



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0019334

(43) 공개일자 2016년02월19일

(51) 국제특허분류(Int. Cl.)

H03K 5/13 (2014.01) H03K 19/173 (2006.01)

(21) 출원번호 10-2014-0104003

(22) 출원일자 2014년08월11일

심사청구일자 2014년08월11일

(71) 출원인

한림대학교 산학협력단

강원도 춘천시 한림대학길 1, 한림대학교(옥천동)

(72) 발명자

이정근

강원도 춘천시 서부대성로 34번지 한신 휴플러스
아파트 103동 1301호

(74) 대리인

특허법인 천지

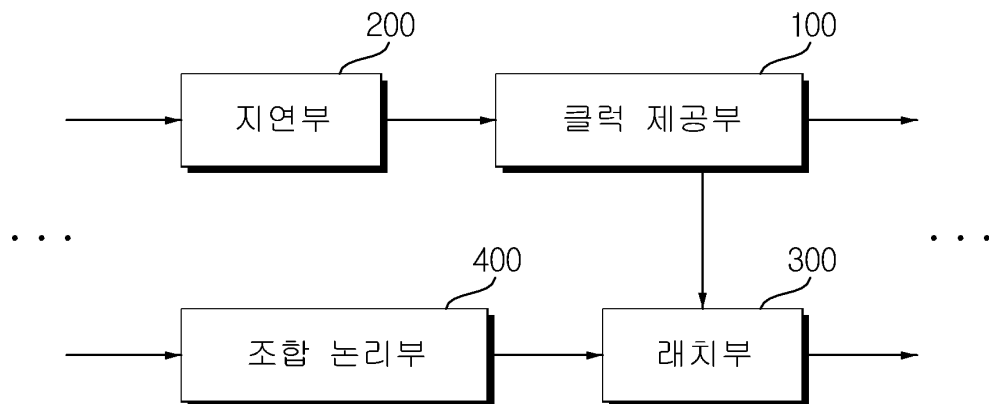
전체 청구항 수 : 총 6 항

(54) 발명의 명칭 비동기 클럭을 가지는 파이프라인 회로 장치

(57) 요약

본 발명은 비동기 클럭을 가지는 파이프라인 회로 장치에 관한 것으로, 이러한 본 발명은 파이프라인 형태의 직렬로 배치되는 복수의 조합논리부와, 각각이 상기 복수의 조합논리부와 일대일로 대응하며, 입력되는 클럭 신호에 따라 상기 복수의 조합논리부 사이에서 이전 스테이지의 조합논리부에서 출력된 데이터를 다음 스테이지의 조합논리부로 전달하는 복수의 래치부와, 상기 복수의 래치부 각각과 일대일로 대응하여 상기 복수의 래치부 각각에 비동기 방식으로 상기 클럭 신호를 제공하는 복수의 클럭제공부를 포함한다. 이러한 본 발명에 따르면, 본 발명의 비동기 방식의 파이프라인 구조의 회로는 대응하는 동기 방식의 파이프라인 구조의 회로와 비교할 때, 전력 소비를 줄일 수 있으며, 본 발명의 비동기 방식의 파이프라인 구조의 회로는 대응하는 동기 방식의 파이프라인 구조의 회로와 비교할 때, 버스트 데이터를 처리하는 경우, 추가의 전력 감소를 이룰 수 있다.

대표도 - 도1a



명세서

청구범위

청구항 1

파이프라인 형태의 직렬로 배치되는 복수의 조합논리부;

각각이 상기 복수의 조합논리부와 일대일로 대응하며, 입력되는 클럭 신호에 따라 상기 복수의 조합논리부 사이에서 이전 스테이지의 조합논리부에서 출력된 데이터를 다음 스테이지의 조합논리부로 전달하는 복수의 래치부; 및

상기 복수의 래치부 각각과 일대일로 대응하여 상기 복수의 래치부 각각에 비동기 방식으로 상기 클럭 신호를 제공하는 복수의 클럭제공부;를 포함하는 것을 특징으로 하는 파이프라인 회로 장치.

청구항 2

제1항에 있어서,

상기 클럭제공부는 제1 C소자 및 제2 C소자를 포함하고, 상기 제1 C소자 및 상기 제2 C소자는 각각 입력, 반전 입력, 리셋 및 출력을 가지며, 상기 제1 C소자의 출력은 상기 제2 C소자의 입력이고, 상기 제2 C소자의 출력은 클럭 신호로 대응하는 래치부에 입력되는 동시에 상기 제1 C소자의 반전 입력이 되는 것을 특징으로 하는 파이프라인 회로 장치.

청구항 3

제2항에 있어서,

상기 클럭제공부의 제1 C소자의 하이 레벨 출력이 클럭제공부의 제2 C소자의 입력에 인가되면 상기 클럭제공부의 제2 C소자의 출력이 하이 레벨이 되고, 상기 클럭제공부의 제2 C소자의 하이 레벨 출력이 래치부에 인가되면, 상기 래치부는 상승 에지에서 오픈되어 상기 조합논리부의 데이터를 다음 스테이지로 전달하는 것을 특징으로 하는 파이프라인 회로 장치.

청구항 4

제2항에 있어서,

상기 제1 C소자 및 상기 제2 C소자 각각은

병렬로 연결되는 제1 내지 제3 AND 게이트; 및

직렬로 연결되는 OR 게이트;를 포함하며,

상기 제1 C소자 및 상기 제2 C소자 각각에 대한 입력은 상기 제1 및 제2 AND 게이트에 인가되며, 반전 입력은 제2 및 제3 AND 게이트에 인가되고, 리셋은 상기 제1 내지 제3 AND 게이트에 인가되며, 상기 제1 내지 제3 AND 게이트의 출력은 상기 OR 게이트에 인가되고, OR 게이트의 출력은 제1 및 제3 AND 게이트에 피드백되는 것을 특징으로 하는 파이프라인 회로 장치.

청구항 5

제2항에 있어서,

상기 복수의 클럭제공부와 일대일 대응하여 대응하는 클럭제공부의 출력을 지연하는 복수의 지연부를 더 포함하는 것을 특징으로 하고,

상기 지연부는 직렬로 연결되어 복수의 지연 AND 게이트를 포함하며,

상기 복수의 지연 AND 게이트 각각은 상기 지연부로의 입력과 이전 지연 AND 게이트의 출력을 입력받아 지연시켜 출력하며,

상기 지연 AND 게이트의 수는 다음의 수학적식을 만족하는 것을 특징으로 하는 파이프라인 회로 장치.

$$t_{wire} + t_{dCL} + t_{wire} \geq t_{CL}$$

여기서, t_{CL} 은 조합연산부의 시간 지연, t_{dCL} 은 상기 조합연산부에 대응하는 지연부의 시간 지연, t_{wire} 은 상기 지연부와 연결된 와이어에서의 시간 지연

청구항 6

제5항에 있어서,

상기 어느 일 스테이지의 클럭제공부의 제2 C소자의 출력이 하이레벨인 동안 상기 일 스테이지의 래치부의 제1 데이터가 상기 일 스테이지의 다음 스테이지로 전달되며,

이전 스테이지의 클럭제공부의 제2 C소자의 출력이 하이레벨이 되어, 상기 이전 스테이지의 래치부의 제2 데이터가 상기 일 스테이지로 전달되기 전, 상기 일 스테이지의 클럭제공부의 제2 C소자의 출력이 로우 레벨이 되도록, 다음의 수학적식을 만족하는 것을 특징으로 하는 파이프라인 회로 장치.

$$t_{FwCycle} < t_{BwCycle}$$

$$t_{FwCycle} = t_{wire} + t_{dCL} + t_{wire} + t_{C-element} + t_{wire} + t_{C-element} +$$

$$t_{lc\downarrow} + t_{Latch-Open}$$

$$= t_{Fw.dCL} + 2 \times t_{C-element} + t_{wire} + t_{lc\downarrow} + t_{Latch-Open}$$

$$t_{Fw.dCL} = t_{wire} + t_{dCL} + t_{wire}$$

$$t_{BwCycle} = t_{wire} + t_{C-element} + t_{wire} + t_{C-element} + t_{lc\uparrow} +$$

$$t_{Latch-Open} + t_{CL}$$

$$= 2 \times t_{wire} + 2 \times t_{C-element} + t_{lc\uparrow} + t_{Latch-Open} + t_{CL}$$

여기서, t_{wire} 는 회선에 대한 지연, $t_{C-element}$ 는 C소자에 대한 지연, t_{dCL} 은 지연부에서의 지연, t_{CL} 은 조합논리부에서의 지연, $t_{lc\downarrow}$ 은 하강 에지에서 지연, $t_{lc\uparrow}$ 은 상승 에지에서 지연, $t_{Latch-Open}$ 은 클럭-투-래치 오픈(clock-to-latch open)에 대한 지연, $t_{Latch-Close}$ 은 클럭-투-래치 클로즈(clock-to-latch close)에 대한 지연

발명의 설명

기술 분야

[0001]

본 발명은 파이프라인 회로 장치에 관한 것으로, 보다 상세하게는, 파이프라인 구조를 가지는 회로에서 파이프라인의 각 스테이지에 비동기 방식의 지역 클럭(local clock)으로 동작하는 회로 장치에 관한 것이다.

배경 기술

[0002] 시스템온칩(SoC, System on Chip)은 칩 하나로 모든 기능을 처리하는 반도체로서, 마이크로프로세서, 메모리반도체, 디지털신호처리칩(DSP), 마이크로컨트롤러(MCU) 등 개별 반도체를 하나의 칩에 통합하는 것을 의미한다. 즉, 인쇄회로기판(PCB, printed circuit board) 상에 복수의 반도체 칩으로 구현되던 시스템이 한 개의 칩으로 집적돼, 연산 기능과 데이터의 저장 및 기억, 아날로그와 디지털 신호의 변화 등을 하나의 칩으로 해결하는 것이다. 마이크로프로세서와 마이크로컨트롤러는 이러한 시스템온칩 설계의 기능에 중요한 역할을 수행한다. 특히, 저전력 및 저에너지 소비는 때때로 모바일 시스템온칩 플랫폼에서 임베디드 프로세서 IP에 대한 매우 까다로운 설계 요구사항이다.

선행기술문헌

특허문헌

[0003] (특허문헌 0001) 한국공개특허 제2012-0072125호 2012년 07월 03일 공개 (명칭: 지연 회로, 이를 이용한 FPGA에서의 비동기 회로 시뮬레이션 장치 및 그 방법)

발명의 내용

해결하려는 과제

[0004] 본 발명의 목적은 파이프라인 회로 구조에서 전역 클럭을 사용하지 않고, 파이프라인의 각 스테이지에서 비동기 방식으로 지역 클럭을 제공하는 파이프라인 회로 장치를 제공함에 있다.

과제의 해결 수단

[0005] 상술한 바와 같은 목적을 달성하기 위한 본 발명의 바람직한 실시예에 따른 파이프라인 회로 장치는 파이프라인 형태의 직렬로 배치되는 복수의 조합논리부와, 각각이 복수의 조합논리부와 일대일로 대응하며, 입력되는 클럭 신호에 따라 복수의 조합논리부 사이에서 이전 스테이지의 조합논리부에서 출력된 데이터를 다음 스테이지의 조합논리부로 전달하는 복수의 래치부와, 복수의 래치부 각각과 일대일로 대응하여 복수의 래치부 각각에 비동기 방식으로 클럭 신호를 제공하는 복수의 클럭제공부;를 포함한다.

[0006] 여기서, 클럭제공부는 제1 C소자 및 제2 C소자를 포함하고, 제1 C소자 및 제2 C소자는 각각 입력, 반전 입력, 리셋 및 출력을 가지며, 제1 C소자의 출력은 제2 C소자의 입력이고, 제2 C소자의 출력은 클럭 신호로 대응하는 래치부에 입력되는 동시에 제1 C소자의 반전 입력이 되는 것을 특징으로 한다.

[0007] 클럭제공부의 제1 C소자의 하이 레벨 출력이 클럭제공부의 제2 C소자의 입력에 인가되면 클럭제공부의 제2 C소자의 출력이 하이 레벨이되고, 클럭제공부의 제2 C소자의 하이 레벨 출력이 래치부에 인가되면, 래치부는 상승 에지에서 오픈되어 조합논리부의 데이터를 다음 스테이지로 전달하는 것을 특징으로 한다.

[0008] 제1 C소자 및 제2 C소자 각각은 병렬로 연결되는 제1 내지 제3 AND 게이트와, 직렬로 연결되는 OR 게이트;를 포함하며, 제1 C소자 및 제2 C소자 각각에 대한 입력은 제1 및 제2 AND 게이트에 인가되며, 반전 입력은 제2 및 제3 AND 게이트에 인가되고, 리셋은 제1 내지 제3 AND 게이트에 인가되며, 제1 내지 제3 AND 게이트의 출력은 OR 게이트에 인가되고, OR 게이트의 출력은 제1 및 제3 AND 게이트에 피드백되는 것을 특징으로 한다.

[0009] 복수의 클럭제공부와 일대일 대응하여 대응하는 클럭제공부의 출력을 지연하는 복수의 지연부를 더 포함하는 것을 특징으로 하고, 지연부는 직렬로 연결되어 복수의 지연 AND 게이트를 포함하며, 복수의 지연 AND 게이트 각각은 지연부로의 입력과 이전 지연 AND 게이트의 출력을 입력받아 지연시켜 출력하며, 지연 AND 게이트의 수는 다음의 수학적식을 만족하는 것을 특징으로 한다.

$$[0010] \quad t_{wire} + t_{dCI} + t_{wire} \geq t_{CI}$$

[0011] 여기서, t_{CI} 은 조합연산부의 시간 지연이고, t_{dCI} 은 조합연산부에 대응하는 지연부의 시간 지연이며, t_{wire} 는 지연부와 연결된 와이어에서의 시간 지연이다.

[0012] 어느 일 스테이지의 클럭제공부의 제2 C소자의 출력이 하이레벨인 동안 일 스테이지의 래치부의 제1 데이터가

일 스테이지의 다음 스테이지로 전달되며, 이전 스테이지의 클럭제공부의 제2 C소자의 출력이 하이레벨이 되어, 이전 스테이지의 래치부의 제2 데이터가 일 스테이지로 전달되기 전, 일 스테이지의 클럭제공부의 제2 C소자의 출력이 로우 레벨이 되도록, 다음의 수학적식을 만족하는 것을 특징으로 한다.

$$t_{FwCycle} < t_{BwCycle}$$

$$t_{FwCycle} = t_{wire} + t_{dCL} + t_{wire} + t_{C-element} + t_{wire} + t_{C-element} + t_{lc\downarrow} + t_{Latch-Open} + t_{LC}$$

$$= t_{Fw.dCL} + 2 \times t_{C-element} + t_{wire} + t_{lc\downarrow} + t_{Latch-Open} + t_{LC}$$

$$t_{Fw.dCL} = t_{wire} + t_{dCL} + t_{wire}$$

$$t_{BwCycle} = t_{wire} + t_{C-element} + t_{wire} + t_{C-element} + t_{lc\uparrow} + t_{Latch-Open} + t_{LC}$$

$$= 2 \times t_{wire} + 2 \times t_{C-element} + t_{lc\uparrow} + t_{Latch-Open} + t_{LC}$$

여기서, t_{wire} 는 회선에 대한 지연이고, $t_{C-element}$ 는 C소자에 대한 지연이고, t_{dCL} 은 지연부에서의 지연이고, t_{LC} 은 조합논리부에서의 지연이고, $t_{lc\downarrow}$ 은 하강 에지에서 지연이고, $t_{lc\uparrow}$ 은 상승 에지에서 지연이고, $t_{Latch-Open}$ 은 클럭-투-래치 오픈(clock-to-latch open)에 대한 지연이고, $t_{Latch-Close}$ 은 클럭-투-래치 클로즈(clock-to-latch close)에 대한 지연이다.

발명의 효과

본 발명에 따르면, 본 발명의 비동기 방식의 지역 클럭으로 동작하는 파이프라인 회로 장치는 대응하는 동기 방식의 파이프라인 회로와 비교할 때, 전력 소비를 줄일 수 있다. 특히, 본 발명의 비동기 방식의 파이프라인 회로는 대응하는 동기 방식의 파이프라인 구조의 회로와 비교할 때, 버스트 데이터를 처리하는 경우, 추가의 전력 감소를 이룰 수 있다.

도면의 간단한 설명

도 1은 본 발명의 일 실시예에 따른 파이프라인 회로를 설명하기 위한 도면이다.

도 2a 내지 도 2c는 본 발명의 실시예에 따른 클럭제공부를 설명하기 위한 도면이다.

도 3은 본 발명의 실시예에 따른 비동기 파이프라인 회로의 동작을 설명하기 위한 타이밍도이다.

도 4는 본 발명의 실시예에 따른 비동기 파이프라인 회로의 시간 지연을 설명하기 위한 파이프라인 회로의 도면이다.

도 5는 본 발명의 실시예에 따른 비동기 파이프라인 회로의 시간 지연을 설명하기 위한 타이밍도이다.

도 6은 본 발명의 실시예에 따른 지연부를 설명하기 위한 회로도이다.

도 7은 동기 방식 및 비동기 방식의 클럭 동작을 비교하여 설명하기 위한 그래프이다.

도 8은 본 발명의 실시예에 따라 각 파이프라인의 스테이지에 비동기 방식으로 지역 클럭을 제공하도록 구성된 프로세서의 일 예를 도시한 블록도이다.

도 9는 대기 비율(idling-ratio)을 다양하게 변화함에 따른 전력 소비에 대한 그래프이다.

발명을 실시하기 위한 구체적인 내용

- [0022] 하기의 설명에서는 본 발명의 실시예를 이해하는데 필요한 부분만이 설명되며, 그 이외 부분의 설명은 본 발명의 요지를 흐트리지 않도록 생략될 것이라는 것을 유의하여야 한다.
- [0023] 이하에서 설명되는 본 명세서 및 청구범위에 사용된 용어나 단어는 통상적이거나 사전적인 의미로 한정해서 해석되어서는 아니 되며, 발명자는 그 자신의 발명을 가장 최선의 방법으로 설명하기 위해 용어의 개념으로 적절하게 정의할 수 있다는 원칙에 입각하여 본 발명의 기술적 사상에 부합하는 의미와 개념으로 해석되어야만 한다. 따라서 본 명세서에 기재된 실시예와 도면에 도시된 구성은 본 발명의 바람직한 실시예에 불과할 뿐이고, 본 발명의 기술적 사상을 모두 대변하는 것은 아니므로, 본 출원시점에 있어서 이들을 대체할 수 있는 다양한 균등물과 변형예들이 있을 수 있음을 이해하여야 한다.
- [0024] 비동기 회로 설계 기술은 CMOS 설계 기법에서 높은 설계 요구사항에 대한 유망한 대안으로서 고려되며, 비동기 회로 설계 방법은 클럭 분배 또는 시스템 동기화 문제에 대한 가능한 해법을 제공할 수 있다. 하지만, 비동기 회로 기술을 사용함으로써, 클럭 네트워크에 의한 전력 소비가 제거될 수 있지만, 핸드셰이킹 오버헤드는 성능 손실 및 초과 전력 소비를 유발할 수 있다. 본 발명은 이러한 핸드셰이킹 기법을 이용하면서도 성능 손실이 없고, 저전력 소비를 제공하는 회로의 설계를 제공한다.
- [0025] 도 1은 본 발명의 일 실시예에 따른 파이프라인 회로를 설명하기 위한 도면이다.
- [0026] 도 1을 참조하면, 본 발명의 실시예에 따른 파이프라인 회로는 복수의 모듈이 파이프라인 형식으로 직렬 연결되는 구조를 가진다. 여기서, 하나의 모듈은 파이프라인의 하나의 스테이지를 이룬다. 어느 하나의 모듈은 클럭제공부(100), 지연부(200), 래치부(300), 조합논리부(400)를 포함한다. 즉, 파이프라인 회로는 복수의 클럭제공부(100: 101, 102, 103), 복수의 지연부(200: 201, 202, 203, 204), 복수의 래치부(300: 301, 302, 303) 및 복수의 조합논리(Combinational logic)부(400, 401, 402, 403)를 포함한다.
- [0027] 클럭제공부(100: 101, 102, 103)는 래치부(300: 301, 302, 303)에 비동기 방식으로 래치부(300: 301, 302, 303) 각각에 클럭을 제공한다. 즉, 클럭제공부(100: 101, 102, 103) 각각은 복수의 래치부(300: 301, 302, 303)에 일대일 대응하며, 전역 클럭(global clock)이 아닌 지역 클럭(local clock)으로 복수의 래치부(300: 301, 302, 303) 각각에 클럭을 제공한다. 예컨대, 제1 클럭제공부(101)은 제1 래치부(301)에 클럭을 제공하고, 제2 클럭제공부(102)는 제2 래치부(302)에 클럭을 제공하며, 제3 클럭제공부(103)는 제3 래치부(303)에 클럭을 제공한다.
- [0028] 지연부(200: 201, 202, 203, 204) 각각은 클럭제공부(100: 101, 102, 103) 혹은 조합논리부(400, 401, 402, 403)에 일대일 대응한다. 지연부(200: 201, 202, 203, 204)는 이전 스테이지의 클럭제공부(100)로부터 출력된 신호를 시간 지연한 후 대응하는 클럭제공부(100)에 입력으로 제공한다. 본 발명의 실시예에 따르면, 지연부(200: 201, 202, 203, 204) 각각의 지연 시간은 대응하는 조합논리부(400, 401, 402, 403)의 연산 시간에 따라 결정될 수 있으며, 아래에서 보다 상세하게 설명될 것이다.
- [0029] 래치부(300: 301, 302, 303)는 적어도 하나의 래치(latch)로 이루어질 수 있다. 이러한 래치(latch)는 입력되는 클럭 신호에 따라, 레벨(예컨대, high, low 또는 1, 0)이 변경되며, 레벨에 따라 조합논리부(400, 401, 402, 403)로부터 출력되는 데이터를 다음 스테이지로 전달할 수 있다. 본 발명의 실시예에서 래치는 하이 레벨(high level, 1)인 동안 입력된 데이터를 출력하는 것으로 설명한다. 하지만, 이 기술분야에서 통상의 지식을 가진 자라면 그 역의 경우(로우 레벨에서 동작)에도 본 발명이 실시될 수 있음을 이해할 수 있을 것이다. 이러한 래치의 동작으로, 래치부(300)는 클럭제공부(100)로부터의 클럭 신호에 따라 동 스테이지의 조합논리부(400)의 출력을 다음 스테이지의 조합논리부(400)의 입력으로 전달할 수 있다. 한편, 래치부(300)는 래치 대신 적어도 하나의 플립플롭을 이용하여 구현될 수도 있다.
- [0030] 조합논리부(400, 401, 402, 403)는 연산을 위한 회로이며, 시독립논리(time-independent logic) 회로라고 칭하기도 한다. 이는 디지털 논리의 일 형태이며, 불린 회로(Boolean circuits)에 의해 구현된다. 조합논리부(400, 401, 402, 403)는 오직 현재의 입력을 연산하여 출력한다. 예컨대, 이러한 조합논리부(400, 401, 402, 403)는 ALU(arithmetic logic unit)를 대표적으로 예시할 수 있다.
- [0031] 그러면, 본 발명의 실시예에 따른 클럭제공부(100)에 대해서 보다 상세하게 설명하기로 한다. 도 2a 내지 도 2c

는 본 발명의 실시예에 따른 클럭제공부를 설명하기 위한 도면이다.

- [0032] 도 2a를 참조하면, 본 발명의 실시예에 따른 클럭제공부(100)는 두 개의 C소자(C-element, 110: 111, 112)가 핸드셰이킹 형식으로 연결되는 구조를 가진다. 즉, 본 발명의 실시예에 따른 클럭제공부(100)는 제1 C소자(111) 및 제2 C소자(112)를 포함한다.
- [0033] 도 2b에 도시된 바와 같이, 어느 하나의 C소자(100)는 입력(a), 반전 입력(b), 리셋(reset) 및 출력(out)을 가진다. 이러한 C소자(100)는 복수의 게이트로 구성될 수 있다. 복수의 게이트로 구성된 C소자(100)의 일 예를 도 2c에 도시하였다.
- [0034] 도 2c에 도시된 바와 같이, C소자(100)는 제1 내지 제3 AND 게이트(11, 12, 13)가 병렬로 연결되며, 하나의 OR 게이트(14)가 직렬로 연결되는 구조를 가진다. 또한, 입력(a) 신호는 제1 및 제2 AND 게이트(11, 12)의 입력으로 인가되며, 반전 입력(b) 신호는 제2 및 제3 AND 게이트(12, 13)의 입력으로 인가된다. 또한, 리셋(reset) 신호는 3개의 AND 게이트(11, 12, 13)의 입력으로 인가될 수 있다. 제1 내지 제3 AND 게이트(11, 12, 13)의 출력은 OR 게이트(14)의 입력으로 인가된다. 또한, OR 게이트(14)의 출력은 제1 및 제3 AND 게이트(11, 13)의 입력으로 인가된다.
- [0035] 한편, 도 2a 및 도 2b를 다시 참조하면, 제1 C소자(111)의 출력은 제2 C소자(112)의 입력(a)이 된다. 또한, 제2 C소자(112)의 출력은 클럭 신호로 래치부(300)로 입력되며, 동시에 제1 C소자(111)의 반전 입력(b)이 된다.
- [0036] 다음으로, 본 발명의 실시예에 따른 비동기 파이프라인 회로의 동작에 대해서 상세하게 설명하기로 한다. 도 3은 본 발명의 실시예에 따른 비동기 파이프라인 회로의 동작을 설명하기 위한 타이밍도이다.
- [0037] 도 3의 타이밍도에서 설명의 편의를 위하여, 3개의 연속된 스테이지, 즉, 스테이지 i, j, k가 있는 것으로 가정한다. 또한, out-i, out-i, out-j, out-j, out-k 및 out-k는 각각 스테이지 i, 스테이지 j 및 스테이지 k의 제1 C소자(111) 및 제2 C소자(112)의 출력을 나타낸다. 즉, 스테이지 i의 제1 C소자(111) 및 제2 C소자(112)의 출력은 각각 out-i, out-i이고, 스테이지 j의 제1 C소자(111) 및 제2 C소자(112)의 출력은 각각 out-j, out-j이며, 스테이지 k의 제1 C소자(111) 및 제2 C소자(112)의 출력은 각각 out-k, out-k이다.
- [0038] 도 1 내지 도 3을 참조하면, 데이터 1 내지 데이터 4(①, ②, ③, ④)는 클럭제공부(100)가 출력하는 클럭 신호에 따라 래치부(300)가 다음 스테이지의 조합논리부(400)로 전달하는 데이터이다.
- [0039] 클럭제공부(100)의 출력중 제1 C소자(111)의 출력 out-i, out-j, out-k이 하이 레벨이면, 이러한 출력은 제2 C소자(112)의 입력(a)에 인가되며, 제2 C소자(112)의 출력 out-i, out-j, out-k는 하이 레벨이 된다. 예컨대, 제2 클럭제공부(102)의 제1 C소자(111)의 출력 out-j이 하이 레벨이면, 이러한 출력(out-j, high)은 제2 클럭제공부(102)의 제2 C소자(112)의 입력(a)에 인가되며, 제2 클럭제공부(102)의 제2 C소자(112) 출력 out-j는 하이 레벨이 된다. 이와 같이, 클럭제공부(100)의 출력중 제2 C소자(112)의 출력 out-i, out-j, out-k가 하이 레벨(high level)이면, 래치부(300)는 데이터를 다음 스테이지의 조합논리부(400)로 전달한다. 예컨대, out-j가 하이 레벨인 동안, 제2 래치부(302)는 스테이지 i의 제2 조합논리부(402)로부터 출력된 데이터를 스테이지 k의 제3 조합논리부(403)로 전달한다.
- [0040] 제1 C소자(111)의 출력 out-i, out-j, out-k이 로우 레벨이고, 다음 스테이지의 제1 C소자(111)의 출력 out-i, out-j, out-k이 하이 레벨이면, 제2 C소자(112)의 출력 out-i, out-j, out-k는 로우 레벨이 된다. 예컨대,
- [0041] 제2 클럭제공부(102)의 제1 C소자(111)의 출력 out-j가 로우 레벨이면, 이는 제2 클럭제공부(102)의 제2 C소자(112)의 입력(a)에 인가되며, 제3 클럭제공부(103)의 제1 C소자(111)의 출력 out-k가 하이 레벨이면, 이는 피드백되어 제2 클럭제공부(102)의 제2 C소자(112)의 반전 입력(b)에 인가된다. 이에 따라, 제2 클럭제공부(102)의 제2 C소자(102)의 출력 out-j는 로우 레벨이 된다. 이와 같이, 클럭제공부(100)의 출력중 제2 C소자(112)의 출력 out-i, out-j, out-k가 로우 레벨(low level)이면, 래치부(300)는 데이터를 다음 스테이지의 조합논리부(400)로 전달한다. 예컨대, out-j가 로우 레벨인 동안, 제2 래치부(302)는 스테이지 i의 제2 조합논리부(402)로부터 출력된 데이터를 스테이지 k의 제3 조합논리부(403)로 전달하지 않는다. 즉, out-j가 로우 레벨인 동안, 제2 래치부(302)는 스테이지 i의 제2 조합논리부(402)의 데이터를 일시 저장한다.
- [0042] 보다 상세히, 스테이지 j에서 데이터 1의 전달 과정을 살펴보면, 다음과 같다. 제2 클럭제공부(102)의 제1 C소자(111)의 출력 out-j가 하이 레벨이면, S100 단계에서 이는 제2 클럭제공부(102)의 제2 C소자(112)의 입력(a)으로 전달된다. 그러면, 제2 클럭제공부(102)의 제2 C소자(112)의 출력 out-j는 하이 레벨이된다. 제2 클럭제공부(102)의 제2 C소자(112)의 출력 out-j가 하이 레벨이되면, 제2 래치부(302)는 하이 레벨이 되는 상승 에지에

서 오픈(open)되어 출력 out-j가 하이 레벨인 동안 데이터가 다음 스테이지(스테이지 k)로 전달된다. 다음 스테이지로 전달된 데이터는 다음 스테이지의 제3 조합논리부(403)로 제공되며, 제3 조합논리부(403)에서의 연산 과정을 거친 후 제3 래치부(303)에 저장될 것이다.

[0043] 이때, S110 단계에서 제2 클럭제공부(102)의 출력 out-j는 순방향으로 다음 스테이지를 진행하도록 요청하기 위한 요청 신호로 제3 지연부(203)를 거쳐 제3 클럭제공부(103)의 제1 C소자(111)로 전달된다. 또한, S120 단계에서 제2 클럭제공부(102)의 출력 out-j는 역방향으로 수신 확인 신호를 전송한다.

[0044] S110 단계의 순방향으로 전파된 요청 신호는 제3 클럭제공부(103)의 제1 C소자(111)의 입력(a)이 된다. 이에 따라, S130 단계에서 제3 클럭제공부(103)의 제1 C소자(111)의 출력 out-k는 하이 레벨이며, 이러한 하이레벨 출력 out-k는 S120 단계에서 제2 클럭제공부(102)의 제2 C소자(112)의 반전 입력(b)이 된다.

[0045] 한편, S130 단계에서, 역방향으로 전송된 하이 레벨 출력 out-j는 제2 클럭제공부(102)의 제1 C소자(111)의 반전 입력(b)이 된다. 이에 따라, 제2 클럭제공부(102)의 제1 C소자(111)의 출력 out-j는 로우 레벨이 된다. 이러한 로우 레벨 출력 out-j는 S140 단계에서, 제2 클럭제공부(102)의 제2 C소자(112)의 입력(a)이 된다.

[0046] 이와 같이, 하이 레벨 출력 out-k가 제2 클럭제공부(102)의 제2 C소자(112)의 반전 입력(b)이 되고, 로우 레벨 출력 out-j가 제2 클럭제공부(102)의 제2 C소자(112)의 입력(a)이 되면, 제2 클럭제공부(102)의 출력 out-j는 로우 레벨이 된다. 이러한 out-j의 하강 에지에서, 제2 래치부(302)는 닫히고(close), 해당 데이터는 제2 래치부(302)에 임시 저장된다.

[0047] 도 4는 본 발명의 실시예에 따른 비동기 파이프라인 회로의 시간 지연을 설명하기 위한 파이프라인 회로의 도면이며, 도 5는 본 발명의 실시예에 따른 비동기 파이프라인 회로의 시간 지연을 설명하기 위한 타이밍도이다.

[0048] S210 내지 S250 단계를 통해, 데이터 1을 전달하는 이벤트가 완료되기 전, 데이터 2가 전달되는 이벤트가 발생하는 상황에 대해서 설명하기로 한다. 도 4 및 도 5를 참조하면, 제2 클럭제공부(102)의 출력 out-j가 하이 레벨이되면, 제2 래치부(302)는 하이 레벨이 되는 상승 에지에서 오픈(open)되어 출력 out-j가 하이 레벨인 동안 데이터 1이 다음 스테이지(스테이지 k)로 전달된다. 다음 스테이지로 전달된 데이터 1은 다음 스테이지의 제3 조합논리부(403)로 제공되며, 제3 조합논리부(403)에서의 연산 과정을 거친 후 제3 래치부(303)에 저장될 것이다.

[0049] 이때, S210 단계에서 제2 클럭제공부(102)의 출력 out-j는 순방향으로 다음 스테이지를 진행하도록 요청하기 위한 요청 신호로 제3 지연부(203)를 거쳐 제3 클럭제공부(103)의 제1 C소자(111)로 전달된다. 또한, S230 단계에서 제2 클럭제공부(102)의 출력 out-j는 역방향으로 수신 확인 신호를 전송한다.

[0050] S210 단계의 순방향으로 전파된 요청 신호는 제3 클럭제공부(103)의 제1 C소자(111)의 입력(a)이 된다. 이에 따라, S220 단계에서 제3 클럭제공부(103)의 제1 C소자(111)의 출력 out-k는 하이 레벨이며, 이러한 출력 out-k는 제2 클럭제공부(102)의 제2 C소자(112)의 반전 입력(b)이 된다. 그러면, 제2 클럭제공부(102)의 출력 out-j는 로우 레벨이 된다. 이러한 out-j의 로우 레벨 출력의 하강 에지에서, 제2 래치부(302)는 닫히고(close), 더 이상의 데이터 전달은 이루어지지 않는다. 이러한 S210 및 S220 단계의 순방향 사이클에 소요되는 순방향 사이클 시간은 $t_{FwCycle}$ 로 표시하였다.

[0051] 한편, S230 단계에서 역방향으로 전송된 하이 레벨 출력 out-j는 제2 클럭제공부(102)의 제1 C소자(111)의 반전 입력(b)이 된다. 이에 따라, 제2 클럭제공부(102)의 제1 C소자(111)의 출력 out-j는 로우 레벨이 되며, 이는 제2 클럭제공부(102)의 제2 C소자(112)의 입력(a)이 되기도 하지만, S240 단계에서와 같이, 제1 클럭제공부(101)의 제2 C소자(112)의 반전 입력(b)이 될 수 있다.

[0052] 그러면, S250 단계에서와 같이, 제1 클럭제공부(101)의 제2 C소자(112)는 새로운 데이터, 즉, 데이터 2를 판독하도록 하는 상승 신호 이벤트를 생성할 수 있다. 그러면, 데이터 2가 스테이지 j에서 제2 래치부(302)의 입력으로 제2 조합논리부(402)를 통해 전파된다. 이러한 S220 내지 S250 단계의 역방향 사이클에 소요되는 역방향 사이클 시간을 $t_{BwCycle}$ 로 표시하였다.

[0053] 만약, $t_{FwCycle} \geq t_{BwCycle}$ 이면, 데이터 1을 전달하는 이벤트가 완료되기 전, 데이터 2가 전달되는 이

벤트가 발생할 수 있다. 따라서, 본 발명의 실시예에 따른 클럭은 다음의 조건 $t_{FwCycle} < t_{BwCycle}$ 를 만족해야만 한다. 이러한 조건에 대해 보다 상세하게 설명하면 다음과 같다.

먼저, $t_{FwCycle}$ 은 다음의 수학적 식 1과 같다.

수학적 식 1

$$\begin{aligned} t_{FwCycle} &= t_{wire} + t_{dCL} + t_{wire} + t_{C-element} + t_{wire} \\ &\quad + t_{C-element} + t_{lc\downarrow} + t_{Latch-Close} \\ &= t_{Fw,dCL} + 2 \times t_{C-element} + t_{wire} + t_{lc\downarrow} \\ &\quad + t_{Latch-Close} \end{aligned}$$

$$t_{Fw,dCL} = t_{wire} + t_{dCL} + t_{wire}$$

또한, $t_{BwCycle}$ 는 다음의 수학적 식 2와 같다.

수학적 식 2

$$\begin{aligned} t_{BwCycle} &= t_{wire} + t_{C-element} + t_{wire} + t_{C-element} \\ &\quad + t_{lc\uparrow} + t_{Latch-Open} + t_{CL} \\ &= 2 \times t_{wire} + 2 \times t_{C-element} + t_{lc\uparrow} + t_{Latch-Open} + t_{CL} \end{aligned}$$

수학적 식 1 및 2에서, t_{wire} 는 회로 구성 사이의 회선(wire)에 대한 지연이며, $t_{C-element}$ 는 C소자(110)에 대한 지연을 의미한다. 또한, t_{dCL} 은 지연부(200)에서의 지연, t_{CL} 은 조합논리부(400)에서의 지연을 의미한다. $t_{lc\downarrow}$ 및 $t_{lc\uparrow}$ 은 지역 클럭 신호에 대한 지연이며, 각각 하강 및 상승 클럭 신호(제2 소자부(112)의 로우 레벨 출력 및 하이 레벨 출력)에 대한 지연이다. $t_{Latch-Open}$, $t_{Latch-Close}$ 은 각각 클럭-투-래치 오픈(clock-to-latch open) 및 클럭-투-래치 클로즈(clock-to-latch close)에 대한 지연이다. 클럭-투-래치 오픈은 래치부(300)의 각 래치의 클럭 포트에서 상승 클럭 신호(하이 레벨)가 입력되는 시간으로부터 래치가 오픈(open)되는 시간까지의 지연이며, 클럭-투-래치 클로즈는 래치부(300)의 각 래치의 클럭 포트에서 하강 클럭 신호(로우 레벨) 이벤트의 시간으로부터 래치가 클로즈(close)되는 시간까지의 지연이다.

$t_{FwCycle}$ 에서 최소 3개의 지연 요소 $t_{wire} + t_{dCL} + t_{wire}$ 의 합은 조합논리부(400)의 지연과 일치하며, 이에 따라, $t_{Fw,dCL}$ 로 나타낼 수 있다. 결과적으로, $t_{Fw,dCL}$ 은 t_{CL} 보다 크거나 또는 동일하다.

여기서, 회로 소자 중 동일한 타입은 동일한 지연을 가진다고 가정한다. 예컨대, 즉, 모든 지연부(201, 202, 203)는 동일한 지연을 가지며, 모든 조합논리부(401, 402, 403) 또한 동일한 지연을 가진다고 가정한다.

또한, $t_{Fw.dCL} \approx t_{CL}$ 이고, $t_{lc\downarrow} \approx t_{lc\uparrow}$ 이며, $t_{Latch-Cl\text{ose}} \approx t_{Latch-Open}$ 이라고 가정한다.

그러면, $t_{FwCycle} < t_{BwCycle}$ 은 다음의 수학적 식 3과 같이 간소화될 수 있다.

수학적 식 3

$$\begin{aligned} t_{FwCycle} &< t_{BwCycle} \\ \Leftrightarrow t_{Fw.dCL} + 2 \times t_{C\text{-element}} + t_{wire} + t_{lc\downarrow} + t_{Latch-Cl\text{ose}} \\ &< 2 \times t_{wire} + 2 \times t_{C\text{-element}} + t_{lc\uparrow} + t_{Latch-Open} + t_{CL} \\ \Leftrightarrow t_{Fw.dCL} + t_{lc\downarrow} + t_{Latch-Cl\text{ose}} \\ &< t_{wire} + t_{lc\uparrow} + t_{Latch-Open} + t_{CL} \\ \Leftrightarrow 0 &< t_{wire} \end{aligned}$$

여기서, 수학적 식 3은 $t_{Fw.dCL} \approx t_{CL}$ 이고, $t_{lc\downarrow} \approx t_{lc\uparrow}$ 이며, $t_{Latch-Cl\text{ose}} \approx t_{Latch-Open}$ 이면, $t_{FwCycle} < t_{BwCycle}$ 이 만족함을 보인다.

도 6은 본 발명의 실시예에 따른 지연부를 설명하기 위한 회로도이다.

도시된 바와 같이, 지연부(200)는 복수의 지연 AND 게이트(211, 212, 213, 214)가 직렬로 연결되며, 이때, 지연부(200)의 입력은 복수의 지연 AND 게이트(211, 212, 213, 214)에 의해 지연된 후 출력된다. 각 지연 AND 게이트는 지연부(200)로의 입력과 앞선 지연 AND 게이트의 출력을 입력으로 받아 이루지기 때문에, 지연 AND 게이트

의 수에 따라 지연 시간이 결정된다. 이에 따라, 지연 AND 게이트의 수는 $t_{Fw.dCL} = t_{wire} + t_{dCL} + t_{wire} \approx t_{CL}$ 혹은 $t_{Fw.dCL} (= t_{wire} + t_{dCL} + t_{wire}) \geq t_{CL}$ 을 만족하도록 구성된다.

보다 상세하게 설명하면, 제2 지연 AND 게이트(212)는 2개의 입력을 가지며, 이는 지연부(200)에 대한 입력과 제1 지연 AND 게이트(211)의 출력을 입력으로 가진다. 마찬가지로, 제3 지연 AND 게이트(213)는 2개의 입력을 가지며, 이는 지연부(200)에 대한 입력과 제2 지연 AND 게이트(212)의 출력을 입력으로 가진다. 또한, 제4 지연 AND 게이트(213)도 2개의 입력을 가지며, 이는 지연부(200)에 대한 입력과 제3 지연 AND 게이트(213)의 출력을 입력으로 가진다. 이에 따라, 지연부(200)의 출력은 제1 내지 제4 지연 AND 게이트(211, 212, 213, 214)에 의해 지연된다.

SoC에서 실행되는 작업은 시간이 지남에 따라 다를 수 있으며, 때때로 시스템에 사용되는 데이터 패턴은 버스트 형식(bursty)으로 입력될 수 있다. 특히, 이러한 데이터 패턴은 동기 회로를 사용하는 경우, 연속적으로 전력 혹은 에너지를 소모하는 프리러닝 클럭(free-running clock)에 기인한 문제가 발생할 수 있다. 클럭 게이팅(clock gating)이 회로에 공급되는 클럭을 정지시키기 위해 사용될 수 있는 슬립 또는 대기 상태인 경우에도, 절전 단계 및 활성 상태 사이의 임시 시점에서 추가 타이밍 및 파워 오버헤드가 요구된다. 뿐만 아니라, 클럭 게이팅은 설계자가 회로의 어떤 부분이 클럭킹없이 대기하여야 하는지, 언제 클럭을 중지하여야 하는지, 그리고, 클럭을 얼마나 오랫동안 중지하여야 하는지에 대한 정보를 알고 있어야 한다. 또한, 이러한 정보는 메모리에 저장하여야 하며, 별도의 제어 회로가 메모리에 저장된 정보에 따라 회로의 특정 부분에 대한 클럭킹을 중지하기 위해 적절한 제어신호를 생성해야 한다. 따라서, 설계에 있어서 클럭 게이팅 응용의 문제는 오버헤드를 감소시키기 위해 중요하게 고려된다.

프로세서 코어에 대한 동적 전력 소비를 위한 일반적인 방정식은 다음의 수학적 식 4 및 5와 같다.

수학식 4

[0070]

$$p = c_{total} \cdot V^2 \cdot f$$

수학식 5

[0071]

$$p = (c_{logic} + c_{signal} + c_{clock}) \cdot V^2 \cdot f$$

[0072]

식 4에서, c_{total} 은 유효 용량, V 는 공급 전압 레벨이고, f 는 주파수이다. c_{total} 은 동기 회로에서 식 5에서와 같이 세 가지 구성 요소 c_{logic} , c_{signal} , c_{clock} 이 될 수 있다. c_{logic} 은 논리 회로의 전력 소비, c_{signal} 은 시그널링에 따른 전력 소비, 그리고, c_{clock} 은 클럭에 따른 전력 소비를 의미한다. 여기서, 동기 회로의 경우, c_{clock} 은 유휴 상태에서도 지속적으로 전력 소비에 기여한다. 반면, 비동기 방식의 회로의 경우, 이벤트 구동 핸드셰이크 프로토콜에 따라 활성화되는 지역적으로 생성된 클럭을 사용하기 때문에, c_{clock} 은 코어의 대기 시간 동안에는 전력 및 에너지 누출에 기여하지 않는다.

[0073]

도 7은 동기 방식 및 비동기 방식의 클럭 동작을 비교하여 설명하기 위한 그래프이다.

[0074]

도 7에서, 대기 시간(Idle phase time)은 코어가 유효한 연산을 수행하지 않는 기간으로 정의하며, 동작 시간(working phase time)은 코어가 유효한 연산을 수행하는 기간으로 정의된다.

[0075]

클럭 주파수 성분은 동작 시간 동안의 주파수 f_{work} 와 대기 시간 동안의 주파수 f_{idle} 의 두 가지 형태의 주파수로 분류될 수 있다. 활성화 팩터로 α 를 사용할 때, $f_{work} = \alpha \cdot f$ 이고, $f_{idle} = (1 - \alpha) \cdot f$ 이다. 여기서, f 는 다음의 수학식 6과 같다.

수학식 6

[0076]

$$f = f_{work} + f_{idle}$$

[0077]

f_{work} 는 유효한 연산을 포함하는 주파수 부분인 반면, f_{idle} 은 어떠한 연산도 수행하지 않는 주파수 부분이다. 동기 회로에서 전력 소비를 P_{sync} 라고 할 때, 동기 회로에서 전력 소비는 다음의 수학식 7과 같고, 비동기 회로에서 전력 소비를 P_{async} 라고 하면, 비동기 회로에서 전력 소비는 다음의 수학식 8과 같다.

수학식 7

$$P_{sync} = (c_{sync_{logic}} + c_{sync_{signal}} + c_{sync_{clock}}) \cdot V^2 \cdot f_{work}$$

$$+ c_{sync_{clock}} \cdot V^2 \cdot f_{idle}$$

수학식 8

$$P_{async} = (c_{async_{logic}} + c_{async_{signal}} + c_{async_{clock}}) \cdot V^2 \cdot f_{work}$$

수학식 8에서 보인 바와 같이, P_{async} 의 경우, 전력 소비가 0이기 때문에 f_{idle} 부분이 없다.

이제, 비동기 회로의 전력 소비가 동기 회로의 전력 소비보다 더 낮을 때의 동작 조건에 대해서 설명하며, 다음의 수학식 9와 같다.

수학식 9

$$P_{async} \leq P_{sync} \Leftrightarrow P_{sync} - P_{async} \geq 0$$

식 9를 요약하면, 다음의 수학식 10과 같은 조건이 유도된다.

수학식 10

$$\frac{f_{idle}}{f_{work}} \geq \frac{\Delta C_{logic} + \Delta C_{signal} + \Delta C_{clock}}{C_{sync_{clock}}}$$

where,

$$\Delta C_{logic} = C_{async_{logic}} - C_{sync_{logic}},$$

$$\Delta C_{signal} = C_{async_{signal}} - C_{sync_{signal}},$$

$$\Delta C_{clock} = C_{async_{clock}} - C_{sync_{clock}}$$

마지막으로, 수학식 10은 다음의 수학식 11과 같이 다시 쓸 수 있다.

수학식 11

$$\frac{f_{idle}}{f_{work} + f_{idle}} \geq \frac{\Delta C_{logic} + \Delta C_{signal} + \Delta C_{clock}}{C_{sync_clock} + \Delta C_{logic} + \Delta C_{signal} + \Delta C_{clock}}$$

따라서 동작 시간(working phase time) 동안 비동기 회로의 전력 소비가 대응하는 동기 회로의 전력 소비보다 더 많더라도 비동기 회로는 평균 전력 소비의 비용에 있어서 동기 회로보다 뛰어날 수 있다. 만약,

$\frac{f_{idle}}{f_{work} + f_{idle}}$ 비율이 수학식 11의 오른쪽 부분으로부터 계산된 값보다 더 높으면 이것은 참이 된다. 이러한 비율을 대기 비율(idling-ratio)라고 정의하고, β 라고 표시하기로 한다. 또한, ΔC_{logic}

ΔC_{signal} , ΔC_{clock} 는 모두 C_{sync_clock} 에 비해 작다고 가정하면, 비동기 회로는 낮은 대기 비율(idling-ratio)을 가지는 동기 회로보다 전력을 덜 소비함을 알 수 있다.

다음으로, 진술한 비동기 회로의 전력 소비 절감에 대한 구체적인 실시예를 들어 설명하기로 한다. 도 8은 본 발명의 실시예에 따라 각 파이프라인의 스테이지에 비동기 방식으로 지역 클럭을 제공하도록 구성된 프로세서의 일 예를 도시한 블록도이다.

여기서, 도 8에 도시된 프로세서는 본 발명의 실시예에 따른 파이프라인 회로의 일 예이다. 도시된 바와 같이, 복수의 조합연산부(IMEM, RF, ALU, DMEM, WB 등), 조합연산부 사이에 배치되는 복수의 래치부(L1 내지 L5), 래치부에 지역 클럭을 제공하는 복수의 클럭제공부(C^2 -HC) 및 복수의 클럭제공부 사이에서 조합연산부(IMEM, RF, ALU, DMEM, WB 등)의 연산 시간이 보장되도록 시간을 지연하는 지연부(d_F , d_E , d_E , d_M)를 포함한다. 본 발명의 효과를 도 8에 도시된 예시적인 프로세서를 이용하여 설명하기로 한다.

도 8에 도시된 클럭제공부(C^2 -HC)를 기반으로 한 비동기 프로세서는 Verilog-HDL을 사용하여 개발되었고, Xilinx ISE Design Suite 13.4 로 합성되었다. 시뮬레이션 및 측정을 위해 Virtex-5 LX110t(스피드 그레이드-3) 및 Spartan-6 LX9 개발 보드를 사용한다. 시뮬레이션 결과는 45nm 기술을 기반으로 한 Virtex-5 장비의 전압 및 온도 변화 영향을 조사하기 위해, 다음과 같은 두 개의 경계 조건(corner conditions)을 갖는 처리 성능의 변화를 관찰한다. 여기서, 최상의 동작 조건은 코어 전압 = 1.05V, 온도 = 0이며, 최악의 동작 조건은 코어 전압 = 0.95V, 온도 = 85이다.

Virtex-5 장비상에서 비동기 방식의 경우, 최상 및 최악 동작 조건의 처리 성능은 각각 122.5 MHz와 106.1MHz이다. 주어진 전압 및 온도의 변화 범위에서 그 성능은 최악의 경우 최상의 경우 보다 대략 1.15초 더 느리다. Spartan-6상에서의 설계는 50.6MHz 처리 성능을 나타낸다. 한편, 동기 방식을 가지는 프로세서에서, 최상의 경우 및 최악의 경우 동작 조건에 대하여 Virtex-5상에서 처리 성능은 각각 216.9MHz와 195.3MHz가 관찰되었다. 이 동기 방식의 경우, 최악 조건 성능은 최상 경우 성능보다 대략 1.11초 더 느리다.

표 1 및 표 2는 동기 프로세서 및 비동기 프로세서의 코어가 각각 하이엔드 Virtex-5(106.1 MHz)와 저전력 Spartan-6(50.6 MHz) 장비로 구성될 때 자원 구성 형태에 의한 코어의 동적인 전력 소비를 나타낸다. Spartan-6 장비에 대해, 1.16V 및 85의 조건이 전력 평가를 위해 사용되었다.

표 1

Dynamic Power	Sync. MIPS	Async. MIPS	+/-
Logic Power	2.58mW	4.14mW	(+)60%
Signal Power	13.12mW	12.83mW	(-)2.2%
Clock Power	47.53mW	22.95mW	(-)51.7%
Total Power	63.23mW	39.92mW	(-)36.8%

[0098]

표 2

Dynamic Power	Sync. MIPS	Async. MIPS	+/-
Logic Power	2.02mW	2.79mW	(+)38.1%
Signal Power	5.24mW	6.28mW	(+)19.8%
Clock Power	5.35mW	6.78mW	(+)26.7%
Total Power	12.61mW	15.85mW	(+)25.6%

[0099]

[0100]

공정한 비교를 위해, 동기 프로세서의 클럭 주파수가 비동기 프로세서의 코어의 평균 주파수로 설정된다. 표 1에서 나타난 바와 같이, 고성능 Virtex-5 에서 클럭 자원을 위한 전력 소비의 상당한 감소 덕분에 36.8%의 소비 전력 감소가 관찰되었다. 다른 한편, 표 2의 저전력 Spartan-6 장비에서 25.6% 이상의 소비 전력 증가가 관찰되었다.

[0101]

동기 프로세서와 비동기 프로세서의 자원 사용은 거의 동일하다. LUT 자원에서 단지 작은 증가가 비동기 프로세서에서 관찰되었다. LUT 자원의 증가는 비동기 핸드셰이크 컨트롤러에 추가된 지연 소자에 의해 발생된다. 논리 소자의 증가는 논리 전력 소비의 증가에 기여한다.

[0102]

전역 클럭 버퍼 및 낮은 왜곡 클럭 신호 자원이 동기 코어에서 사용됨을 주목한다. FPGA 장비의 클럭 버퍼 및 신호 라인은 FPGA상의 칩 회로가 동작할 수 있는 최대 클럭 주파수를 제공하도록 설계되어야 한다. 따라서 장비 또는 장비의 일부분이 최대 주파수에서 동작하지 않는다면 FPGA 칩은 이러한 오버 설계 구성 때문에 전력을 낭비할 것이다. Virtex-5 장비의 동기 설계에서 전력 소비의 상당한 증가는 주로 전력 소모 구성요소를 때문이다. 이러한 결과는 고성능 및 고용량 FPGA에서 비동기 회로를 채택하여 전력 소모를 절약할 수 있다는 것을 의미한다.

[0103]

한편, 저전력 Spartan-6에서, 클럭 라인은 저전력 소비를 위해 최적화되었다. 게다가, Spartan-6 장비는 아주 작은 로직 용량 및 와이어 자원을 가진 LX9 시리즈 모델이다. 낮은 로직 용량과 매우 제한된 자원으로 특정화된 이러한 칩에서, 비동기 프로세스 코어는 저전력을 소비하는 쇼트 와이어가 데이터 신호 및 비동기 지역 클럭을 라우팅하기 위해 충분하지 않기 때문에 더 많은 전력을 소비할 수 있다. 그럼에도 불구하고, 저전력 Spartan-6상에서의 비동기 프로세서의 설계는 앞서 정의된 대기 비율(idling-ratio)에 따른 이벤트 구동 동작 환경에서 더 좋은 해결책이 될 수 있다.

[0104]

수학식 11에 기초한 단순화된 1차 분석은, 만약 대기 비율(idling-ratio)이 표 2에서 주어진 값(ΔC_{logic}
 $(2.79 \text{ mW} - 2.02 \text{ mW}) / (V^2 \cdot f) = 0.77 \text{ mW} / (V^2 \cdot f)$, $\Delta C_{signal} = (6.28 \text{ mW} - 5.24 \text{ mW}) / (V^2 \cdot f) = 1.04 \text{ mW} / (V^2 \cdot f)$, $\Delta C_{clock} = (6.78 \text{ mW} - 5.35 \text{ mW}) / (V^2 \cdot f) = 1.43 \text{ mW} / (V^2 \cdot f)$ 및 $C_{sync}^{clock} = 5.35 \text{ mW} / (V^2 \cdot f)$)인 0.38보다 크다면, 비동기 프로세서를 기초로 한

Spartan-6가 동기 프로세서 보다 전력을 덜 소모하는 것을 나타낸다.

[0105]

이벤트 구동 환경에서 버스트 작업 특성은 특정된 시간 동안에는 어플리케이션 코드를 실행하고, 다른 특정된 시간 동안에는 아무것도 하지 않도록 설계된다. 인터럽트 기반의 시스템에서, 어플리케이션 코드는 새로운 데이터에 대한 새로운 인터럽트 신호를 기다리는 동안 버스트 작업 데이터로 인터럽트 처리를 완료한후, 인터럽트 코드가 코어 상에서 버스트 데이터를 처리하고, 그 다음 코어가 대기 상태로 돌아가도록 하기 위한 인터럽트 핸들러로 생각될 수 있다. 인터럽트 처리를 위한 런타임은 동작 시간(working phase time)으로 생각될 수 있고, 새로운 이벤트를 기다리는 대기 시간은 대기 시간(Idle phase time)으로 고려될 수 있다. 결과적으로, 이벤트 구동 환경에서 버스트 작업으로부터의 전력 이득의 평가는 동작 시간(working phase time)과 대기 시간(Idle phase time)에 의해 특정된 비율을 달리함으로써 수행될 수 있다.

[0106]

동작 시간(working phase time)과 대기 시간(Idle phase time)의 영향을 평가하기 위하여, 비율 요소

$$\beta = \frac{f_{idle}}{f_{work} + f_{idle}}$$

을 사용하며, 이러한 결과를 표 3에 나타내었다.

표 3

$\beta = f_{idle}/(f_{work} + f_{idle})$	Sync. MIPS Dynamic Power				Async. MIPS Dynamic Power			
	Logic	Signal	Clock	Total	Logic	Signal	Clock	Total
0 : Fully working	2.02	5.24	5.35	12.61	2.79	6.28	6.78	15.85
0.1	1.98	5.22	5.34	12.54	2.64	5.96	6.08	14.68
0.2	1.73	4.52	5.34	11.59	2.30	5.19	5.41	12.9
0.3	1.48	3.87	5.34	10.69	1.98	4.48	4.73	11.19
0.4	1.25	3.27	5.34	9.86	1.67	3.80	4.06	9.53
0.5	1.03	2.70	5.33	9.06	1.37	3.14	3.38	7.89
0.6	0.81	2.13	5.33	8.27	1.07	2.49	2.71	6.44
0.7	0.61	0.59	5.33	6.99	0.79	1.86	2.03	4.96
0.8	0.4	1.05	5.33	6.32	1.24	0.52	1.36	2.67
0.9	0.19	0.54	5.33	6.06	0.65	0.25	0.68	1.58
1 : Fully idling	0	0	5.33	5.33	0	0	0	0

[0107]

[0108]

표 3에 나타난 바와 같이, 전력 소비는 β 에 매우 선형적이다. 또한, 도 9는 대기 비율(idling-ratio)을 다양하게 변화함에 따른 전력 소비에 대한 그래프를 나타낸다. 전력 소비가 0.4보다 더 크면, 크로싱 포인트(Crossing Point)로 표시되고, 도 9에 따르면, 이 크로싱 포인트에서 값은 0.35와 0.4 사이의 값이 된다. 이에 따라, 본 발명의 실시예에 따른 비동기 프로세서가 낮은 전력 소비를 위해 더 나은 선택이 될 것임을 확인할 수 있다. 표 2에서 주어진 데이터를 식 11을 사용하여 간단하게 분석하여 파생한 값의 범위가 아이들링 비율 0.38을 포함하는 것은 주목할 만하다.

[0109]

전술한 바와 같이, 본 발명의 실시예에 따른 비동기 방식의 프로세서는 대응하는 동기 방식의 프로세서와 비교할 때, 고성능 Virtex-5상에서 정확하게 동작하고, 36.8%의 동적 전력 감소를 얻었다. 저용량 Spartan-6상에서, 25.6%의 전력이 더 소모되었지만, 이벤트 구동 작업 환경의 버스트 동작으로 인해 전력 감소가 이루어질 수 있다.

[0110]

본 발명에 따른 방법은 다양한 컴퓨터 수단을 통하여 관독 가능한 소프트웨어 형태로 구현되어 컴퓨터로 관독 가능한 기록매체에 기록될 수 있다. 여기서, 기록매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 기록매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 예컨대 기록매체는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(Magnetic Media), CD-ROM(Compact Disk Read Only Memory), DVD(Digital Versatile Disk)와 같은 광 기록 매체(Optical Media), 플롭티컬 디스크(Floptical Disk)와 같은 자기-광 매체(Magneto-Optical Media), 및 롬(ROM), 램(RAM, Random Access Memory), 플래시 메모리, SSD(Solid State Disk), HDD(Hard Disk Drive) 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치를 포함한다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드 뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함할 수 있다. 이러한 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수

있으며, 그 역도 마찬가지이다.

[0111]

이상 본 발명을 몇 가지 바람직한 실시예를 사용하여 설명하였으나, 이들 실시예는 예시적인 것이며 한정적인 것이 아니다. 이와 같이, 본 발명이 속하는 기술분야에서 통상의 지식을 지닌 자라면 본 발명의 사상과 첨부된 특허청구범위에 제시된 권리범위에서 벗어나지 않으면서 균등론에 따라 다양한 변화와 수정을 가할 수 있음을 이해할 것이다.

부호의 설명

[0112]

11, 12, 13: AND 게이트

14: OR 게이트

100, 101, 102, 103: 클럭제공부

110, 111, 112: C소자

200, 201, 202, 203: 지연부

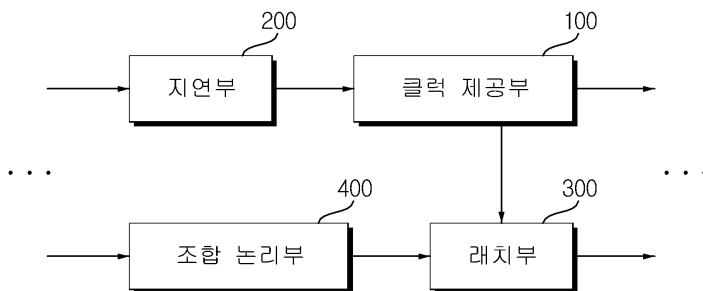
211, 212, 213, 214: 지연 AND 게이트

300, 301, 302, 303: 래치부

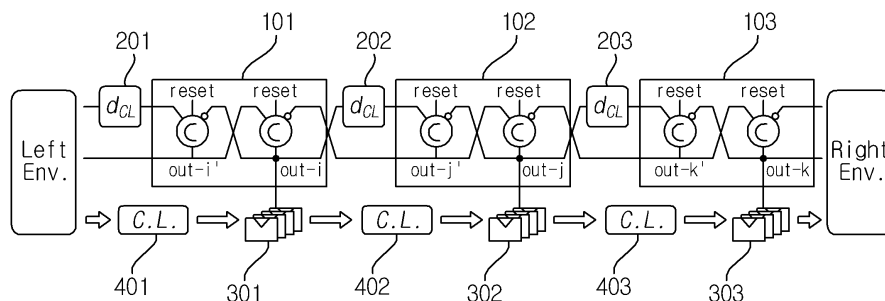
400, 401, 402, 403: 조합논리부

도면

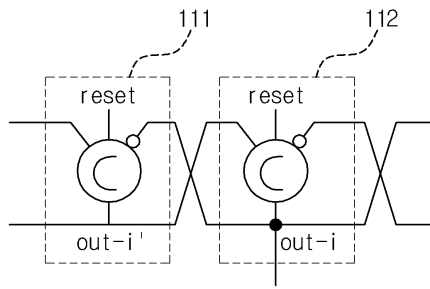
도면1a



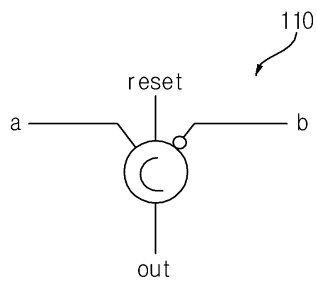
도면1b



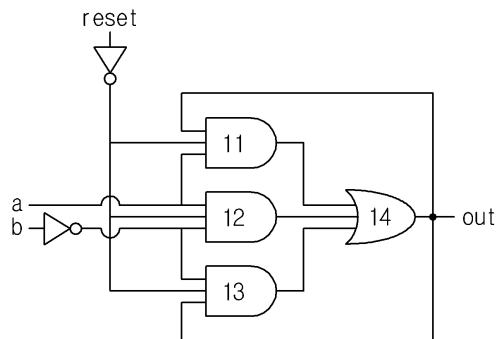
도면2a



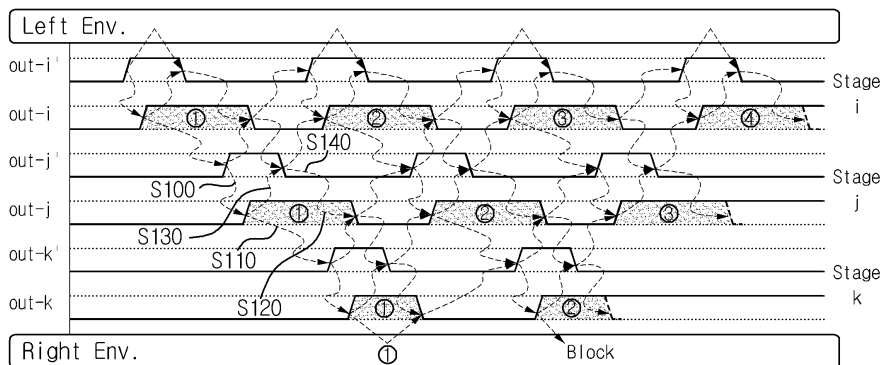
도면2b



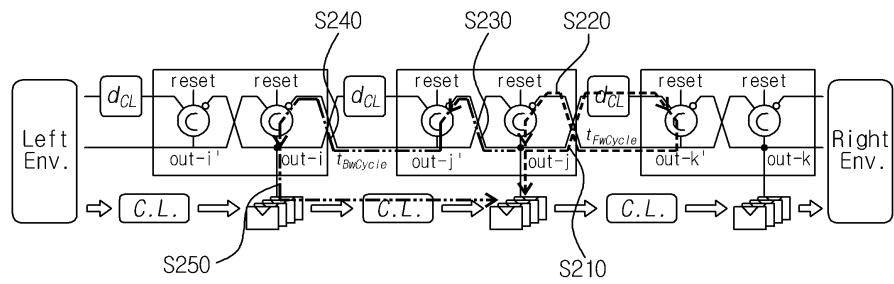
도면2c



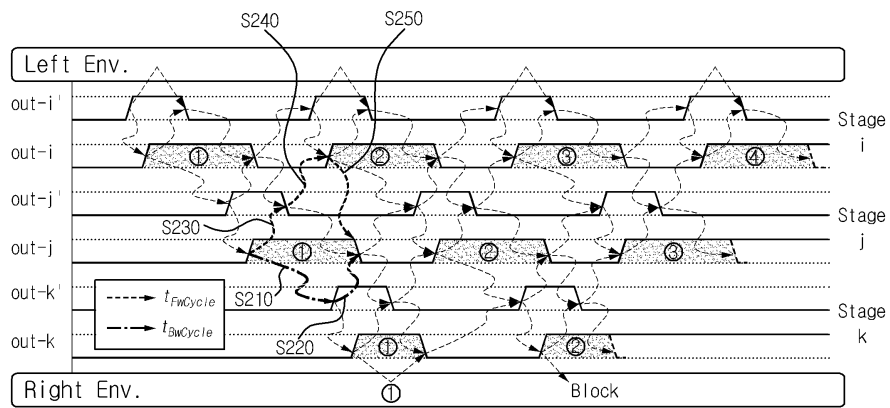
도면3



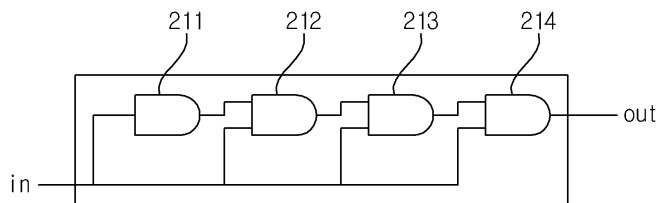
도면4



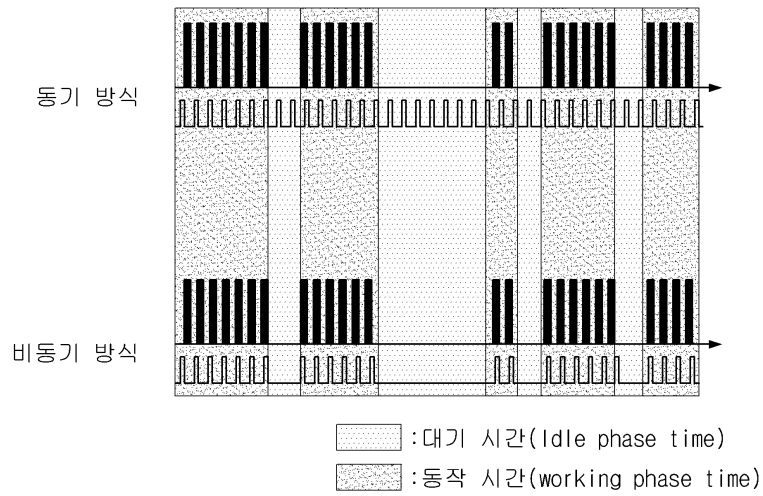
도면5



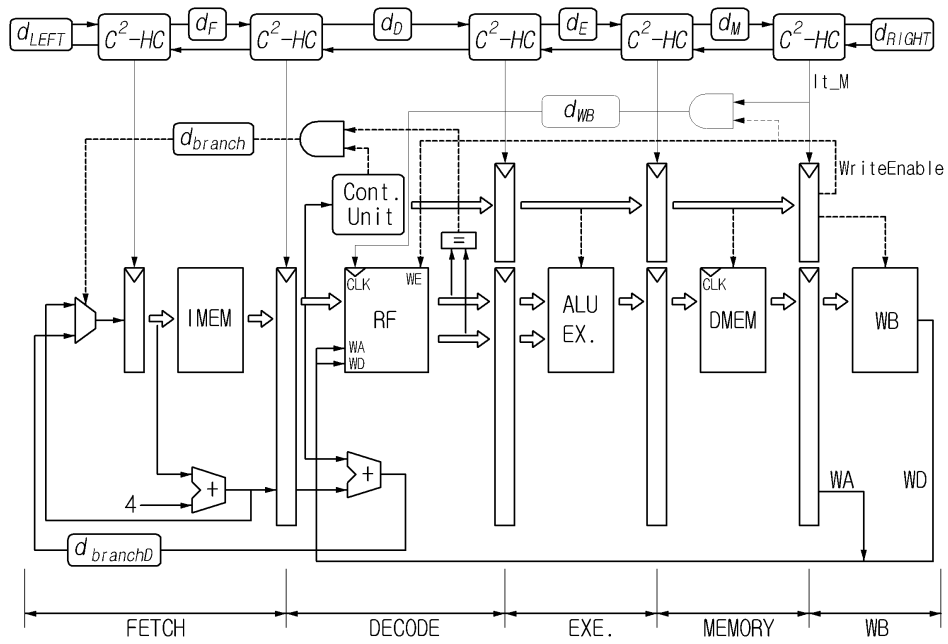
도면6



도면7



도면8



도면9

