

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) Int. Cl. ⁶ G06F 17/50	(45) 공고일자 1999년01월15일	(11) 등록번호 특0165992	(24) 등록일자 1998년09월21일
(21) 출원번호 특 1995-025408	(65) 공개번호 특 1996-008609	(43) 공개일자 1996년03월22일	
(22) 출원일자 1995년08월18일	(30) 우선권주장 94-215215 1994년08월18일 일본(JP)		
(73) 특허권자 닛본덴기 가부시끼가이샤	(72) 발명자 아사키 도시하루		
(74) 대리인 장수길, 구영창			

심사관 : 이은철

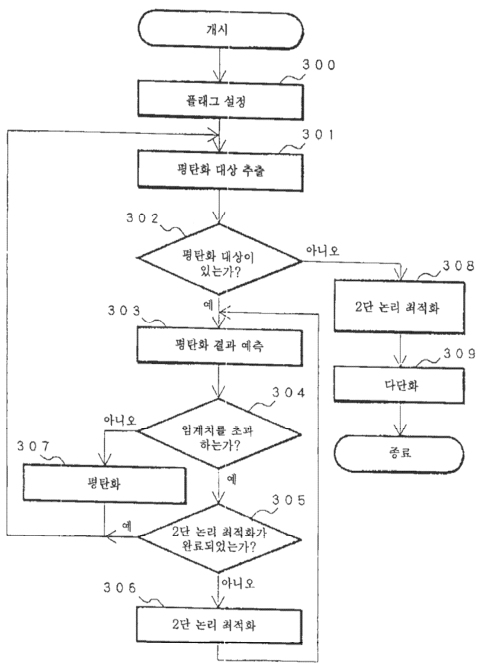
(54) 논리 회로 합성 방법 및 장치

요약

본 발명의 목적은 논리 평탄화 처리에 요하는 메모리 사용량을 삭감, 처리 시간을 단축하고, 대규모 회로에 대해 효율 좋게 최적화하는 논리 회로 합성 방법 및 장치를 제공하는 것이다.

논리 평탄화 처리 대상의 조합 회로를 추출하고(301), 논리 평탄화 처리 대상이 있는 경우, 논리 평탄화 처리 대상에 대해 평탄화 처리 실행 후의 회로 규모를 예측하며(302), 예측한 회로 규모가 미리 정한 임계치를 초과한 경우는 논리 평탄화 처리 대상이 2단 논리 최적화 완료인 경우 처리(301)로 복귀하고, 2단 논리 최적화 완료가 아니면, 논리 평탄화 처리 대상에 대해 2단 논리 최적화를 실행하여(306), 평탄화 결과를 다시 예측한다. 임계치를 초과하지 않는 경우, 논리 평탄화 처리 대상에 대해 논리 평탄화 처리를 실행하고(307), 처리(301)로 복귀한다.

대표도



명세서

[발명의 명칭]
논리 회로 합성 방법 및 장치

[도면의 간단한 설명]
제1도는 본 발명의 한 실시예를 도시한 흐름도.

또한, 다단화(「논리 다단화」라고도 함)란, 예를 들어 다음 식(3)의 논리식에 대해, 항 $a \cdot b \cdot c$, $a \cdot b \cdot d$ 에는 $a \cdot b$ 라고 하는 공통 부분이 있기 때문에, 공통 부분을 추출해 새롭게 중간 변수 $e = a \cdot b$ 를 도입하여 논리식 f 를 식(4)와 같이 다단화(즉, 다단 회로로 변환)하는 것을 말한다.

$$f = a \cdot b \cdot c + a \cdot b \cdot d \quad \dots (3)$$

$$e = a \cdot b$$

$$f = e \cdot c + e \cdot d \quad \dots (4)$$

테크놀로지 매핑이란, 논리 합성 툴의 기능중 하나로, 게이트 레벨의 논리를 특정한 반도체 기술에 의존한 설계 데이터(넷리스트 ; 접속 정보)로 변환하는 것을 말한다. 테크놀로지 매핑에 의해, 예를 들면 하나의 ATL(레지스터 트랜스퍼 레벨) 기술(記述)로부터 예를 들면 서로 다른 반도체 메이커 등의 게이트 어레이 또는 셀베이스 LSI 등의 설계 데이터를 작성할 수 있다.

그런데, VHDL(VHSIC Hardware Description Language) 등의 하드웨어 설명 언어로 기술된 회로 등에서는, 설계자가 기술한 논리 구조를 가능한 한 남겨두는 편이 효율 좋게 최적화 할 수 있기 때문에, 조합 회로를 전부 평탄화하지 않고 부분적으로 평탄화해 2단 논리 최적화를 행하여, 이것을 다단화해서 테크놀로지 매핑하는 방법이 있다.

지연을 목적 함수로 하여 논리 회로를 최적화하는 경우, 일반적으로 논리 단수(段數)를 적게 하면 최대 지연값을 삭감할 수 있다는 것이 공지되어 있다.

즉, 크리티컬 패스(critical path) 상의 조합 회로를 평탄화함으로써, 논리 단수를 적게 하여 지연을 삭감한다.

조합 회로 부분을 전부 평탄화하면, 일반적으로 지연은 단축되지만, 면적이 증대한다. 그래서, 면적 증가를 억제하면서 지연을 최적화하기 위해, 부분 회로의 평탄화를 행한다. 그리고, 평탄화한 회로는 필요에 따라 다단화하여 테크놀로지 매핑한다.

도면을 참조하여 종래의 평탄화 처리에 대해 설명한다.

제5도와 같이, 조합 회로(501)(논리 함수는 x)가 조합 회로(502)(논리 함수는 f)에서 팬 아웃(fan out)하고 있는 것으로 한다. 이들 논리 함수 f , x 에 관하여 일반적으로 섀넌(Shannon) 전개라고 불리는 아래의 식(5)가 성립한다는 것이 공지되어 있다.

$$f = x f_x + \overline{x} f_{\overline{x}} \quad \dots (5)$$

여기에서, \overline{x} 는 x 의 부정 논리(반전), f_x 는 논리 함수 f 의 x 에 대한 여인수(cofactor)라고 불리우고, 논리 함수 f 의 입력 변수 x 를 1로 했을 때의 논리 함수를 나타낸다. $\overline{f_x}$ 는, 마찬가지로 논리 함수 f 의 입력 변수 x 를 0으로 했을 때의 논리 함수를 나타낸다. 상기 식(5)를 이용하여 평탄화 처리를 행한다.

평탄화 처리는 제6도에 도시한 바와 같이 수순으로 행하여진다. 제5도에 도시한 바와 같은 접속 관계에 있는 2개의 논리 함수 f, x 에 대해 평탄화를 행하는 경우, 먼저 x 의 논리 부정 \overline{x} 를 취한다(스텝 602).

다음에, 논리 함수 f 의 x 에 대한 여인수 f_x 와 x 의 논리곱을 취하여 $f1$ 으로 한다(스텝 603).

이어서, f 의 \overline{x} (반전)에 대한 여인수와 \overline{x} 의 논리곱을 취하여 $f0$ 이라고 한다(스텝 604).

그리고, $f1$ 과 $f0$ 의 논리합을 취하면, 최종적으로 f 에 대해 x 를 평탄화한 것이 된다(스텝 605).

조합 회로(502)의 모든 팬 인(fan in)에 대해 평탄화 할 경우, 스텝 601로 복귀하여 스텝 601 ~ 605의 처리를 반복한다.

상기 평탄화 처리에 있어서, 논리 부정, 논리곱을 취하는 처리는 대상으로 되는 조합 회로의 곱항수(積項數)를 n 이라 하면, 처리의 계산 복잡도는 n 의 2승에 비례하고, 이 때문에 대규모의 회로에 대해서는 사용하는 계산기 상의 메모리를 대량으로 필요로 하여 처리 시간도 증가하게 된다.

이와 같이, 논리 합성 기술(技術)에 있어서, 평탄화 처리는 매우 중요하고, 특히 대규모의 회로를 효율 좋게 최적화할 때, 평탄화에 필요한 메모리의 삭감, 처리시간 단축의 달성이 과제로 되고 있다.

제7도를 참조하여, 면적 최적화의 한 예를 사용하여 설명한다. 먼저, 조합회로 부분의 평탄화를 행하고(스텝 701), 평탄화된 2단 논리 회로를 논리 압축함으로써 2단 논리 최적화하며(스텝 702), 공통 부분 등

을 추출하면서 다단화하고(스텝 703), 타겟으로의 테크놀로지 매핑을 행한다(스텝 704).

평탄화에 관한 이러한 종류의 논리 회로 합성 방법으로서, 예를 들면 특개평 4-352283호 공보에 개시된 것이 있다. 이 공보에는 논리식 중에 존재하는 중간 변수 모두를 전개하지 않고, 중간 변수의 전개 처리에 앞서 전개 후의 곱항수를 조사하여, 곱항수가 소정값을 넘지 않을 경우에만 중간 변수의 전개 처리를 행하는 것으로, 중간 변수의 처리 전개수를 저감해 처리 시간을 단축하는 다단 조합 회로의 합성 방법이 제안되어 있다. 또, 특개평 4-352283호 공보 중의 전개 처리는 평탄화 처리와 같은 뜻이다.

제8도를 참조하여 상기 특개평 4-352283호 공보에 제안된 평탄화 처리를 설명한다. 또, 제8도는 제7도의 평탄화 처리(701)의 상세한 흐름도에 대응하는 것이다.

평탄화 처리 대상을 추출하고(스텝 801), 중간 변수, 즉 평탄화해야 할 조합회로가 남아 있는지의 여부를 판정하며(스텝 802), 평탄화 후의 곱항수를 예측하고(스텝 803), 예측한 곱항수가 미리 정한 임계치(「허용 최대치」라고도 함)를 초과하고 있을 경우는 평탄화하지 않고 스텝 801로 복귀하고, 예측한 곱항수가 임계치를 초과하고 있지 않을 경우에만 평탄화 처리를 행하여(스텝 805) 스텝 801로 복귀한다.

스텝 802의 판정 처리에 있어서 모든 평탄화 대상의 처리가 종료하면 2단 논리 최적화하여(스텝 806), 다단화한다(스텝 807).

이와 같이, 제8도에 도시된 종래의 평탄화 처리에서는 중간 변수 중의 곱항수를 미리 정해진 임계치와 비교하여, 예측한 곱항수가 임계치를 초과하지 않을 경우에만 평탄화 처리를 행하여 처리 시간의 단축화를 도모하는 것이지만, 곱항수가 많을 경우에는 충분한 평탄화 처리가 행해지지 않고, 이 때문에 평탄화 후의 회로 규모의 축소가 달성되지 않아 면적의 최적화를 충분히 행할 수 없다는 문제가 있다. 또한, 종래의 평탄화 처리에서는 대규모 회로의 최적화에 즈음하여 메모리 용량의 삭감, 처리 시간의 대폭적인 단축화가 도모되지 않는다는 문제가 있다.

따라서, 본 발명은 상기 문제를 해소하고, 특히 대규모의 회로를 최적화할 때에, 평탄화에 필요한 메모리의 삭감 및 처리 시간의 대폭적인 단축화를 달성하고, 대규모 회로에 대해 효율 좋게 최적화하는 논리 회로 합성 장치 및 논리 회로 합성 방법을 제공하는 것을 목적으로 한다.

상기 목적을 달성하기 위해 본 발명은, 논리 평탄화 처리 대상의 회로에 대해 평탄화 처리 후의 회로 규모를 예측하고, 이 회로 규모의 예측값이 미리 정한 임계치를 초과할 때에 이 회로에 2단 논리 최적화를 실행하며, 이 2단 논리 최적화를 행한 회로에 대해 평탄화 처리 후의 회로 규모를 다시 예측하고, 상기 회로 규모의 예측값이 상기 임계치를 초과하지 않는에 대해 논리 평탄화 처리를 행하도록 한 것을 특징으로 하는 논리 회로 합성 방법을 제공한다.

본 발명의 논리 회로 합성 방법에 있어서는, 논리 회로의 최적화 처리는 바람직하게는 (A) 논리 평탄화 처리 대상 회로를 추출하고, (B) 논리 평탄화 처리 대상 회로에 대해 평탄화 처리 후의 회로 규모를 예측하며, (C) 이 회로 규모의 예측치가 미리 정한 임계치를 초과하는지의 여부를 판정하고, (D) 이 회로 규모의 예측치가 상기 미리 정한 임계치를 초과하는 경우에 있어서, 상기 논리 평탄화 처리 대상 회로가 2단 논리 최적화를 완료하지 않을 경우에는 2단 논리 최적화를 행한 후에 상기 공정 (B)에서 2단 논리 최적화 완료의 회로에 대해서 회로 규모를 다시 예측하며, (E) 상기 회로 규모의 예측값이 상기 미리 정한 임계치를 초과할 경우에 있어서, 상기 논리 평탄화 처리 대상 회로가 2단 논리 최적화를 완료했을 경우에는 상기 공정 (A)에서 논리 평탄화 처리 대상 회로를 추출하고, (F) 상기 회로 규모의 예측값이 상기 미리 정한 임계치를 초과하지 않을 경우에는 상기 논리 평탄화 처리 대상 회로에 대해 평탄화 처리를 행하며, (G) 논리 평탄화 처리 대상 회로가 존재하지 않을 경우에는 2단 논리 최적화를 행하고, 또한 (H) 논리 다단화를 행하는, 상기 각 공정을 포함하는 것을 특징으로 한다.

본 발명의 논리 회로 합성 방법에 있어서는 상기 회로 규모로서 곱항수 또는 팬인이 이용되거나, 또는 곱항수 및 팬인이 이용된다.

본 발명의 논리 회로 합성 방법에 있어서는 중간 변수에 2단 논리 최적화를 행했는지 여부의 플래그(flag)를 갖추고, 초기화시에 이 플래그를 2단 논리 최적화 처리 미실행 상태로 리셋하는 것을 특징으로 하고 있다.

또한, 본 발명은 논리 회로의 테크놀로지 매핑에 이용하는 테크놀로지 라이브러리(library) 블록의 정보를 입력하는 라이브러리 입력부와, 논리 합성 대상으로 되는 논리 회로 기술(記述)을 입력하는 회로 입력부와, 지연 또는 면적 등으로 이루어지는 논리 회로의 제약 조건을 입력하는 제약 조건 입력부와, 논리 회로의 최적화 처리를 행하는 논리 최적화부에서, 논리 평탄화의 처리 중에 2단 논리 최적화를 행하여 논리 최적화를 행하는 논리 최적화부와, 주어진 제약 조건을 목적 함수로 하여 상기 테크놀로지 라이브러리 블록을 논리 회로에 할당하는 테크놀로지 매핑부와, 처리 결과의 회로를 출력하는 회로 출력 처리부를 갖는 것을 특징으로 하는 논리 회로 합성 장치를 제공한다.

본 발명의 논리 회로 합성 장치에 있어서는, 논리 최적화부가 논리 평탄화 처리 대상의 회로에 대해 평탄화 처리 후의 회로 규모를 예측하고, 이 회로 규모의 예측값이 미리 정한 임계치를 초과할 때에 이 회로에 2단 논리 최적화를 행하고, 이 2단 논리 최적화를 행한 회로에 대해 평탄화 처리 후의 회로 규모를 다시 예측하며, 상기 회로 규모의 예측치가 상기 임계치를 초과하지 않은 것에 대해 논리 평탄화 처리를 행하도록 구성된 것을 특징으로 하는 것이다.

상술한 종래의 논리 회로 합성 방법에 대해, 본 발명은 논리 최적화 처리에 있어서, 평탄화 후의 회로 규모를 예측하고, 예측된 회로 규모가 미리 정해진 임계치를 초과하는 경우에는 평탄화 처리 대상의 조합 회로에 대해 2단 논리 최적화를 실행함으로써, 곱항수를 충분히 적게 함으로써 평탄화 후의 예측 회로 규모를 축소할 수 있다는 이점을 갖는다. 또한, 본 발명은 평탄화 처리의 처리 효율(메모리 용량, 처리 시간)을 대폭 향상시키고 있다.

도면을 참조하여 본 발명의 실시예를 이하에 설명한다.

[실시예 1]

제2도는 본 발명의 한 실시예에 관한 논리 회로 합성 장치의 하드웨어 구성예를 도시한 것이다.

제2도에 도시한 바와 같이, 본 실시예는 논리 회로의 테크놀로지 매핑에 이용하는 테크놀로지 라이브러리 블록의 정보를 입력하는 라이브러리 입력부(200)과, 논리 합성 대상으로 되는 논리 회로 기술(記述)을 입력하는 회로 입력부(201)과, 지연이나 면적 등의 논리 회로의 제약 조건을 입력하는 제약 조건 입력부(202)와, 논리의 최적화를 행하는 논리 최적화부(203)과, 주어진 제약 조건을 만족시키는 것을 목적 함수로 하여 테크놀로지 라이브러리 블록을 논리 회로에 할당하는 테크놀로지 매핑부(204)와, 처리 결과의 회로를 출력하는 회로 출력부(205)와, 테크놀로지 라이브러리와 논리 회로와 제약 조건을 유지하는 기억부(206)으로 구성된다.

다음에, 제1도 및 제2도를 참조하여 본 실시예의 동작을 설명한다. 제1도는 본 실시예의 처리 흐름을 도시한 흐름도이다.

라이브러리 입력부(200)에서, 테크놀로지 라이브러리 블록의 논리, 팬 아웃 제약 등의 정보가 설명된 테크놀로지 라이브러리(100)를 입력하고, 이것을 기억부(206)에 유지한다(스텝 101의 라이브러리 입력 처리).

다음에, 회로 입력부(201)에서 논리 회로 기술(102)을 입력하고, 기억부(206)에 유지한다(스텝 103의 회로 입력 처리).

다음에, 제약 조건 입력부(202)에 있어서, 제약 조건(104)을 입력하고, 기억부(206)에 유지한다(스텝 105의 제약 입력 처리).

논리 최적화부(203)에 있어서, 기억부(206)에 유지되어 있는 논리 회로에 대해 논리의 최적화를 행한다(스텝 106의 논리 최적화 처리).

테크놀로지 매핑부(204)에 있어서, 기억부(206)에 보유되어 있는 논리 회로에 대해 기억부(206)에 유지되어 있는 제약 조건을 만족시키는 것을 목적 함수로 하여 기억부(206)에 유지되어 있는 테크놀로지 라이브러리 블록을 할당한다(스텝 107의 테크놀로지 매핑 처리).

회로 출력부(205)에 있어서, 기억부(206)에 유지되어 있는 논리 회로를 출력하여 논리회로 설명(109)이 얻어진다(스텝 108의 회로 출력 처리).

다음에, 제3도를 참조하여 제1도의 논리 최적화 처리(106)에 있어서의 논리 평탄화 처리의 상세를 설명한다.

제3도는 논리 최적화 처리(106)의 상세한 흐름도의 한 예이다.

기억부(206)에 유지되어 있는 논리 회로의 조합 회로 부분에 대해 논리 최적화 처리를 실행한다.

모든 중간 변수에 2단 논리 최적화 처리 미실행 플래그를 설정한다(스텝 300).

논리 평탄화 처리 대상 회로, 즉 중간 변수에 대응하는 조합 회로와 그 팬 아웃측의 조합 회로를 추출한다(스텝 301). 논리 평탄화 대상의 회로가 논리 합성 대상 회로의 부분 회로인 경우에는 그 부분 회로 내에서부터 논리 평탄화 대상을 추출한다. 또한, 중간 변수가 많은 팬 아웃측을 갖고 있어서, 평탄화를 행하지 않는 편이 논리 최적화에 효과적이라고 판단되는 경우에는 평탄화 처리 대상으로서 추출하지 않는다.

논리 평탄화 처리 대상 회로가 있는지의 여부를 판정하고(스텝 302), 논리 평탄화 처리 대상 회로가 있을 경우에는 스텝 303으로 진행한다. 논리 평탄화 처리 대상 회로가 없을 경우에는 스텝 308로 진행한다.

논리 평탄화 처리 대상 회로에 대해 논리 평탄화 처리 실행 후의 회로 규모를 예측한다(스텝 303).

스텝 303에 의해 예측된 회로 규모가 미리 정해진 회로 규모의 임계치를 초과하는지의 여부를 판정하고(스텝 304), 논리 평탄화 처리 실행 후의 회로 규모의 예측값이 임계치를 초과하지 않을 경우에는 스텝 305로 진행한다. 논리 평탄화 처리 실행 후의 회로 규모의 예측치가 임계치를 초과하지 않을 경우에는 평탄화 처리를 행한다(스텝 307).

논리 평탄화 처리 실행 후의 회로 규모의 예측치가 임계치를 초과했을 경우에는 논리 평탄화 처리 대상 회로가 2단 논리 최적화 처리 미실행 플래그가 해제되어 있는지의 여부를 판정하고(스텝 305), 2단 논리 최적화 처리 미실행 플래그가 해제되어 있는 경우에는 2단 논리 최적화 완료이기 때문에 그대로 스텝 301로 복귀한다.

2단 논리 최적화 처리 미실행 플래그가 설정되어 있는 경우에는 논리 평탄화 처리 대상 회로에 대해 2단 논리 최적화를 실행하고, 2단 논리 최적화 처리 미실행 플래그를 해제하고(스텝 306), 스텝 303으로 복귀한다. 이 경우, 스텝 303에서는 2단 논리 최적화가 행해진 회로에 대해 논리 평탄화 처리 실행 후의 회로 규모를 예측한다.

논리 평탄화 처리 실행 후의 회로 규모의 예측값을 임계치를 초과하지 않은 경우에는 논리 평탄화 처리 대상 회로에 대해 논리 평탄화 처리를 실행하고, 논리 평탄화 처리 대상 회로의 팬 아웃측의 회로의 논리 최적화 처리 미실행 플래그를 재설정하고(스텝 307), 스텝 301로 복귀한다.

스텝 308에서 2단 논리 최적화 처리 미실행 플래그가 설정되어 있는 중간 변수에 대해 2단 논리 최적화 처리를 행하고, 스텝 309에서 필요에 따라 논리 다단화 처리를 행한다.

다음에, 제 4도의 논리식을 참조하여 본 실시예의 논리 최적화 처리(106)에 있어서의 논리 평탄화 처리(제3도 참조)의 적용예를 설명한다.

여기서는 회로 규모의 척도로서 조합 회로의 곱항수를 이용한다.

스텝 300에서 모든 중간 변수에 2단 논리 최적화 처리 미실행 플래그를 설정한다.

스텝 301에서 조합 논리 회로 중의 중간 변수 $x(401)$ 과 팬 아웃측 $f(402)$ 가 논리 평탄화 처리 대상으로서 추출된다고 하자.

여기서, $i1, i2, i3, i4, i5$ 는 조합 논리 회로 중의 중간 변수이어도 좋고, 논리 회로의 외부 입력 단자 이어도 좋고, 플립플롭 등의 조합 논리 소자 이외의 회로 소자의 출력 단자이어도 좋다. 또한, 중간 변수 x 는 f 이외의 팬 아웃측이어도 좋다. 또한, 제4도 중, 연산 기호 $*$ 는 논리곱, $+$ 는 논리합, $-$ 는 논리 부정(반전)을 각각 나타낸다.

스텝 302에서는 논리 평탄화 처리 대상 회로 f 와 x 가 추출되어 있기 때문에, 스텝 303으로 진행한다.

스텝 303에서는 미리 정해진 임계치와 비교하기 때문에 논리 평탄화 처리 결과의 회로 규모를 예측한다. 상기와 같이 회로 규모로서 곱항수를 이용하는 것으로 한다. 논리 평탄화 처리 결과의 곱항수는 다음식(6)에 따라 구해진다.

[수]

$$\text{곱항수} = (\underline{f\text{중의 } x\text{를 포함하는 곱항수}}) * (\underline{x\text{의 곱항수}})$$

$$+ (\overline{f\text{중의 } x\text{를 포함하는 곱항수}}) * (\overline{x\text{의 곱항수}})$$

$$+ (\underline{f\text{중의 } x, \overline{x}\text{를 포함하지 않는 곱항수}}) \cdot \cdot \cdot (6)$$

이 경우, f 중의 x 를 포함하는 곱항수는 2, x 의 곱항수는 2, f 중의 \overline{x} 를 포함하는 곱항수는 2, \overline{x} 의 곱항수는 2, f 중의 x, \overline{x} 를 포함하지 않는 곱항수는 1이기 때문에 상기 식(6)에 의해,

$$2*2+2*2+1=9$$

이 되고, 논리 평탄화 후의 곱항수는 9로 예측된다.

스텝 304에 의해, 곱항수 9가 미리 정해진 임계치를 초과하지 않는다고 판단된 경우에는 스텝 307로 진행하고, 스텝 304에 의해 곱항수 9가 미리 정해진 임계치를 초과하고 있다고 판단된 경우에는 스텝 305로 진행한다.

스텝 305에 의해, 논리 평탄화 처리 대상 회로 f, x 함께 2단 논리 최적화 처리 미실행 플래그가 해제되어 있을 경우에는 f, x 함께 2단 논리 최적화 완료라고 판단되어 스텝 301로 복귀하고, 새로운 논리 평탄화 처리 대상 회로를 추출한다.

논리 평탄화 처리 대상 회로 f, x 중 어느 하나에 2단 논리 최적화 처리 미실행 플래그가 설정되어 있을 경우에는 스텝 306으로 진행한다.

스텝 306에서는 논리 평탄화 처리 대상 회로 f, x 의 2단 논리 최적화 처리 미실행 플래그가 설정되어 있는 것에 대해 2단 논리 최적화를 실행한다. 또, 2단 논리 최적화 처리는, 쿠아인 맥클러스키(Quine McCluskey)법 등으로 대표되는 논리의 용장성을 제거하는 논리 최적화 처리이고, 공지된 기술이며, 상기 용어 설명의 내용과 실질적으로 동일하다.

2단 논리 최적화의 결과, 논리 평탄화 처리 대상 회로의 f, x 는 제4도의 403, 404와 같이 된다.

2단 논리 최적화 후, f, x 의 2단 논리 최적화 처리 미실행 플래그를 해제하고 스텝 303으로 복귀한다.

스텝 303에 의해 논리 평탄화 후의 곱항수를 예측한다. 논리 평탄화 처리 대상 회로 f 중의 x 를 포함하는 곱항수는 1, x 의 곱항수는 1, f 중의 \overline{x} 를 포함하는 곱항수는 1, \overline{x} 의 곱항수는 1, f 중의 x, \overline{x} 를 포함하지 않는 곱항수는 1이므로, 상기식(6)에 의해,

$$1*1+1*1=3$$

이 되고, 곱항수는 3으로 예측된다.

곱항수 3이 미리 정해진 임계치를 초과하지 않는다고 판단된 경우에는 스텝 307로 진행하지만, 임계치를 초과하고 있다고 판단될 경우에는 스텝 305로 진행하고, f, x 함께 2단 논리 최적화 완료이기 때문에 스텝 301로 복귀한다.

스텝 307에서는 논리 평탄화 처리를 행하고, f 의 2단 논리 최적화 처리 미실행 플래그를 설정한다. x 는 논리 평탄화 처리에 의한 논리 변경은 없기 때문에 2단 논리 최적화 처리 미실행 플래그의 변경은 행할 수 없다. 논리 평탄화의 결과, 405의 논리가 얻어진다.

상기 설명에서는 조합 회로의 논리 표현에 논리식을 이용했지만, 진리값표, BDD(Binary Decision

Diagram) 등을 이용해도 좋다.

논리 최적화 전의 곱항수가 1736인 회로에 대해, 본 실시예를 적용하지 않는 종래예에서는 필요 메모리는 프로세스 사이즈로 6552k byte(킬로 바이트), CPU 시간이 78.6초인데 대해 본 실시예를 적용했을 경우, 필요 메모리는 프로세스 사이즈로 6424k byte(약 2% 감소), CPU 시간이 16.6초(약 80% 감소)라는 결과를 얻었다

[실시예 2]

본 발명의 제2실시예를 이하에 설명한다. 본 실시예에서는 회로 규모로서 논리 평탄화 후의 예측 곱항수를 이용하는 대신, 논리 평탄화 후의 예측 팬인을 이용하는 것으로 한다. 논리 평탄화 후의 팬인은 다음 식(7)에 따라 구한다.

[수 2]

$$\text{팬인} = (\text{f의 팬인}) + (\text{x의 팬인})$$

$$- (\text{f와 x에 공통인 팬인}) - 1 \cdot \cdot \cdot (7)$$

제4도의 예에서는 f의 팬인은 4, x의 팬인은 2, f와 x에 공통인 팬인은 0이기 때문에, 상기 식(7)에서

$$2+4-0=5$$

로 되고, 논리 평탄화 후의 팬인은 5로 예측된다.

본 실시예에 있어서는 논리 최적화 처리에 있어서, 회로 규모의 척도로서 곱항수 대신 팬인을 이용하고 있다는 점만이 상기 제1실시예와 상위(相違)하고, 그 이외의 구성은 상기 제1실시예와 동일하다. 보다 상세하게는, 제3도를 참조하여, 본 실시예에서는 스텝 303의 평탄화 결과 예측 처리에 있어서 상기 식(7)에 의해 논리 평탄화 후의 예측 팬인을 산출하고, 스텝 304에서 예측치를 미리 정해진 팬인의 임계치와 비교하는 점이 상기 제1실시예와 상위하다. 그 이외는 상기 제1실시예와 동일하기 때문에, 설명을 생략한다.

[실시예 3]

또한, 본 발명의 다른 실시예로서 회로 규모의 예측값으로서 논리 평탄화 후의 곱항수와 팬인의 양자를 조합해 이용해도 좋다.

[발명의 효과]

이상 설한 바와 같이, 종래의 논리 회로 합성 방법에 있어서의 논리 최적화 처리에서는 메모리 제약이나 실행 시간 제약 등에 의해 충분한 논리 평탄화 처리가 행해지지 않았던 것에 대해, 본 발명의 논리 회로 합성 방법에 의하면, 논리 평탄화 처리 대상에 대해 2단 논리 최적화 처리를 적용함으로써, 이제까지 불가능했던 부분의 논리 평탄화 처리의 실현을 가능하게 함으로써, 논리 최적화의 효율을 대폭 향상시킨다는 효과가 있다.

논리 최적화 전의 곱항수가 1736인 회로에 대해, 본 발명을 적용하지 않은 논리 최적화에서는 필요 메모리 용량은 프로세스 사이즈로 6552k byte, CPU 시간은 78.6초인데 대해, 본 발명의 논리 최적화 처리에 의하면, 필요 메모리 용량은 프로세스 사이즈로 6424k byte(약 2% 감소), CPU 시간은 16.6초(약 80% 감소)로 되고, 논리 최적화 처리의 처리 시간을 약 80%나 삭감하고 있다.

그리고, 본 발명에 있어서는 회로 규모의 예측 수로서, 곱항수, 또는 팬인, 또는 양자를 이용함으로써도 상기 효과를 마찬가지로 달성하는 것이다.

또한, 본 발명의 논리 회로 합성 장치는 논리 평탄화 처리 대상에 대해 2단 논리 최적화 처리를 적용함으로써, 이제까지 불가능했던 부분의 논리 평탄화 처리의 실현을 가능하게 함으로써, 평탄화 처리에 요하는 메모리 용량의 삭감, 처리 시간의 대폭적인 단축을 달성하고, 대규모 회로의 논리 합성을 처리 효율을 특별하게 향상하는 것이다.

(57) 청구의 범위

청구항 1

논리 회로 합성 방법에 있어서, 논리 회로의 최적화 처리에 있어서, 논리 평탄화 처리 대상의 회로에 대해 평탄화 처리 후의 회로 규모를 예측하고, 이 회로 규모의 예측값이 미리 정한 임계치를 초과할 때에, 이 회로에 2단 논리 최적화를 실행하며, 이 2단 논리 최적화를 행한 회로에 대해 평탄화 처리 후의 회로 규모를 다시 예측하고, 상기 회로 규모의 예측치가 상기 임계치를 초과하지 않는 것에 대해 논리 평탄화 처리를 행하도록 한 것을 특징으로 하는 논리 회로 합성 방법.

청구항 2

논리 회로 합성방법에 있어서, 논리 회로의 최적화 처리가, (A) 논리 평탄화 처리 대상 회로를 추출하는 공정, (B) 논리 평탄화 처리 대상 회로에 대해 평탄화 처리 후의 회로 규모를 예측하는 공정, (C) 상기 회로 규모의 예측값이 미리 정한 임계치를 초과하는지의 여부를 판정하는 공정, (D) 상기 회로 규모의

예측값이 상기 미리 정한 임계치를 초과하는 경우에 있어서, 상기 논리 평탄화 처리 대상 회로가 2단 논리 최적화 완료가 아닌 경우에는 2단 논리 최적화를 행한 후에 상기 공정 (B)에서 2단 논리 최적화 완료의 회로에 대해서 회로 규모를 다시 예측하는 공정, (E) 상기 회로 규모의 예측값이 상기 미리 정한 임계치를 초과하는 경우에 있어서, 상기 논리 평탄화 처리 대상 회로가 2단 논리 최적화 완료인 경우에는 상기 공정 (A)에서 논리 평탄화 처리 대상 회로를 추출하는 공정, (F) 상기 회로 규모의 예측치가 상기 미리 정한 임계치를 초과하지 않는 경우에는 상기 논리 평탄화 처리 대상 회로에 대해 평탄화 처리를 행하는 공정, (G) 논리 평탄화 처리 대상 회로가 존재하지 않을 경우에는 2단 논리 최적화를 행하는 공정, 및 (H) 논리 다단화를 행하는 공정을 포함하는 것을 특징으로 하는 논리 회로 합성 방법.

청구항 3

제1항에 있어서, 상기 회로 규모의 척도로서 곱항수(積項數) 및 /또는 팬인(fan in)을 이용하는 것을 특징으로 하는 논리 회로 합성 방법.

청구항 4

제2항에 있어서, 중간 변수에 2단 논리 최적화를 행했는지 여부의 플래그를 갖추고, 초기화시에 이 플래그를 2단 논리 최적화 처리 미실행 상태로 리셋하는 것을 특징으로 하는 논리 회로 합성 방법.

청구항 5

논리 회로 합성 장치에 있어서, 논리 회로의 테크놀로지 매핑(technology mapping)에 이용하는 테크놀로지 라이브러리(library) 블록의 정보를 입력하는 라이브러리 입력부와, 논리 합성 대상이 되는 논리 회로 기술(記述)을 입력하는 회로 입력부와, 지연 또는 면적 등으로 이루어지는 논리 회로의 제약 조건을 입력하는 제약 조건 입력부와, 논리 회로의 최적화 처리를 행하는 논리 최적화부로서, 논리 평탄화의 처리중에 2단 논리 최적화를 행하여 논리 최적화를 행하는 논리 최적화부와, 주어진 제약 조건을 목적 함수로 하여 상기 테크놀로지 라이브러리 블록을 논리 회로에 할당하는 테크놀로지 매핑부와, 처리 결과의 회로를 출력하는 회로 출력 처리부를 포함하는 것을 특징으로 하는 논리 회로 합성 장치.

청구항 6

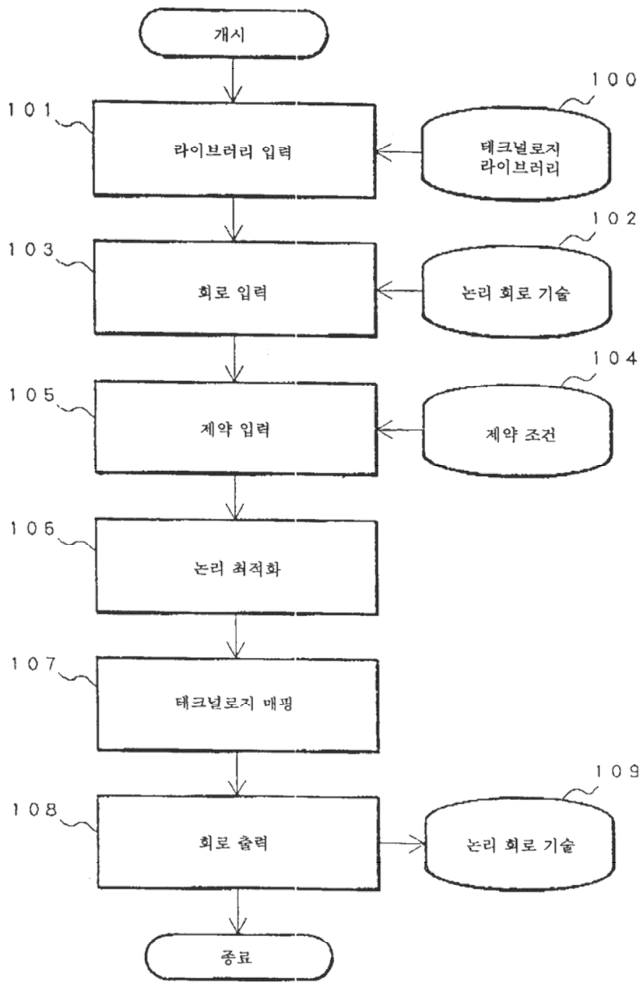
제5항에 있어서, 상기 논리 최적화부가 논리 평탄화 처리 대상의 회로에 대해 평탄화 처리 후의 회로 규모를 예측하고, 이 회로 규모의 예측치가 미리 정한 임계치를 초과할 때에, 이 회로에 2단 논리 최적화를 행하며, 이 2단 논리 최적화를 행한 회로에 대해 평탄화 처리 후의 회로 규모를 다시 예측하고, 상기 회로 규모의 예측치가 상기 임계치를 초과하지 않는 것에 대해 논리 평탄화 처리를 행하도록 구성된 것을 특징으로 하는 논리 회로 합성 장치.

청구항 7

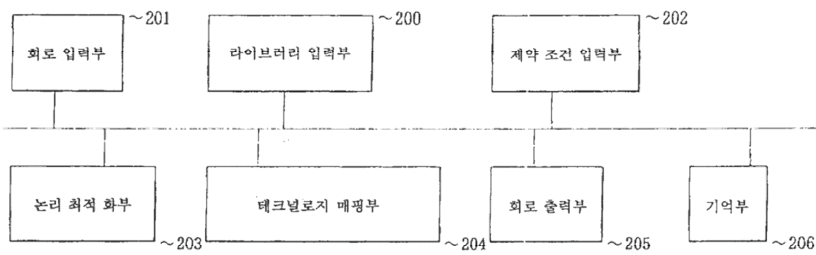
제2항에 있어서, 상기 회로 규모의 척도로서 곱항수(積項數) 및/또는 팬인(fan in)을 이용하는 것을 특징으로 하는 논리 회로 합성 방법.

도면

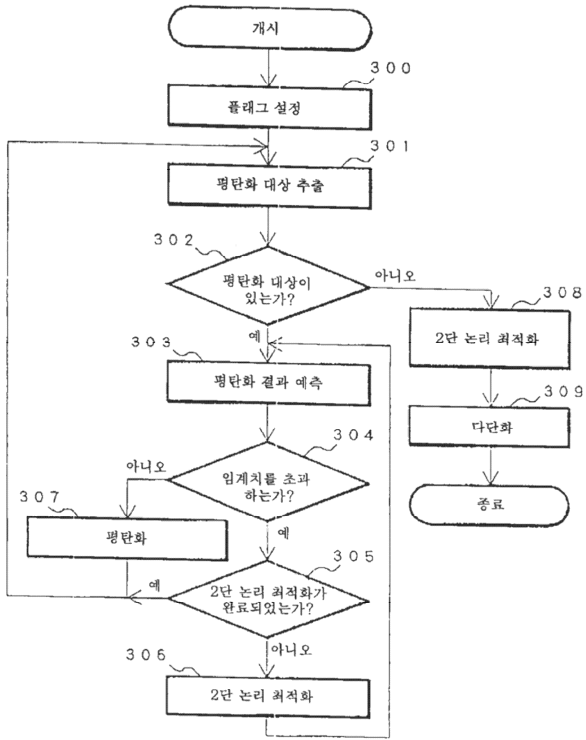
도면1



도면2



도면3



도면4

$$\left. \begin{aligned} x &= i1 * i2 + i1 * i2 \\ x &= i1 * i2 + i2 \end{aligned} \right\} \sim 401$$

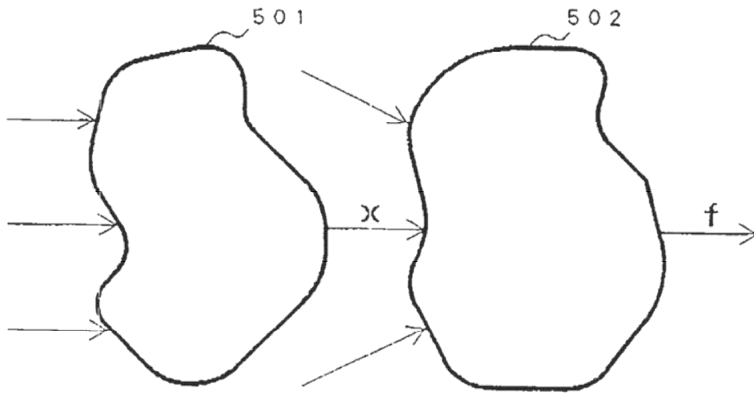
$$f = x * i3 * i4 + x * i3 * i4 + x * i4 * i5 * x * i4 * i5 + i5 \quad \sim 402$$

$$\left. \begin{aligned} x &= i2 \\ x &= i2 \end{aligned} \right\} \sim 403$$

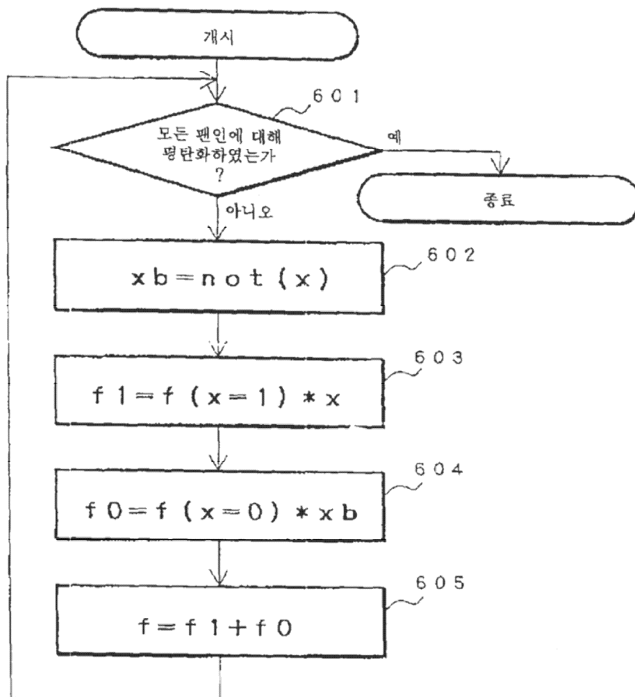
$$f = x * i3 + x * i4 + i5 \quad \sim 404$$

$$f = i2 * i3 + i2 * i4 + i5 \quad \sim 405$$

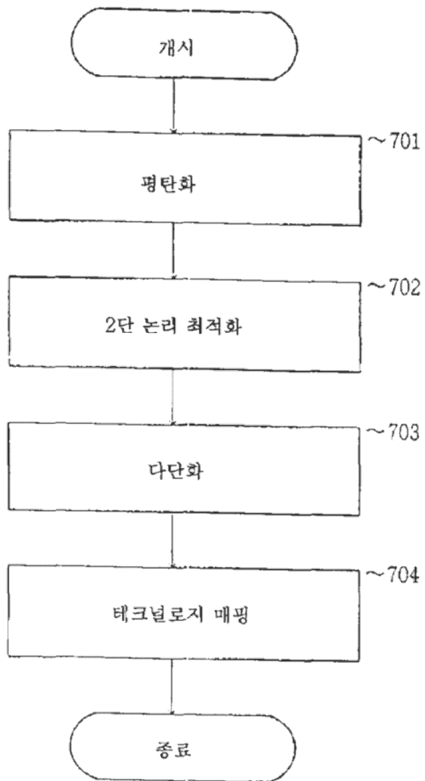
도면5



도면6



도면7



도면8

