

조합논리회로를 위한 새로운 Path-Based 타이밍 최적화 알고리즘

(Path-based New Timing Optimization Algorithm for Combinational Networks)

梁 世 陽,* 洪 鳳 喜*

(Sae Yang Yang and Bong Hee Hong)

要 約

논리회로 합성시 조합논리회로의 타이밍 최적화를 위한 새로운 알고리즘을 제안한다. 즉 조합논리회로의 고장검출에서 쓰여왔던 redundancy 개념을 확대한 P-path redundancy 개념을 도입하여 임계경로의 side input들의 P-path redundancy를 제거함으로써 임계 지연시간을 최소화 시키며, statically sensitizable path와 statically unsensitizable path들을 체계적으로 고려함으로써 기존의 방법들보다 정확한 타이밍 최적화를 할 수 있다. 즉, 최장경로에 기초한 기존의 알고리즘들에 의한 타이밍 최적화 방법은 회로의 임계 지연시간을 오히려 늘릴 위험이 존재하나 본 방법은 그런 가능성을 완전히 배제할 수 있다. 또한 본 방법은 논리회로 합성시 기술 독립적인 단계와 기술 종속적인 단계 모두에 쓰여질 수 있다.

Abstract

In this paper, the new timing optimization algorithm for combinational networks is proposed. First, we introduce the concept of P-path redundancy which is the extension of redundancy concept used in the testing of combinational networks. In this approach, the critical delay is minimized by removing the P-path redundant side inputs of the critical path, and more accurate timing optimization is possible by systematically considering the statically unsensitizable paths as well as the statically sensitizable paths. It's possible with all previous longest path based approaches that the critical delay of resulting network after timing optimization may be even increased. However, the proposed method guarantees to exclude such a possibility, and can be applied to optimize the timing of combinational networks in technology independent, and dependent phase.

I. 서 론

일반적으로 논리회로합성(logic synthesis)시의 타

이밍 최적화는 임계 지연시간(critical delay)을 최소화시킴으로 이루어진다. 지금까지 논리회로 합성시의 타이밍 최적화 연구는 주로 다음 두 갈래의 방향으로 연구되어 왔다. 첫째는 K. J. Singh 등이 제안한 방법으로 임계부회로(critical sub-circuit)를 구하여 이를 collapsing하고, algebraic resynthesis를

*正會員, 釜山大學校 컴퓨터工學科
(Dept. of Computer Eng., Pusan Nat'l Univ.)
接受日字: 1992年 2月 19日

통하여 타이밍 최적화를 이루는 시도였으며,^[1] K. C. Chen과 S. Muroga가 제안한 들깨방법은 제한된 DC (Don't Care) set을 이용하여 임계경로의 level을 줄임으로서 문제를 해결하고자 하였다.^[2] 본 논문에서는 제안되는 방법은 기존의 방법들과 다른, 다음절에서 자세히 기술될 P-path redundancy 개념을 이용한 새로운 알고리즘이다. 즉 조합논리회로의 고장검출에서 쓰여왔던 redundancy 개념을 확대한 P-path redundancy 개념을 도입하여 임계경로의 side input들의 P-path redundancy를 제거함으로써 임계경로의 지연시간을 최소화 시킨다.

일반적으로 논리회로합성이 수행된후 합성된 회로가 원래의 타이밍 제약조건을 만족하는지를 검증하기 위하여 타이밍 분석을 하여야 한다. 이 경우 정확한 타이밍 분석을 하기 위하여서는 회로내의 false path와 viable path들이^[3, 4, 5] 고려되어야 하는데 이들을 고려하여 타이밍 분석을 하는 경우 많은 CPU time이 요구된다. 본 방법은 회로의 타이밍 분석시 많은 시간을 요하게 하는 경로들중 statically unsensitizable path를 체계적으로 제거함으로써 정확한 타이밍 분석을 효과적으로 할수있는 논리회로를 합성한다. 또한 지금까지 발표된 대부분의 타이밍 최적화방법들은 statically unsensitizable path들을 체계적으로 고려하지 않음으로 인하여 항상 다음과 같은 예기치 않은 결과를 얻을 가능성이 있다. 즉 이 경로들을 고려하지 않고 타이밍 최적화를 하는 경우 타이밍 최적화 과정을 거친 결과 회로가 원래의 회로와 비교하여 전혀 개선이 없되거나 최악의 경우 오히려 임계경로 지연시간이 커지는 결과를 초래할수 있다. 그러나 본 논문에서 제안되는 새로운 방법은 statically sensitizable path와 statically unsensitizable path들을 체계적으로 고려하며 타이밍 최적화를 수행함으로써 이를 효과적으로 방지할수 있다. 또한 본 방법의 또 다른 장점으로서는 기술독립적인 단계에서 뿐만 아니라 기술종속적인 단계에서도 적용이 가능하다는 것이다.

본 논문은 다음과 같이 구성된다. II 장에서는 나머지 장들에서 쓰여지는 용어들의 정의를 하며, III 장에서는 본 알고리즘의 기본 아이디어와 statically unsensitizable path를 고려했음때의 접근방법에 대해 설명하며, IV 장에서 알고리즘을 소개하고, 실험과 결론을 V 장에서 언급한다.

II. 정 의

본 장에서는 본 논문에서 쓰여지는 용어들의 정확한

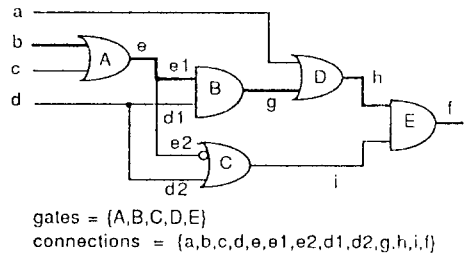
정의를 내리고 이를 간단한 예들로써 설명한다. 이들 용어들은 참고문헌 [3]에서 많이 인용하였다.

[정의 1]: 조합회로(combination network)는 gate(또는 node)들과 gate들간의 connection(또는 edge)들로 이루어진 directed acyclic graph다. 각 gate G는 지연시간 $d(G)$ 를 갖는다.

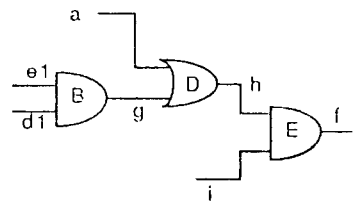
[정의 2]: 조합회로에서 경로(path)라 함은 connection들과 gate들의 alternating sequence이다. 이때 제일 처음의 connection을 starting connection이라 하며 제일 마지막의 connection을 ending connection이라 한다.

[정의 3]: Gate G1의 출력이 gate Gj의 입력에 연결되어 있으면, Gi는 Gj의 fan in이라 하고 Gj는 Gi의 fan out이라 한다. 만약 G1에서 G2로 경로가 있다면 G2는 G1의 transitive fan out(G1는 G2의 transitive fan in)이라 한다. 또한 connection Ci에서 connection들 Cj, ..., Cl들이 갈라져 나갈때 Cj, ..., Cl을 Ci의 fan out connection(Ci를 Ci, ..., Cl의 fan in connection)이라 하고 |fan-out|은 Ci에 연결되어 있는 Cj, ..., Cl의 갯수이다.

[예제 1]: 그림 1. a)는 하나의 간단한 조합회로다. (b, A, e, e1, B, g, D, h, E, f)는 하나의 경로이며 여기서 starting connection은 b이며 ending connection은 f이다. Gate D(Gate E)는 gate E(gate D)의



(a)



(b)

그림 1. 조합회로의 예
Fig. 1. The example of combinational network.

fan in(fan out)이며 gate E는 gate A의 transitive fan out이다.

[정의 4]: Literal은 Boolean 변수가 complemented나 uncomplemented 형태로 나타나는 것이다. Cube는 literal들의 product이다(예, $x^{\wedge}yz$, \wedge 는 complement를 표현). Minterm은 Boolean 함수의 모든 입력변수들이 나타나는 cube이다.

[정의 5]: Event는 0에서 1로, 또한 1에서 0으로 변하는 transition이다. 하나의 event가 조합회로의 임의의 경로를 따라서 전달이 될때 이 경로를 sensitizable하다고 하며, 이 event는 이 경로를 따라 propagation된다고 한다.

[정의 6]: 경로 P의 길이(length)는 그 경로상의 모든 gate들의 delay의 합으로 나타낸다. 임계경로는 조합회로의 모든 sensitizable 경로중 가장 길이가 긴 것이다. 이 임계경로가 시작하는 외부입력(external input)을 임계입력이라 하고 끝나는 외부출력(external output)을 임계출력이라 한다.

[정의 7]: 회로의 최장경로(longest path)는 회로 내의 모든 경로들중 길이가 가장 긴 것이다. 즉 최장경로는 sensitizable 경로뿐만 아니라 unsensitizable 경로도 포함하여 고려한 것이다.

[정의 8]: Gate 입력의 controlling value라 함은 gate의 출력값이 다른 입력들의 값에 상관없이 결정되는 값을 말하며 noncontrolling value라 함은 controlling value가 아닌 값이다. 예를 든다면 OR, NOR(AND, NAND)gate의 controlling value는 1(0)이며 noncontrolling value는 0(1)이다.

[정의 9]: 조합회로에서 임의의 경로를 P라 하자. 이 P에 포함된 gate Gi의 입력들(즉 connection들)중에서 P에 포함되지 않은 입력들을 P의 side input들이라고 한다. 하나의 side input Si에 대하여 경로 P의 부경로(sub-path)는 Si에서부터 시작하여 P를 따라서 P의 ending sequence에서 끝나는 경로를 말한다. 부경로의 정의는 원칙적으로 특정 side input에 대하여 정의되므로 부경로 정의시 이 side input을 명확하게 밝혀야 되나(즉 "Si에 대한 경로 P의 부경로" 식으로) 이 side input이 명확한 경우에는 생략할수 있다.

[정의 10]: 임의의 경로는 이 경로를 통하여 event가 propagation될때, 이 경로의 모든 side input들을 noncontrolling value들로 하는 input cube가 존재하는 경우 statically sensitizable하다고 하며, 이 경로를 statically sensitizable path라고 한다. 따라서 statically unsensitizable path는 statically sensitizable path가 아닌 경로이다.

논리회로합성시의 임계경로 지연시간 최소화를 위하여 다양한 delay model이 쓰여질수 있다. 본 논문에서 제안되는 방법은 특정 delay model에만 적용되지 않고 어떠한 delay model에도 적용되어 질수 있으나, 빠른 이해를 도모하기 위하여 본 논문에 나오는 모든 예제 회로들에는 특별한 언급이 없는 한 unit delay model을 사용한다.

[예제 2]: 그림 1. a)에서 경로 CP1=(b, A, e, e1, B, g, D, h, E, f)는 $\wedge a^{\wedge}cd$ 인 경우 statically sensitizable 하며 길이가 4로 경로(c, A, e, e1, B, g, D, h, E, f)와 함께 임계경로이다. 이 임계경로 CP1의 side input들은 {c, d1, a, i}이며, 이때 side input d1에 대하여 경로 CP1의 부경로는 경로(d1, B, g, D, h, E, f)이다.

[예제 3]: 그림 2에서 경로 CP1=(a, A, d, B, e, e1, C, f2)는 statically unsensitizable path이다. 왜냐하면 이 경로로 event를 전달시키기 위해서는 이것의 side input들을 모두 noncontrolling value로 하여야 한다. 즉 side input b1, c는 1로, side input b2는 0으로 하여야 하는데, b1=1과 b2=0으로 하기 위해서는 외부입력 b를 동시에 0과 1로 하여야 하는 inconsistency가 발생한다.

[정의 11]: 조합회로에서 임의의 connection상의 stuck-at 0/1(SA0/1) 고장이 특정경로 P를 따라서 출력단에서 검출되어지면 이 connection은 P-path irredundant(또한 이 fault는 P-path testable)하다고 하며, 그렇지 않으면 P-path redundant(또는 P-path untestable)하다고 한다.

[정의 12]: 조합회로에서 경로 P가 있을때, 이 경로 P로 구성되는 부회로라 함은 이 경로가 포함되는 최소의 부회로(minimum sub-circuit)이다.

[예제 4]: 그림 1. a)의 회로에서 경로 (b, A, e, e1, B, g, D, h, E, f)를 CP라고 하면 side input d1에 대한 경로 CP의 부경로 (d1, B, g, D, h, E, f)를 경로 P라 했을때 d1의 SA1은 경로 P를 따라 검출이 불가능하므로 d1은 P-path redundant하다. 그림 1. b)는 경로 P로 구성되는 부회로이다.

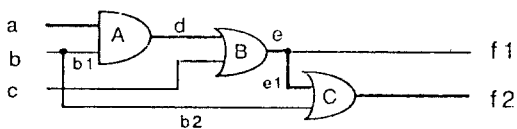


그림 2. Statically unsensitizable path를 가진 회로
Fig. 2. The circuit having a statically unsensitizable path.

III. 접근방법

1. 기본아이디어

본 논문에서 제안되는 알고리즘의 기본 아이디어는 다음과 같다. 회로의 임계경로 지연시간을 줄이기 위하여 임계경로를 CP라고 했을때 이 CP의 side input 들상의 fault들이 해당되는 connection에 대한 CP의 부경로 P에 대하여 P-path redundant인가를 조사하여 이 CP의 길이를 줄여주는 것이다. 임계경로 CP의 side input들중 일부가 부경로 P에 대하여 P-path redundant 하다면 이것들은 임계출력의 Boolean 함수를 변화시키지 않고 제거할수 있다. 이 제거과정은 회로고장검출에서의 redundancy 제거 방법을 쓸수 있다.

그림 1. a)의 회로를 예로 하여 설명하면 이 회로의 임계경로는 그림에서 굵은 선으로 표시된 경로 CP=(b, A, e, e1, B, g, D, h, E, f)이다. 이 경우 side input d1에 대한 경로 CP의 부경로 P=(d1, B, g, D, h, E, f)에 대하여 d1의 SA1은 P-path untestable하다. 이 redundancy를 위의 규칙에 따라 제거하면 그림 3의 회로를 얻을수 있다. 이 경우 바뀌어진 회로의 임계 지연시간은 3으로 1만큼 줄어든다.

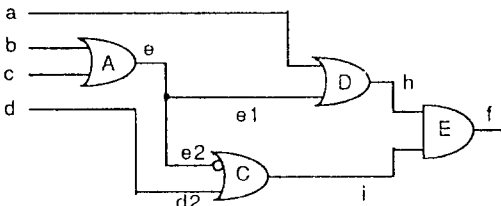


그림 3. 타이밍 최적화된 회로.
Fig. 3. The timing optimized circuit.

한가지 언급할 것은 제안된 본 방법은 testing에서의 redundancy 제거방법이 아니라는 것이다. 차이점으로 첫째, 타이밍 최적화할 회로가 100% stuck-at fault testable 하더라도—즉, 회로내에 redundancy가 존재하지 않는 경우라도—본 방법은 임계 지연시간을 줄일수 있는데 이는 제안된 방법의 redundancy 개념이 “특정” 경로(a specific path)에 기초한 것인 반면에, 기존 redundancy 제거방법은 transitive fan out 관계에 있는 “모든” 경로에 기초한 것이기 때문이다. 둘째, 참고문헌 [3]에서는 임의의 redundancy 제거는 최장 길이에 기초한 타이밍 최적화와 마찬가지로 회로의 임계 지연시간을 오히려 증가시킬수 있다는 것을

보였다. 그러나 다음에서 보이는 바와 같이 본 방법은 최악의 경우라도 임계 지연시간이 증가하지 않음을 보장해 준다(본 방법과 redundancy 제거법과의 차이점, [예제 5] 참고).

위의 두가지 언급중 첫번째를 더 자세히 설명하면 임계경로 P1의 side input들상의 fault들은 P1의 부경로 P2에 대하여서는 P2-path redundant 하더라도 다른 경로 P3에 대하여서는 P3-path irredundant할수 있다는 것이다. 따라서 이 경우 redundancy는 존재하지 않고, redundancy 제거방법을 무조건 적용하면 전체회로의 Boolean 함수가 변화됨으로 원래의 회로와 등가인(equivalent) 회로를 얻을수 없다. 이와같은 경우에는 다음과 같이 회로의 일부를 복제(duplication)하는 것이 필요하다.

원래의 조합회로에서 임계경로를 CP라고 하자. 또 이 CP의 side input들중 CP의 부경로인 P1에 대하여 P1-path redundant하나 다른 경로 P2에 대하여 P2-path irredundant한 connection이 존재한다고 하자. 이때 경로 P1과 경로 P2에서 공통적으로 존재하는 경로를 P12라고 하고 이 P12를 구성하는 부회로를 Ns12라고 했을때, 이 Ns12를 복제하여 Ns12'를 만들고 경로 P2에서 경로 P12의 ending connection과 연결되는 connection을 Ck라고 하면 이 Ck와 Ns12의 출력과의 연결을 끊고 Ns12'의 출력과 연결시킨다. 이와같은 복제를 행함으로써 발생하는 효과는 임계경로 CP의 부경로 P1에 대하여 P1-path redundant한 side input을 원래 조합회로의 Boolean 함수를 변화시키지 않으며 제거할수 있게 됨으로서 임계경로 지연시간을 줄여줄수 있다는 것이다.

[예제 5]: 그림 4. a)의 회로에서 임계경로는 CP1=(a, d, d1, B, e, e1, D, f2), CP2=(b, d, d1, B, e, e1, D, f2), CP3=(a, d, d2, g, g1, D, f2), CP4=(b, d, d2, g, g1, D, f2)들이며 임계경로 지연시간은 3이다. 또한 이 회로는 100% single stuck at fault testable한—redundancy가 없는—회로이다. 따라서 redundancy 제거방법으로는 회로를 전혀 변환시킬수 없다. 그러나 본 방법으로는 다음과 같이 임계 지연시간을 줄일수 있다. 우선 CP1과 CP2상의 지연시간을 줄이기 위하여 이들의 side input들의 path redundancy를 조사하면 c1의 SA1이 CP1과 CP2의 부경로인 P1=(c1, B, e, e1, D, f2)에 대하여 P1-path untestable하다. 그러나 c1의 SA1은 경로 P2=(c1, B, e)에 대하여서는 P2-path testable 하므로 회로의 일부 복제가 필요하다. 이 경우 P12=(c1, B, e)이고 Ns12는 connection d1과 c1을 입력으로 하고 e를 출

력으로 하는 gate B이다. 그림 4. b)는 위의 방법에 따라 회로의 일부가 복제된 것을 보여준다. 이 경우 c1의 SA1은 경로 P3=(c1, B, e, D, f2)에 대하여 P3-path untetsable하며 이 redundancy를 제거하더라도 전체 회로의 Boolean 함수는 변하지 않는다. 이 redundancy를 제거한 회로가 그림 4. c)이며 이 경우 두 경로 CP1과 CP2상으로의 지연시간은 1만큼 줄어든 2가 된다. 그러나 아직 다른 두 경로 CP3과 CP4상으로의 지연시간은 3이므로 다음으로서는 이들의 지연시간을 줄여주기 위하여 이들의 side input들의 path redundancy를 조사하여야 한다. 이 경우도 회로의 부분 복제가 필요하며 그림 4. d)가 그것이다. 여기서 c2의 SA0가 path redundant하며 이를 제거한 최종회로가 그림 4. e)의 회로이다. 이 회로의 임계경로 지연시간은 2이므로 초기회로보다 1이 줄어든 것을 알수있다.

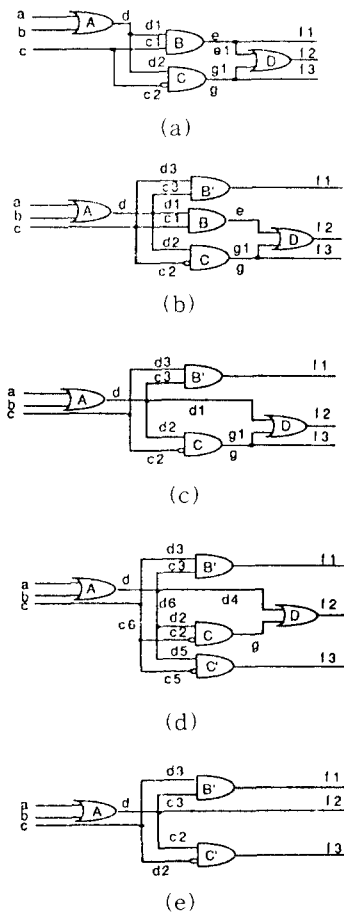


그림 4. 부회로의 복제가 필요한 회로
 Fig. 4. The circuit that needs the duplication of sub-circuit.

[정리 1]: Statically sensitizable path Ps상의 connection들에서 발생할수 있는 SA0이나 SA1 fault들은 모두 Ps-path testable하다.

[증명]: Statically sensitizable path의 정의에 의하면 event가 Ps를 따라서 propagation될때 Ps의 모든 side input들을 noncontrolling value로 하는 input cube Ic가 존재한다. 즉 Ic를 가하면 Ps의 모든 side input들은 Cs에 0이나 1이 오는것에 상관없이(즉, Cs에 가해지는 value에 독립적으로) 늘 noncontrolling value를 갖는다. 또한 Ic가 가해지면 경로 Ps상의 모든 connection들을, Cs에 0이나 1 값을 가함으로써 0이나 1로(즉, Cs에 가해지는 value에 종속적으로) 값을 줄수있다. 따라서 Ps의 임의의 connection Ci에서 발생한 SA0(SA1) fault는 Ic중 Ci를 1(0)로 하는 input cube에 의하여 fault excitation(Ci에 D 또는 \bar{D} 발생)이 되며, 동시에 이 input cube에 의해서 모든 side input들이 noncontrolling value를 가지므로 excited fault는 경로 Ps를 따라서 Ps의 ending connection으로 fault propagation(외부출력에 D 또는 \bar{D} 가 전달)되므로 검출되어진다. 따라서 Ps의 모든 connection상의 SA0와 SA1 fault는 Ps-path testable하다.

[정리 2]: Statically sensitizable path Ps의 side input상의 fault는 side input에 대한 Ps의 부회로를 Psb라고 하면, 이 side input을 입력으로 하는 gate의 noncontrolling value와 일치하는 stuck-at 고장만이 Psb-path untetsable 할 수있다. 바꾸어 말하면 controlling value와 일치하는 stuck-at 고장은 항상 Psb-path testable하다.

[증명]: Ps의 side input에서 controlling value와 일치하는 stuck-at 고장이 Psb-path testable하기 위해서는 fault excitation과 Psb-path를 통하여 fault propagation이 가능한 input cube가 존재하면 된다. 그런데 Ps를 statically sensitizable하게 하는 input cube Ic중 하나가 그것이다. 즉, Ic를 가하면 해당 side input은 noncontrolling value로 됨으로서 fault excitation이 되며 Ic중 이 side input을 입력으로 하는 gate Gi로 들어오는 Ps에 포함되는 connection을 Ps의 starting connection에다 0또는 1을 줌으로서 이 Gi의 noncontrolling value로 만들 수 있다. 따라서 Gi의 출력으로 D 또는 \bar{D} 가 발생하며, Ic는 Ps의 모든 side input들을 이미 noncontrolling value로 만들었으므로 Gi의 출력에 있는 D 또는 \bar{D} 는 Psb-path를 따라서 Psb의 ending connection에 나타나게 된다. 따라서 이 fault는 Psb-path testable하다.

[정리 3]: Statically sensitizable path Ps의 connection들과 Ps의 side input들인 connection들중 P-path redundant, 또는 P_{sb}-path redundant한 것들이 존재하는 경우(P_{sb}는 side input에 대한 Ps의 부경로), 이들을 제거하면 Ps의 길이만 감소하며 이 경로 Ps는 회로상에서 소멸하지 않는다.

[증명]: 정리 1에 의하면 Ps의 모든 connection들상의 SA0와 SA1은 Ps-path testable하므로 제거할 수 없다. 다만 정리 2에 의하여 Ps의 side input들중 이 side input의 noncontrolling value와 일치하는 stuck-at 고장만이 P_{sb}-path untestable 가능하므로 이 경우에만 이 redundant side input을 제거할 수 있다. 그러나 이 side input의 redundancy는 noncontrolling value와 일치하는 redundancy이므로(예로, AND(NAND) gate에서는 input SA1 untestable, OR(NOR) gate에서는 input SA0 untestable) 이를 제거하더라도 경로 Ps는 해당 gate만이 빠지는 것으로만 변형이 이루어지며 회로에서 소멸될 수 없다.

2. 타이밍 최적화시 statically unsensitizable path들의 영향과 대책

조합논리회로에서 존재하는 모든 경로들중에는 일반적으로 statically unsensitizable path들이 존재할 수 있다. 타이밍 최적화시, 이들 statically unsensitizable path들을 고려하지 않으면 타이밍 최적화 과정을 거친 결과회로의 임계경로 지연시간이 초기회로의 임계경로 지연시간과 비교하여 전혀 줄어들지 않거나, [5] 최악의 경우 길어질 수도 있다. [9] 따라서 타이밍 최적화되어진 결과회로의 임계경로 지연시간이 원회로의 그것보다 커지는 것을 방지하기 위해서 타이밍 최적화시 statically unsensitizable path들을 체계적으로 고려하여 제거하는 것이 필요하다.

[정리 4]: 경로 Pf가 statically unsensitizable path이고 경로 Pf의 starting connection에서 경로 Pf이외의 경로로 fan out되는 connection이 없으면 (즉, starting connection에서의 |fan-out|=1 이면) starting connection상의 SA0와 SA1 fault 모두는 Pf-path untestable 하다. 또한 starting connection에서의 |fan-out| > 1인 경우에는 이 starting connection과 연결되고 경로 Pf에 포함된 fan out connection상의 SA0와 SA1 fault 모두는 Pf-path untestable 하다.

[증명]: 경로 Pf가 statically unsensitizable path가 되기 위해서는 이것의 정의에 의하면 Pf의 side input들을 모두 noncontrolling value로 할때, Pf의

표 1. Path-based 타이밍 최적화 알고리즘 흐름도
Table.1. The flow of Path-based timing optimization algorithm.

```

min_cp.delay(init_ckt, timing_const, area_const)
:
do locate the unchecked longest path P1:
  if(P1 is a statically false path)
    : let the starting connection of P1 be Cs
      if(there exists other path P1 from Cs, and
        SA0 and SA1 is P1 path testable)
        : duplicate sub circuit:
          /* let the fan out connection of Cs which */
          /* is included in P1 be Cfo: */
          assign the appropriate 0 or 1 to Cfo and
          remove the redundancy:
          :
        else
          : assign the appropriate 0 or 1 to Cs and
            remove the redundancy:
            :
          :
        :
      :
    : identify all side inputs of P1:
    do:
      Si=an unexamined side input whose gate has the largest delay:
      if(Si is P1 path redundant)
        /*P1 is the sub path of P1 w.r.t Si.*/
        : if(there exists P2 path such that Si
          is P2 path irredundant)
          duplicate sub-circuit:
          remove the redundancy:
          :
        if( resulting circuit violates the area constraint)
          disregard the resulting circuit and
          restor circuit:
          :
        while(all side inputs are not examined, and
          current P1 is still the longest path)::
          while(timing const is not met or
            no more improvement):
          :
  :

```

starting connection Cs을 제외한 외부입력 변수들중 하나 이상에서 inconsistency가 발생하여야 한다. 그러나 |fan-out|=1 일때 Cs의 SA0나 SA1가 Pf-path testable하거나, |fan-out| > 1 일때 fan out connection Cfo가 Pf-path testable하기 위해서는 excited fault를 Pf-path를 통하여 propagation시켜야 하므로 Pf의 모든 side input들은 noncontrolling value가 되어야 한다. 즉, 이들 fault들은 경로 Pf로

fault propagation이 불가능하므로 모두 Pf-path untestable하다.

정리 4에 의하여 statically unsensitizable path Pf가 회로내에 존재하는 경우 이를 다음 두경우를 고려하여 체계적으로 제거할 수 있다. 첫째로는, 이 경로 Pf외의 경로를 통하여서도 testable 하지않는 경우에는, 경로 Pf의 starting connection에 0나 1중 적당한 값을 정하여 redundancy 제거 규칙에 따라 회로를 간소화하여 statically unsensitizable path Pf를 제거할 수 있다. 둘째로는, starting connection상의 SA0와 SA1 fault가 경로 Pf를 통하여서는 untestable하나 다른 경로 P'를 통하여서는 testable한 경우에는, III장에서 설명한 것과 같이 경로 Pf와 경로 P'에서 공통적으로 존재하는 경로를 Pc라 하고 이 Pc를 구성하는 부회로를 Ncs라고 했을 때, 이 Ncs를 복제하여 Ncs'를 만들고 경로 P'에서 경로 Pc의 ending connection과 연결되는 connection을 Ck라고 하면 이 Ck와 Ncs의 출력과의 연결을 끊고 Ncs'의 출력과 연결시킨다. 이와같은 복제를 행함으로써 원회로의 starting connection에서 새로운 fan out connection들 두개가 생기는데 이 fan out connection들중 경로 Pf에 포함되는 connection상의 SA0와 SA1 fault는 untestable하게 되어지므로 이 connection에 0나 1중 아무것으로나 값을 정하여 회로를 간소화하면 이 경우에도 statically unsensitizable path Pf를 제거할 수 있다.

본 장에서 설명된 방법과 정리들을 이용하면 statically unsensitizable path들이 존재하는 어떠한 임의의 회로에서도 이들을 체계적으로 회로내에서 제거될 수 있다. 또한 이들 경로들은 타이밍 분석시 정확한 타이밍 분석을 매우 어렵고 많은 CPU 시간을 소모하게 만드는데 이들을 타이밍 최적화 과정시 제거함으로써 정확한 타이밍 분석을 효과적으로 빠른 시간에 할 수 있게 한다.

IV. 알고리즘

본 장에서는 전장 까지에서 설명된 방법을 바탕으로 타이밍 최적화 알고리즘을 정식으로 기술한다. 표 1은 전체 알고리즘의 pseudo code이다.

조합논리회로의 타이밍 최적화를 위하여서 회로의 면적은 일반적으로 증가될수 있다는 것은 이미 잘 알려진 사실이다.^{13,14} 본 논문에서 제안된 방법도 위에서 설명한 바와 같이 부분회로의 복제 과정시 면적이 늘어난다. 그러나 이렇게 함으로서 노출된 untestable fault를 제거하는 과정에서 redundant 부분이 제거되

므로 최종회로의 면적이 부분회로의 복제전보다(초기 회로보다) 항상 증가하는 것이 아니고 때로는 감소할 수 있다.

V. 실험 및 결론

실험으로서 본 논문에서 제안된 알고리즘을 다음과 같은 회로들에 적용시켜 보았다. 첫번째 회로는 CLN1(greater-than)과 CLN2(equal)이 OR gate로 합쳐진 greater-than-or-equal 2-bit comparator 이고, 두번째 회로는 참고문헌 [6]에서 쓰여진 회로를 약간 변형한 회로이며, 세번째 회로는 4 bit carry skip adder로 이는 4-bit ripple carry adder와 하나의 four-input AND gate, 네개의 EXOR, 그리고 하나의 2 to-1 MUX로 구성된 회로이다. 이들 회로에서 gate 내에 있는 숫자는 gate의 지연시간으로 two input simple gate들은 이를 1로 하고 나머지를, EXOR, four-input AND, 2-to-1 MUX는 2로 하였다.

이들 회로를 가지고 타이밍 최적화를 수행하여 얻은 결과가 표 2에 있다. 표 2는 이들 회로의 타이밍 최적화 전후의 임계경로 지연시간(임계지연), 면적을 상대적으로 비교하기 위해서 회로의 모든 gate들을 two-input gate만 사용하여 구성했을때의 equivalent two-input gate의 수(gate2), 그리고 undetectable stuck at fault들의 수(undetected)를 보여준다. 이들의 결과로써 언급할 점은 첫째, 두번째와 세번째 회로에는 statically unsensitizable path들이 존재하는데, 본 알고리즘이 이들을 체계적으로 제거하여 결과회로를 얻음으로서 결과회로의 임계경로 지연시간이 초기 회로의 그것보다 증가하지 않았다는 것이며, 둘째, 초기회로중 첫번째와 세번째 회로에는 undetectable stuck-at 고장들이 존재하나 타이밍 최적화를 거친 결과회로는 undetectable stuck-at 고장이 존재않는 100% stuck at fault testable 회로들을 얻었다는 것이다. 사실 본 알고리즘을 조금만 확장한다면 타이밍 최적화된 결과회로로 항상 100% stuck-at 고장 testable 회로를 얻는것을 보장할 수 있으며 이는 어려운

표 2. 실험결과
Table.2. Experimental results.

회로	최적화 전			최적화 후		
	임계지연	gate2	undetected	임계지연	gate2	undetected
2 bit comparator	5	13	4	3	5	0
참고문헌[6] 회로	5	9	0	4	8	0
4 bit CSA	12	34	2	9	39	0

일이 아니다. 또한 제안된 알고리즘은 회로의 크기가 큰 경우에도 효과적으로 적용될수 있는데 이는 알고리즘에서 가장 큰 수행시간을 요구하는 P-path redundancy를 알아내는 과정이 이미 언급한 바와 같이 특정한 "단일" 경로에 대하여서만 적용이 되므로 testing의 일반적인 redundancy 검사와는 달리 많은 CPU 시간이 요구되지는 않기 때문이다.

결론으로서, 본 논문에서는 조합논리회로의 합성시 임계경로 지연시간을 최소화하는 새로운 타이밍 최적화 알고리즘을 제안하였다. 즉 논리회로의 stuck-at 고장 검출시 쓰여진 redundancy 개념을 확대하여 P-path redundancy 개념을 도입하고 임계경로의 side input들중 P-path redundant connection들을 제거하여 임계경로 지연시간을 최소화하며, 또 경로들을 statically sensitizable path와 statically unsensitizable path로 구분하여 statically unsensitizable path들을 체계적으로 제거함으로써 기존의 타이밍 최적화 알고리즘들이 범할 수 있는, 결과회로의 임계경로 지연시간이 초기회로의 임계경로 지연시간과 비교하여 개선이 없되거나 최악의 경우 커지는 것을 방지할 수 있음을 보였다. 본 방법의 또다른 장점으로서는 논리회로 합성시 technology independent 단계에서 뿐만 아니라 technology mapping이 끝난후에 이를 적용할 수 있다는 것인데, 즉 이때 회로가 전면적으로 변환(global modification)되지 않기때문에 technology mapping을 다시 할 필요가 없이 타이밍 최적화된 회로를 얻을 수 있다는 것이다.

參 考 文 獻

- [1] K. J. Singh, A. R. Wang, R. K. Brayton, and A. Sangiovanni-Vincentelli, "Timing optimization of combinational logic", in *Proc. of ICCAD*, pp. 282-285, 1988.
- [2] K. C. Chen and S. Muroga, "Timing optimization for multi-level combinational networks", in *Proc. of DAC*, pp. 339-344, 1990.
- [3] K. Keutzer, S. Malik, and A. Saldanha, "Is redundancy necessary to reduce delay?", *IEEE Trans. on CAD*, vol. 10, no. 4, pp. 427-435, Apr. 1991.
- [4] P. C. McGeer, and R. K. Brayton, "Efficient algorithms for computing the longest viable path in a combinational network", in *Proc. of DAC*, pp. 561-567, 1989.
- [5] P. C. McGeer, A. Saldanha, P. Stephan, R. K. Brayton, and A. Sangiovanni-Vincentelli, "Timing analysis and delay-fault test generation using path recursive functions", in *Proc. of International Workshop on Logic Synthesis*, 1991.
- [6] D. Brand, and V. S. Iyengar, "Timing analysis using functional relationships", in *Proc. of ICCAD*, pp. 126-129, 1986.
- [7] R. K. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, and A. R. Wang, "MIS: A multiple-level logic optimization system", *IEEE Trans. on CAD*, vol. 6, no. 6, pp. 1062-1081, Nov. 1987.
- [8] A. J. de Geus, and D. J. Gregory, "The SOCRATES logic synthesis and optimization system", in "Design systems for VLSI circuits", G. De Micheli et al. (Ed.), Martinus Nijhoff Publishers, pp. 473-498, 1987.
- [9] 양세양, 김종안, "타이밍 최적화시 Unsensitizable Path의 영향", 대한전자공학회 '92 하계 학술대회 논문집, pp. 613-616, 1992.

著 者 紹 介



梁 世 陽(正會員)

1977年~1981年 고려대 전자공학과(공학사). 1983년~1985년 고려대 대학원 전자전산기공학과(공학석사). 1986년~1990年 University of Massachusetts, Amherst 전기 및 컴퓨터공학과(공학박사).

1990년~1991年 Microelectronics Center of North Carolina(MCNC), VLSI-CAD group 선임연구원. 1991年 3月~현재, 부산대학교 컴퓨터공학과 재직. 주 관심분야는 logic synthesis, testing, high-level synthesis 및 VLSI 설계 등임.



洪 鳳 喜(正會員)

1982年 서울대 컴퓨터공학과 졸업(공학사). 1984年 서울대 대학원 컴퓨터공학과 졸업(공학석사). 1984年 서울대 대학원 컴퓨터공학과 졸업(공학박사). 1987년부터 부산대 컴퓨터공학과 조교수로 재직중, 주

관심분야는 CAD framework, CAD 데이터 모델링, 객체 중심 시스템 등임.