

論文99-36C-10-6

비동기 회로 합성을 위한 평선 해저드 제거 알고리즘

(An Algorithm on Function Hazard Elimination for Asynchronous Circuit Synthesis)

柳 光 基 *, 鄭 正 和 *

(Kwang-Ki Ryoo and Jong-Wha Chong)

요 약

본 논문에서는 비동기 논리 회로 합성을 위해서 신호선이 그래프 상에서 직접 평선 해저드를 제거하고 신호선의 추가에 따른 면적의 오버헤드를 최소화하는 새로운 알고리즘을 제안한다. 기존의 평선 해저드 제거 방법은 신호선 사이의 전이 관계를 나타내는 신호선이 그래프로부터 상태를 할당하여 얻어지는 상태 그래프를 이용하였다. 이 방법은 해저드의 제거를 위해 동기 시스템에서 사용하는 방법을 그대로 적용할 수 있으나, 상태 그래프의 구성과 조작에 많은 시간이 소요되는 단점이 있다. 이에 따라 신호선이 그래프를 직접 이용하는 방법이 제시되었으나 해저드의 제거에 따른 면적의 오버헤드는 고려되지 않았다.

본 논문에서는 신호선이 그래프로부터 직접 해저드를 제거함으로써 기존의 상태 그래프를 이용하는 방법에 비해 계산량을 줄이고, 추가되는 신호를 구현하기 위한 논리회로의 크기는 최소항과 적항의 개수를 조절하여 최소화하였다. 제안하는 알고리즘을 벤치마크 데이터로 실험한 결과 면적의 오버헤드가 평균 15% 이상 감소함을 확인하였다.

Abstract

In this paper, a new function hazard elimination algorithm is proposed for asynchronous circuit synthesis. In previous approach, function hazard is eliminated by using state graph which is obtained from the state assignment on STG(signal transition graph) representing transition relationship among signals. These algorithms can use conventional hazard removal and synthesis method applied in synchronous system, but it has much computational complexity and takes much time to handle the state graph. Although some hazard elimination algorithm from STG were proposed, it could not reduce the area overhead due to the addition of new signals.

The proposed algorithm eliminate function hazard directly on STG and also control the number of minterms and product-term of added signal in order to minimize the area overhead. Experimental results on benchmark data shows that overall circuit area after hazard elimination is decreased about 15% on the average than that of previous method.

Keyword : asynchronous, function, hazard, STG, state

I. 서 론

비동기 시스템은 전체 시스템의 동기화를 위한 글로벌

클락을 사용하지 않고 필요한 모듈에만 클락을 사용 때문에 동기 시스템에 비해 고주파로 동작하더라도 클락 스쿠 문제가 발생하지 않고, 클락으로 인한 전력 소비를 줄임으로서 저전력 회로의 구현이 용이하다는 장점을 갖는다. 그러나 비동기 시스템은 글로벌 클락을 사용하지 않기 때문에 모듈간의 통신시 정확한 동기화가 곤란하기 때문에 해저드가 발생하기 쉽다. 이러한

* 正會員, 漢陽大學校 電子工學科

(Dept. of Electronics Eng., Hanyang University)

接受日字:1999年6月10日, 수정완료일:1999年9月28日

해저드는 비동기 시스템의 정상적인 동작을 위해 반드시 제거되어야만 한다. 이때 해저드의 제거를 위해서는 부가적인 논리 회로를 구현해서 추가해야 하므로 회로의 크기가 증가하는 문제점이 발생한다^[1].

비동기 회로 합성을 위하여 해저드를 제거하는 방법은 크게 상태 그래프(state graph)를 이용하는 방법 [2-5]과 신호전이 그래프(signal transition graph)를 이용하는 방법[6-9]으로 나눌 수 있다. 기존의 해저드 탐색 및 제거 방법은 신호전이 그래프로부터 논리 회로 내의 신호의 변화에 따라 각각의 상태를 할당한 상태 그래프를 이용하였다. 이 방법은 해저드의 발견 및 제거에 기존의 동기 논리 회로에서 사용된 방법을 그대로 이용할 수 있으나 상태 그래프의 크기가 신호선의 개수에 대하여 지수 함수적으로 비례해서 커지기 때문에 회로의 신호의 개수가 증가함에 따라 상태 그래프의 계산과 조작에 과도한 시간이 소요되는 단점이 있다. 상태 그래프를 이용하는 대표적인 Lavagno[2]의 방법은 상태 그래프의 계산에 많은 시간이 소요되고 알고리즘의 적용 범위가 특정한 경우, 즉 safe, free choice petri net에 한정되었다. 이를 보완하여 동기 회로 합성에 사용되는 FSM(finite state machine)을 이용하여 모든 형태의 신호전이 그래프에 적용되는 Moon^[3]의 방법이 제안되었으나 상태 그래프를 사용하여 상태의 개수에 비례하는 계산 복잡도를 가진다.

반면 신호전이 그래프를 이용하는 방법은 신호선의 수에 비례하는 계산 복잡도를 가지므로, 상태 그래프를 이용하는 경우보다 훨씬 계산 복잡도를 줄일 수 있는 장점이 있다. 그러나, 최근에 연구가 진행되어 아직까지 평선 해저드 제거를 위한 완전한 알고리즘이 개발되지 않은 상태이다. 예를 들어 Meng^[6]은 상태 그래프를 구성하지 않고 신호전이 그래프에서 직접 평선 해저드를 발견하는 방법을 제시했으나 한정된 형태의 신호전이 그래프인 MG(marked graph) 신호전이 그래프만을 대상으로 했으며, 해저드를 제거하는 구체적인 알고리즘은 제시하지 않았다. Nagalla^[11]는 Meng의 알고리즘을 개선하여 모든 형태의 신호전이 그래프에 적용할 수 있는 해저드 탐색 및 제거 알고리즘을 제시하였으나 추가되는 신호선에 따른 면적의 오버헤드를 감소시키는 방법은 제시하지 못했다.

본 논문에서는 이러한 점을 개선하여 신호전이 그래프를 이용해서 평선 해저드를 제거하는 방법 중에서 아직까지 연구가 진행되지 않은 면적의 오버헤드 최소

화에 관해서 연구하였으며, 제안하는 알고리즘의 효율성을 미국의 버클리 대학에서 개발한 논리합성 시스템인 SIS(sequential interactive synthesis)의 결과와 비교하여 입증하였다. 본 논문의 구성은 다음과 같다. 2장에서는 해저드의 정의 및 종류에 대하여 설명하고, 3장에서는 비동기 논리 회로의 모델링에 사용되는 신호전이 그래프에 대해서 설명하고 평선 해저드가 신호전이 그래프에서 어떻게 발견되어 제거되는지에 대해서 알아본다. 4장에서는 본 논문에서 제안하는 면적 오버헤드의 최소화를 목적함수로 하는 평선 해저드 알고리즘에 대하여 설명한다. 5장에서는 실험 결과를 기존의 방법과 비교하여 설명하고, 6장에서 결론을 맺는다.

II. 해저드

논리 회로에서 입력 신호의 변화에 대해 예상되지 않는 출력 신호의 변화가 발생했을 때 이러한 신호를 해저드라 한다. 해저드는 출력 신호가 변화하는 형태에 따라서 정적(static)해저드와 동적(dynamic) 해저드로 분류하며, 해저드의 발생 원인에 따라 로직(logic)해저드와 평선(function) 해저드로 분류한다. 그림 1(a)와 같이 입력신호에 따른 출력신호가 변화 없이 0(또는 1)으로 유지되어야 할 때 불필요하게 나타나는 1(또는 0)의 신호를 정적 해저드라 하며, 그림 1의 (b)에서와 같이 출력의 초기 상태(initial state)와 정상 상태(steady state)가 다른 상태에서 나타나는 불필요한 신호를 동적 해저드라 한다.

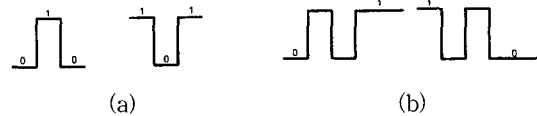


그림 1. 정적해저드와 동적해저드

(a) 정적해저드 (b) 동적해저드

Fig. 1. Static hazard and dynamic hazard.

(a) static hazard (b) dynamic hazard.

발생 원인에 따라 해저드를 분류하면, 하나의 함수를 구성하는 논리가 원인인 해저드를 로직 해저드라 하며, 논리를 구현하는 함수 자체가 원인이 되어 발생하는 해저드를 평선 해저드라 한다. 로직 해저드는 함수를 구성하는 논리를 수정해서 제거해야 하며, 평선 해저드는 함수의 온-셀(on-set)과 오프-셀(off-set)을 수정하여 새로운 함수를 구성함으로써 제거한다.

그림 2의 (a)는 로직 해저드가 발생하는 회로를 나타내며, 그림 2의 (b)는 로직 해저드를 제거한 회로를 나타낸다. 아래의 식 1과 식 2가 등호로 연결된 것은 같은 함수이기 때문이다.

$$F = ab + b'c \dots\dots\dots (식 1)$$

$$= ab + b'c + ca \dots\dots\dots (식 2)$$

그러나 식 1과 식 2는 그림 2의 (a)와 (b)에서 알 수 있듯이 서로 다른 논리로 구현되어 있다. 즉 동일한 입력 신호에 대해서 동일한 출력 신호를 나타내지만 함수를 구현하는 실제 회로(또는 논리)는 다르다. 그림 2의 (a)에서 입력 a와 c가 1로 유지되고 입력 b가 1에서 0으로 바뀔 경우 ①번 게이트가 ②번 게이트보다 더 큰 지연시간을 갖고 있다면 출력 F는 정상적으로 1이 유지된다. 그러나 ①번 게이트가 ②번 게이트보다 작은 지연시간을 갖고 있다면 순간적으로(정확히 ①번과 ②번의 지연 시간차이에 해당하는 시간동안) 0의 값이 출력 F에 나타난다. 이러한 신호를 로직 해저드라 하며 그림 2의 (b)에서와 같이 ③번 게이트와 같은 컨센서스(consensus)항을 추가함으로써 제거할 수 있다.

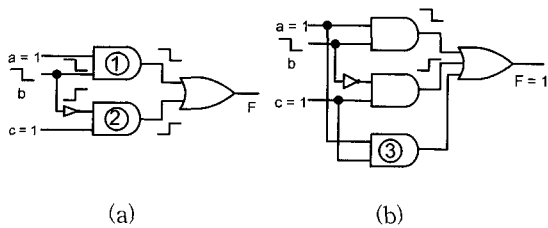


그림 2. 로직 해저드가 존재하는 회로와 제거된 회로
 (a) 로직해저드가 존재하는 회로
 (b) 로직 해저드를 제거한 회로
 Fig. 2. Example with logic hazard and its removal.
 (a) Example circuit with logic hazard.
 (b) Elimination of logic hazard.

평선 해저드가 존재하는 회로의 예를 그림 3에 보인다. 그림의 회로에서 입력(abc)가 (111)의 상태에서 (001)의 상태로 전이되는 경우, 입력 a와 b중 어떤 신호가 먼저 변화하느냐에 따라 출력 파형이 다르게 나타난다. 즉 입력 b가 a보다 먼저 0으로 전이된 후 a가 전이된다면 출력 F는 1을 유지한다. 그러나 입력 a가 b보다 먼저 전이된다면 출력 F는 순간적으로 0을 나타내므로 평선 해저드가 발생한다.

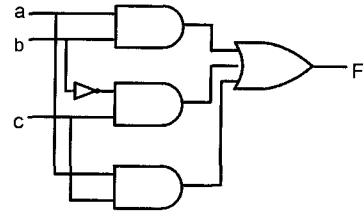


그림 3. 평선해저드가 존재하는 회로
 Fig. 3. Example circuit with function hazard.

이러한 해저드는 카르노 맵에서 (011)상태를 온-셀으로 바꿈으로서 제거할 수 있다. 즉 그림 4와 같이 a와 b의 전이순서에 관계없이 출력을 일정하게 유지하기 위해 입력이 (011)인 상태를 온-셀으로 바꿔서 새로운 함수를 만들어야만 제거된다.

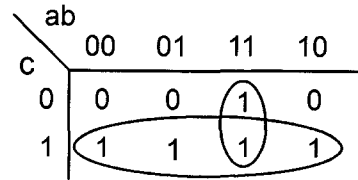


그림 4. 평선 해저드가 제거된 회로의 카르노 맵
 Fig. 4. Karnaugh map after elimination of function hazard.

III. 신호천이 그래프와 평선 해저드

신호천이 그래프는 비동기 논리 회로에서 각 신호간에 전이 관계를 나타낸 그래프이다. 이것은 페트리 네트(Petri net) 또는 타이밍 다이어그램으로부터 얻어지며 Chu^[1]에 의해 처음 소개된 이후 비동기 논리 회로의 합성에 가장 널리 쓰이고 있다. 그림 5는 타이밍 다이어그램으로부터 신호천이 그래프를 구성하는 예를 보인다. 신호천이 그래프에서 각 노드는 신호들의 전이를 나타내며 ‘↑’는 신호의 상승(rising)을 ‘↓’는 하강(falling)을 의미한다. 노드들을 연결한 연결선은 신호의 시간적 순서 관계를 나타낸다. 그림 5의 (a)에서 a와 b의 상승전이가 c의 상승전이를 발생시키는 관계가 그림 (b)에서 연결선으로 표시되어 있다. 그림 (b)에서 양쪽 가장자리의 두 연결선에 그려져 있는 점들은 마킹(marking)이라 하며 최초의 발생신호가 c-입을 나타내고 있다.

서는 x 의 온-셀(그림 7에서 회색으로 칠해진 부분)수를 조절하기 위해 각 경우마다 새로운 상태 그래프를 구성해야만 한다. 따라서 이 방법은 신호의 수가 늘어날 수록 상태 그래프의 계산에 소요되는 시간이 신호의 개수에 대한 지수함수에 비례해 커지기 때문에 회로의 신호수가 많은 회로일수록 계산 시간이 증가하게 된다.

그림 6에서 상태 충돌이 발생한 원인을 살펴보면 그림 6(a)의 신호전이 그래프에서 a^+ 가 일어난 후 곧바로 a^- 가 일어났기 때문인 것을 알 수 있다. 즉 신호 a 가 상승된 후 곧바로 하강했기 때문에 상태 그래프에서 상태 충돌로 나타나게 된 것이다. 따라서 상태 그래프에서의 상태 충돌은 신호전이 그래프에서 보수관계에 있는 천이(complementary transition)들의 집합으로 나타나게 된다. 이것은 다음과 같이 정의되며 신호전이 그래프에서 CSC여부를 검사하는 중요한 기준이 된다^[26].

[정의3] 신호전이 그래프에서 어떠한 천이(transition)와 그와 보수관계에 있는 천이들로 구성된 경로 중에서 전체 천이를 모두 포함한 경우를 제외한 나머지를 complementary path라 정의한다.

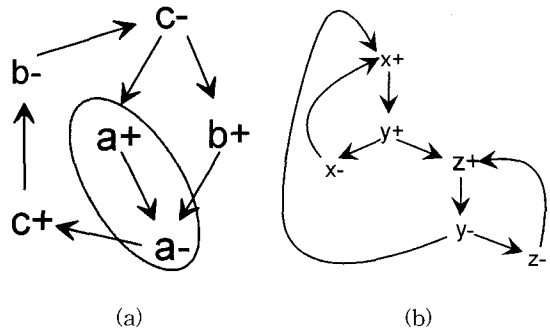


그림 8. CP의 예
Fig. 8. Example of CP.

그림 8은 간단한 complementary path(이하 CP)의 예를 나타낸다. 그림 8(a)에서 CP는 $\{a^+, a^-\}$ 가 되며 그림 8(b)에서는 $\{x^-, x^+\}$, $\{y^+, z^+, y^-, z^-\}$, $\{y^-, x^+, y^+, x^-\}$ 가 CP가 된다. CP에 속해 있는 천이들을 거치면 상태 그래프에서 할당된 이진 코드가 같은 상태를 생성하게 되며 이것은 평선 해저드가 존재한다는 증거가 된다.

IV. 면적 최소화를 위한 평선 해저드 제거

평선 해저드를 제거하는 방법은 크게 두 가지로 분

류할 수 있는데 첫째는 신호전이 그래프로부터 얻어진 상태 그래프에서 상태 충돌을 찾아낸 후 새로운 변수를 추가로 할당하여 상태 충돌을 제거하는 방법을 말하며, 둘째는 신호전이 그래프에서 complementary path를 찾아낸 후 새로운 변수를 삽입해서 complementary path를 제거하는 방법이다. 기존의 알고리즘은 첫 번째 방법의 상태 그래프를 이용하기 때문에 계산 복잡도가 신호의 개수에 지수함수적으로 비례하는 단점이 있었다. 이 점을 보완해서 평선 해저드의 탐색 및 제거에 신호전이 그래프를 직접 이용함으로써 계산 복잡도를 신호선의 개수에 비례하는 계산 복잡도로 낮추고 보다 큰 회로에서의 응용성을 높이는 방법이 두 번째 방법이다. 두 번째 방법과 같이 신호전이 그래프를 직접 이용하는 방법은 평선 해저드를 발견하는 알고리즘은 발표되었으나 별도로 추가되는 논리 회로의 최소화를 위한 알고리즘은 개발되지 않았다. 제안하는 알고리즘은 이를 보완해서 신호전이 그래프에서 직접 평선 해저드를 찾아서 제거하고 추가되는 신호 x 의 위치에 따른 온-셀의 크기를 조절함으로써 추가되는 회로의 최소화를 유도한다.

신호전이 그래프를 이용한 평선 해저드 제거 방법은 먼저 CP를 찾아낸 후, CP가 CSC를 위반하는 경우에 CSC를 만족하도록, 즉 CP가 존재하지 않도록, 새로운 신호를 신호전이 그래프에 추가한다. 그림 8(a)에서 $\{a^+, a^-\}$ 와 같은 경로를 신호전이 그래프에서 찾아내서 그 사이에 새로운 신호인 x 를 삽입한다. 제안하는 알고리즘은 추가 신호선을 구현하기 위한 회로 크기를 최소화하기 위해 신호선 x 의 논리식에 포함되는 리터럴 수를 가능하면 적게 유도한다. 리터럴의 개수를 최소화하려면 카르노 맵 상에서 온-셀의 크기가 2의 지수승 개수를 만족하여야 한다. 따라서 초기에는 온-셀의 크기를 2의 지수승 개수로 설정한 후, CP가 존재하는지 여부를 판단하고, 만약 온-셀의 크기가 2의 지수승을 만족하면서 CP가 제거되는 경우가 없을 경우에는 온-셀의 크기를 3개 혹은 그 이상으로 증가시켜가면서 CP의 제거 여부를 판단한다.

추가 신호 x 에 대한 논리식은 그림 7의 상태 그래프에서 회색으로 칠해진 x 의 온-셀을 카르노 맵에 표시한 후 얻어진다. 여기서 x 의 온-셀은 x^+ 천이 이후에 x^- 천이가 나타날 때까지의 상태들의 집합이다. 각 상태에서 x 가 1인 상태들이 온-셀이 아님에 유의해야 한다. 신호전이 그래프에서는 x^+ 와 x^- 사이의 천이의 개

수에 하나를 더한 값이 이 x 에 대한 온-셀의 크기이다. 여기서 x 의 논리식을 최소화하기 위해서는 온-셀의 크기에 따른 적항의 개수를 조절해야 한다. 적항의 개수는 리터럴의 개수와 함께 구현 논리 회로의 간소화를 위해 최소화되어야 한다. 가장 이상적인 경우의 적항의 개수가 1개인 경우(즉, 온-셀의 개수가 2의 지수승일 경우)에 리터럴의 개수가 최소로 되는 경우이다. 그러나 실험적으로, 적항의 개수가 1개로 줄어들 확률은 온-셀의 개수에 비례하여 지수 함수적으로 감소한다. 따라서 CP를 제거하기 위해 새로운 신호를 추가할 경우 새로 추가된 신호의 상승 천이($x+$)과 하강 천이($x-$)의 간격을 2의 지수승으로 제한하였을 때, 즉 온-셀의 크기가 2의 지수승일 때, 적항의 개수가 1개로 될 확률은 두 천이 사이의 간격이 커질수록(온-셀의 크기가 커질수록) 작아진다. 그러나 온-셀의 크기가 2개일 경우, 즉 $x+$ 와 $x-$ 사이의 거리가 2인 경우는 그림 9에서와 같이 또 다른 CP($\{a+,x+,a-,x-\}$)를 생성하기 때문에 CSC를 만족하지 못하게 되므로 제외된다.

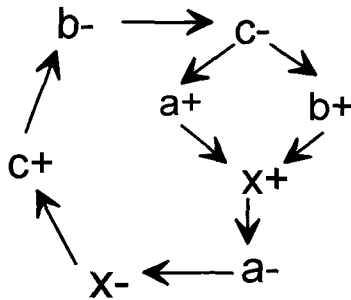


그림 9. CP가 존재하는 신호천이 그래프
Fig. 9. STG with CP.

제안하는 알고리즘에서는 위와 같은 경우를 모두 고려하여 $x+$ 와 $x-$ 의 온-셀의 수를 초기에 2의 지수승 개수로 설정하여 CP를 제거하는 $x+$ 와 $x-$ 의 위치를 탐색한 후 그러한 위치가 존재하지 않는 경우에는 적항의 개수가 두 개로 구성되는 위치를 탐색한다. 그림 10은 온-셀의 크기가 3인 경우에 CP를 제거하는 예를 나타낸다. 그러나 상태 그래프를 고려하지 않고 $x+$ 와 $x-$ 의 위치를 조절하면 두 천이가 서로 concurrent한 위치에 있을 수가 있다. Complementary transition은 concurrency 관계에 있을 수 없으므로 이러한 경우는 $x+$ 와 $x-$ 의 위치를 다시 바꾸어 온-셀의 개수를 4개 혹은 그 이상으로 조절하면서 더 이상 concurrency가 나타나지 않을

때까지 $x+$ 와 $x-$ 의 위치를 변화시킨다. 그림 10은 제안하는 알고리즘에 의해 그림 6(a)를 수정한 것이며 그림 11은 그림 10에 대한 상태 그래프와 카르노 맵을 나타낸다. 그림 10에서는 $x+$ 와 $x-$ 가 concurrency관계에 있지 않으므로 온-셀의 개수가 초기 값 3개로 유지되었다.

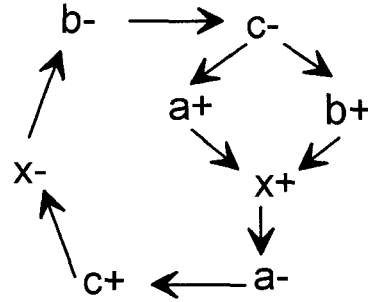


그림 10. CP를 제거한 신호천이 그래프
Fig. 10. STG without CP.

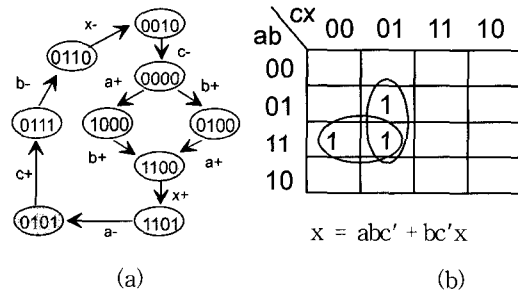


그림 11. 그림 10에 대한 상태 그래프와 카르노 맵
Fig. 11. State graph and K-map of Fig. 10.

그림 11은 제안하는 알고리즘으로 구성된 그림 10의 신호천이 그래프로부터 구성된 상태 그래프와 추가 신호 x 에 대한 카르노 맵을 나타낸다. 신호 x 의 논리식이 [9][4]에 의한 방법에 의한 논리식보다 최소화되었음을 알 수 있다.

그림 12는 제안하는 알고리즘에 대한 전체적인 순서도로 알고리즘의 전체 과정은 다음과 같다. CSC를 만족하지 않는 신호천이 그래프로부터 CP를 찾아낸 후, 각 CP에 추가되는 신호 x_i- 를 삽입하여 CP를 제거하고 CSC를 만족하는 신호천이 그래프를 생성한다. 이때 x_i+ 와 x_i- 는 그림 10과 같이 CP의 마지막 신호($a-$) 바로 앞에 x_i+ 를 삽입하고 설정된 온-셀의 크기에 따라 CP 다음의 신호 다음에 x_i- 를 삽입한다. 실험적으로 온-셀의 크기가 2의 지수승의 개수이면서 CP를 제

거하는 경우보다 온-셀의 개수가 세 개인 경우가 대부분이었다. 예를 들어 온-셀의 크기를 세 개로 제한한 경우에는 CP의 마지막 신호(a-) 바로 앞에 xi+를 삽입하고 CP 바로 다음의 신호 다음에 xi-를 삽입한다. xi+와 xi-가 concurrency 관계에 있을 경우 xi-의 위치를 xi-다음 신호인 b-의 다음 위치로 이동시키며, xi-를 더 이상 변화시킬 수 없을 경우는 xi+를 xi+이전 신호(a+ 또는 b+)의 팬인 위치로 이동시킨다.

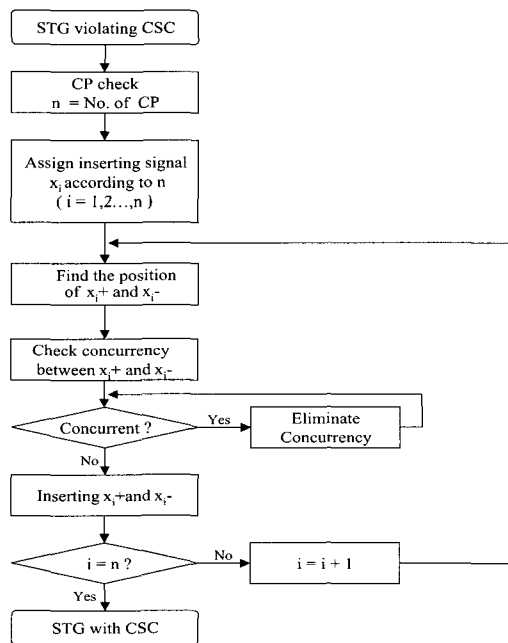


그림 12. 제안하는 알고리즘의 순서도
Fig. 12. Flow chart of proposed algorithm.

순서도에 따라 알고리즘을 단계별로 설명하면 다음과 같다.

단계 1 : 신호천이 그래프에서 CP를 탐색하여 추출한다.

단계 2 : CP의 개수(n)에 따라 CP에 삽입할 신호 xi를 할당한다.

단계 3 : 추가되는 신호 xi에 대해 xi+와 xi-를 초기 위치에 삽입한 후, CP를 제거하는 위치를 찾는다. 초기 위치는 xi+를 CP내의 마지막 신호 바로 앞에 삽입하고 xi-를 xi+와의 거리가 2의 지수승 개로 초기화한다. 이 조건을 만족하는 위치에서 CP가 존재할 경우 온-셀의 크기를 3부터 증가시키면서 CP를 제거하는 위치에 삽입한다.

단계 4 : xi+와 xi-의 concurrency를 조사한다.

단계 5 : 만약 xi+와 xi-가 concurrency 관계에 있다면 xi+와 xi-의 위치를 변화시켜 concurrency 관계를 제거한다. 이 때의 xi+와 xi-의 위치는 xi에 대한 온-셀의 수를 단계적으로 1개씩 늘려서 변화시키며, xi+와 xi-의 위치가 concurrency관계가 아닐 때 xi+와 xi-를 신호천이 그래프에 삽입한다.

단계 6 : 모든 CP가 제거되었으면 알고리즘을 마치고, 모든 CP가 제거되지 않았으면 단계 3으로 돌아가 모든 CP가 제거될 때까지 단계 3, 4, 5의 과정을 반복한다.

V. 실험 결과

본 논문에서는 제안하는 평선 해저드 제거 알고리즘을 C-언어로 프로그래밍하여 벤치마크 테스트 데이터에 대하여 실험한 결과를 비교하였다. 실험은 Lavagno와 Moon^[2]의 알고리즘이 적용된 논리합성 시스템인 SIS 프로그램을 이용하여 평선 해저드가 제거된 신호천이 그래프와 제안하는 알고리즘을 이용하여 평선 해저드가 제거된 신호천이 그래프를 각각 SIS의 논리합성 과정을 거쳐서 완전한 이단 논리 회로로 합성한 후, 그 결과에 대하여 각각 게이트의 개수, 회로의 크기, 리터럴의 개수 및 수행 시간의 네 가지 항목에 대해서 비교하였다. 위의 과정을 통하여 얻어진 실험 결과 표 1에 비교하여 나타내었다. 비교 항목은 SIS를 기준으로 했을 때 제안하는 알고리즘에 의해 감소된 면적을 백분율(%)로 표시한 것이다. 비교란은 감소하였을 경우 음의 부호로 표시하였다. 표에서 알 수 있듯이 게이트의 개수는 최대 20개까지 줄일 수 있으며 평균 15.24% 줄어들었다. 회로의 크기를 면적으로 비교한 결과 최대 47.6%, 평균 17.67%이상 크게 감소하였음을 확인하였다. 수행 시간을 보면 회로의 크기가 작은 경우에는 다소 증가하는 예도 있으나 대부분 수행 시간이 감소하였다. 특히 회로의 크기가 크게 합성되는 경우엔 수행 시간이 크게 감소하는 것으로 나타났으며 평균 수행 시간은 9.86% 감소하였다.

VI. 결론

본 논문에서는 비동기 논리 회로의 합성을 위한 평선 해저드 제거 및 면적의 오버헤드의 최소화에 관한 알고리즘을 제안하였다. 기존의 방법은 상태 그래프를

표 1. 실험 결과 비교 데이터

Table 1. Experimental results.

회로명	Moon ^[2] 의 방법				제안하는 방법				비교(%)			
	Gate	Area	Lits	Time	Gate	Area	Lits	Time	Gate	Area	Lits	Time
alex2n	10	30192	21	3.1	7	20210	11	3.2	-30	-33.1	-47.6	+3.2
atod	20	30384	32	4.5	12	30256	24	4.2	-40	-0.4	-25	-6.7
breq	19	20376	32	4.7	15	10368	29	4.5	-11.8	-49.1	-10.3	-4.2
breq+do	42	40192	79	17.1	40	30880	76	16.2	-4.8	-23.2	-3.8	-5.3
master-read	85	71856	175	32.4	65	51344	117	21.6	-23.5	-28.5	-33.1	-33.3
sbuf-ram-write	44	50382	69	14.3	41	30848	77	11.0	-6.8	-38.8	+11.6	-23.1
sbuf-read-ctl	11	10248	24	3.7	11	10248	24	3.6	0	0	0	-2.7
sbuf-send-ctl	52	31064	83	18.7	45	25015	63	14.3	-13.5	-19.5	-24.1	-23.5
nousc-ser	11	10240	23	3.0	10	10120	20	3.2	-9.1	-1.2	-13	+6.7
vbe4a	39	30776	64	13.8	28	30576	49	11.5	-28.2	-0.6	-23.4	-16.7
vbe5b	13	20272	22	3.4	13	20272	13	3.3	0	0	0	-2.9
vbe6a	Error				20	38244	36	9.5

이용하여 평선 해저드를 제거하였으나, 이 방법은 상태 그래프의 복잡도가 신호의 개수에 지수함수적으로 증가함으로 인해 알고리즘의 계산 복잡도가 커지는 문제를 가지고 있으며, 해저드를 제거하기 위하여 추가되는 신호에 대한 부가적인 회로의 크기를 최소화하지 못하는 단점을 가진다. 본 논문에서는 이러한 단점을 보완하여 평선 해저드의 제거에 신호선이 그래프를 직접 이용하여 계산 복잡도를 낮추었고 추가되는 신호의 온-셀의 크기와 적항 수를 조절함으로써 추가 신호의 구현 논리식과 그에 따른 면적의 오버헤드의 최소화를 유도하는 새로운 알고리즘을 제안하였다.

제안하는 알고리즘의 효과를 입증하기 위하여 벤치마크 데이터로 실험한 결과, 게이트의 개수, 합성된 후의 회로의 크기 및 수행 시간의 측면에서 모두 우수한 것으로 나타났다. 특히 수행 시간은 회로의 크기가 커짐에 따라 훨씬 향상된 결과를 나타냄으로써 회로의 크기가 큰 경우에도 적용 가능성을 알 수 있다. 따라서 큰 규모의 회로에 존재하는 해저드 제거에 특히 효율적이며 실질적인 VLSI 회로의 합성에 적용될 수 있다고 판단된다. 앞으로의 연구과제는 회로의 동작에 영향을 미치는 concurrency를 최대한 유지하면서 동시에 해저드의 제거에 따른 면적의 오버헤드를 줄여나가기 위한 연구가 지속되어야 하겠다.

참 고 문 헌

- [1] Scott Hauck. "Asynchronous Design Methodologies: An Overview," Proc. of the IEEE, 83(1): 69-93, January 1995.
- [2] L. Lavagno, C. Moon, R. Brayton, and A. Sangiovanni Vincentelli. "Solving the State Assignment Problem for Signal Transition Graphs," Proc. of 29th DAC, pages 568-572, 1992.
- [3] C. W. Moon P. R. "Synthesis and Verification of Asynchronous Circuits from Graphical Specification" PhD thesis, U.C. Berkeley, 1992.
- [4] R. Puri and J. Gu "A Modular Partitioning Approach for Asynchronous Circuit Synthesis," Proc. of DAC, pages 63-69, 1994.
- [5] P. Vanbekbergen, F. Catthoor, G. Goossens, and H. Deman. "A Generalized State Assignment Theory for Transformations on Signal Transition Graphs," Proc. of ICCAD, pages 112-117, 1992.
- [6] Meng. L. Yu, and P. A. Subrahmanyam. "A

- New Approach for Checking the Unique State Coding Property of Signal Transition Graphs," Proc. of EDAC, pages 312-321, 1992.
- [7] K.J. Lin, and C. S. Lin "Automatic Synthesis of Asynchronous Circuits," Proc. of 28th DAC, pages 296-301, 1991.
- [8] P. Vanbekbergen, F. Catthoor, G. Goossens, and H. Deman. "Optimized Synthesis of Asynchronous Control Circuits from Graph-theoretic Specifications," Proc. of ICCAD, pages 183-187, 1990.
- [9] E. Pastor, and J. Cortadella. "An Efficient Unique State Coding Algorithm for Signal Transition Graphs," Proc. of DAC, pages 173-177, 1994.
- [10] Tam-Anh Chu, "Synthesis of Self-Timed VLSI circuits from Graph-theoretic Specifications," PhD thesis, MIT, June, 1987.
- [11] Radhakrishna Nagalla and Graham Hellestrand "A Visual Approach for Asynchronous Circuit Synthesis," International Conf. VLSI Design, pages 329-335, 1996.
- [12] E. B. Eichelberger. "Hazard Detection in Combinational and Sequential Switching Circuits," IBM Journal of Research and Development, pages 90-99, 1965.

저 자 소 개



鄭正和(正會員)

1975년 한양대학교 전자공학과(학사). 1977년 한양대학교 전자공학과(석사). 1981년 일본 와세다대학교 전자통신공학과(박사). 1979년~1980년 일본 NEC 중앙연구소 위촉연구원. 1983년~1984년 KIET(Korea Institute of Electronics & Technology) 위촉연구원. 1986년~1987년 미국 Berkeley대학 초빙교수. 1993년~1994년 대한전자공학회 CAD 및 VLSI 분과 위원장. 1995년~1996년 대한전자공학회 교육이사. 1996년~1997년 영국 Newcastle대학 초빙교수. 1997년~1998년 대한전자공학회 편집이사. 1997년~1999년 한양대학교 정보통신원 원장. 1999년~현재 대한전자공학회 학술이사. 1981년~현재 한양대학교 전자전기공학부 교수. 관심분야는 VLSI의 CAD, ASIC 에뮬레이션 시스템 개발, MPEG 디코더/인코더 설계, 통신 회로 설계, 특히 무선 모뎀 칩 개발



柳光基(正會員)

1986년 한양대학교 전자공학과(학사). 1988년 한양대학교 전자공학과(석사). 1991년 한양대학교 전자공학과 박사과정 수료. 1991년~1994년 육군사관학교 교수부 전자공학과 교수. 1994년~현재 한양대학교 강사. 관심분야는 VLSI의 CAD, ASIC 에뮬레이션 시스템 개발, 초미세 배선 설계