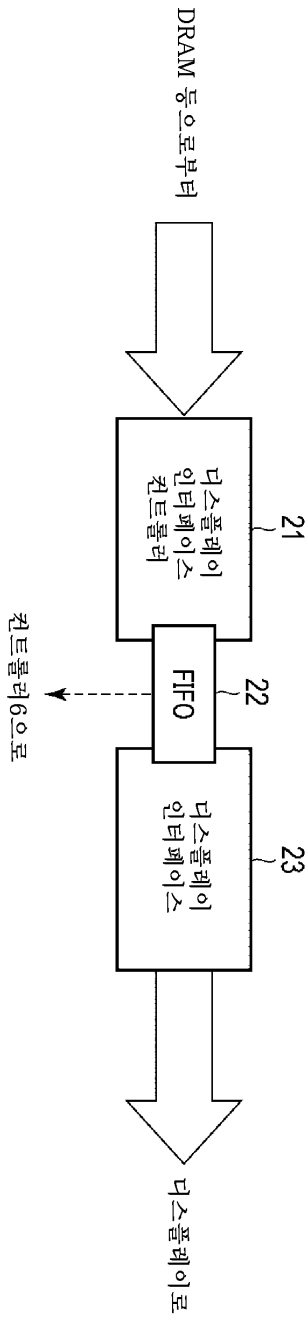
도면6


클럭 컨트롤 모듈
레지스터
컨트롤러
PLL
Div
Div
CG
CG
CG
CG
1
2
3
4
5
6
10
100
B0
B1
B2
B3
BB
AB
블록
(Master0)
(Master1)
(Master2)
(Master3)
애터비터
모니터
버스 블록
트랜잭션이 발생하였는지
여부를 감시

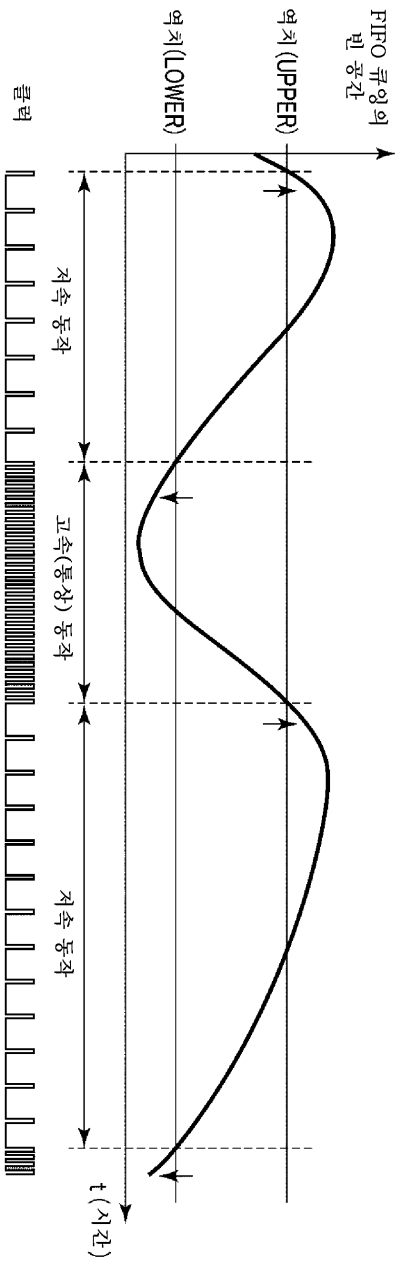
도면7


Master 0
Master 1
Master 2
클럭
트랜잭션
트랜잭션
트랜잭션
트랜잭션
트랜잭션
트랜잭션
IDLE 기간
버스 액세스 기간
IDLE 기간
버스 액세스 기간
IDLE 기간
저속 동작
고속(통상) 동작
저속 동작
고속(통상) 동작
저속 동작

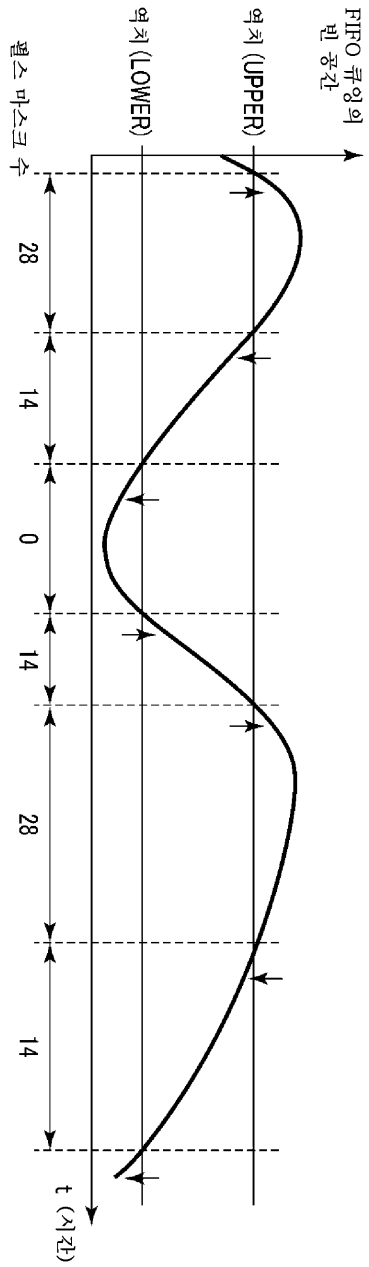
도면8


DRAM 등으로부터
디스플레이 인터페이스 컨트롤러
21
FIFO
22
컨트롤러6으로
디스플레이 인터페이스
23
디스플레이로

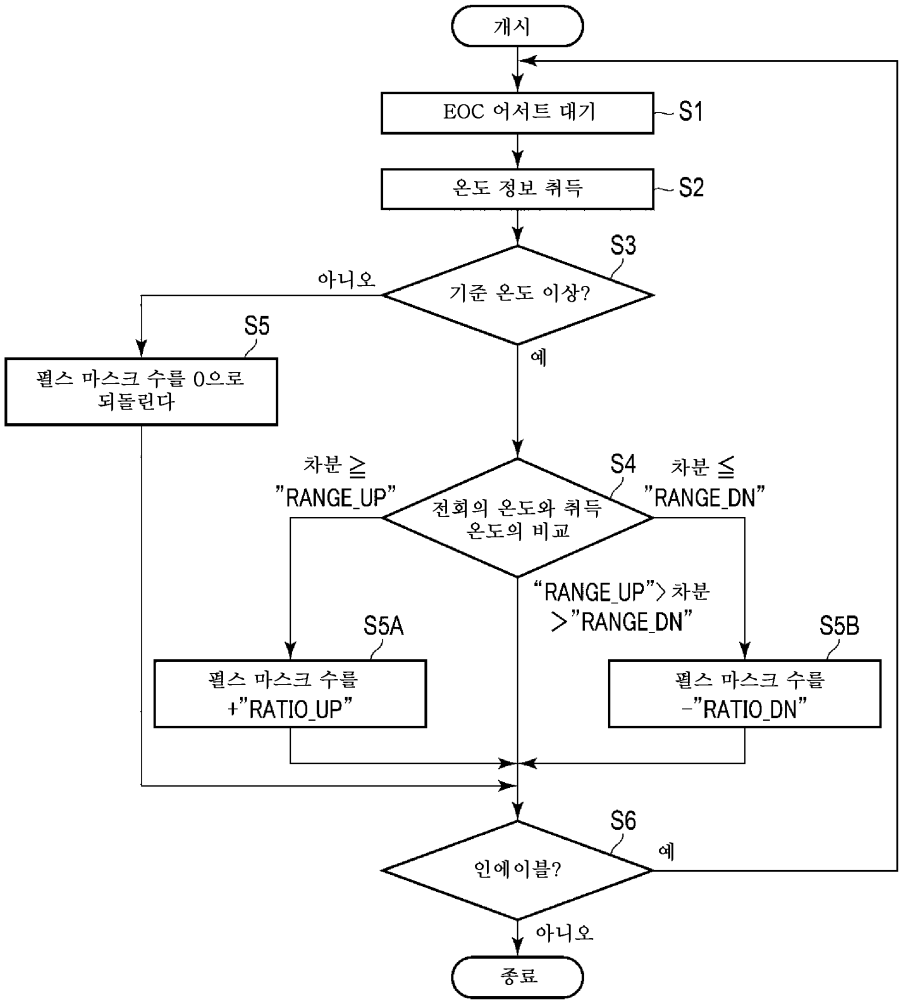
도면9a



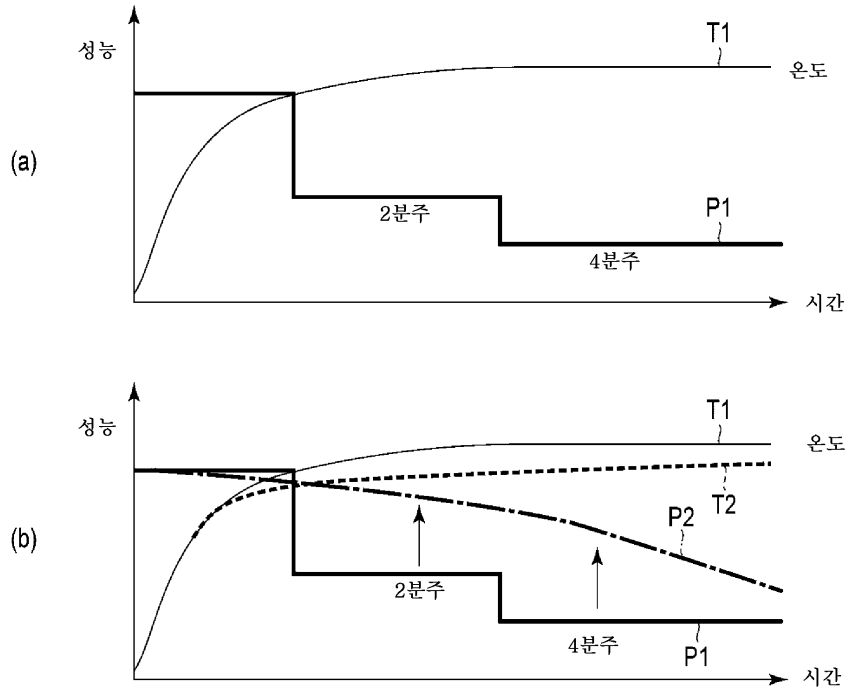
도면9b



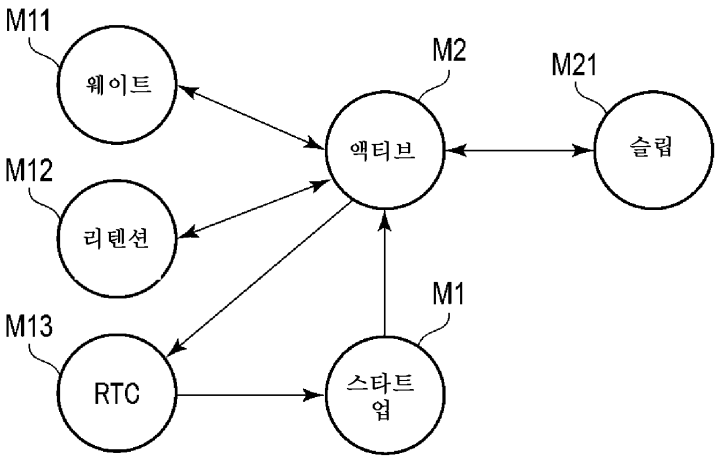
도면10


개시
EOC 어서트 대기
S1
온도 정보 취득
S2
S3
기준 온도 이상?
아니오
예
S5
펄스 마스크 수를 0으로
되돌린다
차분 ≧
"RANGE_UP"
S4
전회의 온도와 취득
온도의 비교
차분 ≦
"RANGE_DN"
"RANGE_UP"> 차분
>"RANGE_DN"
S5A
펄스 마스크 수를
+"RATIO_UP"
S5B
펄스 마스크 수를
-"RATIO_DN"
S6
인에이블?
예
아니오
종료

도면11


(a)
성능
T1
온도
2분주
P1
4분주
시간
(b)
성능
T1
온도
T2
P2
2분주
4분주
P1
시간

도면12


M11
웨이트
M2
M21
액티브
슬립
M12
리텐션
M1
M13
RTC
스타트업