

수행시간을 개선한 CPLD 기술 매핑 알고리즘 개발

윤충모*, 김장옥*, 김재진**, 박남서***

*서일대학 정보통신과

**극동정보대학 전산정보처리과

***청주대학교 전자공학과

e-mail:gildong@somewhere.sck.ac.kr

Development of CPLD Technology Mapping Algorithm Improving Run-Time

Choong-Mo Youn*, Jang-Ok Kim*, Jae-Jin Kim**, Nam-Seo Park***

*Dept of Information and Communication, Seo-il College

**Dept of Computer Information Process, Keuk-dong College

***Dept of Electronic Engineering, Chong-ju University

요약

본 논문은 시간 제약 조건하에서 수행 시간을 개선한 CPLD 기술 매핑 알고리즘을 제안하였다. 제안된 기술 매핑 알고리즘은 주어진 시간 제약 조건을 고려하여 가장 빠른 시간에 기술 매핑을 수행 할 수 있도록 속도의 개선에 중점을 두었다. 입력된 회로를 DAG로 표현한 후 입력부터 출력의 방향으로 노드들을 검색하여 매핑 가능 클러스터를 생성한다. 생성된 매핑 가능 클러스터들 중에서 시간 제약 조건에 적합한 매핑 가능 클러스터를 선택하여 기술 매핑을 수행함으로써 전체 수행 시간이 다른 알고리즘에 비해 빠르게 수행되는 결과를 나타내었다.

1. 서론

FPGA(Field Programmable Gate Array)와 CPLD(Complex Programmable Logic Device)는 디지털 회로 구현에 널리 사용되고 있다. 대부분의 FPGA는 LUT(Look up table)라는 기본 논리 블록(block)으로 구성되어 있으며, 몇몇의 FPGA의 경우 멀티플렉서(multiplexer)를 기본 구조로 구성되어 있는 것도 있다. 반면에 CPLD는 PLA(Programmable Logic Array) 또는 PLD(Programmable Logic Device) 형태의 기본 구조를 가지고 있다.[1][2]

기존에 제안된 기술 매핑 알고리즘들은 대부분 LUT 구조에 맞도록 제안되었으며, 면적 최소화와 지연시간 최소화에 중점을 두고 개발되었다.[3][4]

이에 반해 CPLD의 경우 DDMAP, TEMPLA, TMCPLD, TMCPLD-II 등 극소수의 기술 매핑 알고리즘만이 제안되어 있다. 이러한 알고리즘들 중에서 주어진 시간 제약 조건을 고려한 기술 매핑 알고리즘으로는 TMCPLD와 TMCPLD-II 만이 제안되

어 있다. 그러나 이러한 알고리즘들은 면적 최소화 에 중점을 두어 수행 시간이 길다는 단점이 있다.

따라서 본 논문에서는 수행 시간을 개선한 CPLD 기술 매핑 알고리즘을 제안하였다.

2. 배경

디지털 회로의 조합논리 부분은 DAG(directed acyclic graph)로서 표현될 수 있다. DAG에서 각 노드(node)는 회로에서 하나의 연산자 만을 가지고 있다. 노드로 표현된 논리 함수는 SOP(Sum Of Product)의 형태로 재구성된다.

CPLD를 구성하고 있는 논리 블록은 여러 형태의 입력수와 서로 다른 형태의 논리 함수로서 구성된다. PLA 형태의 블록은 프로그램 가능한 AND-어레이(array)와 OR-어레이로 구성되어 있다. 이러한 형태의 논리 함수를 최소화하기 위해서는 2단 구조(two-level)의 최소화를 수행하여야 한다. 대다수의 CPLD는 많은 수의 PLA 형태를 가진 논리 함수들

로 구성되어 있다.

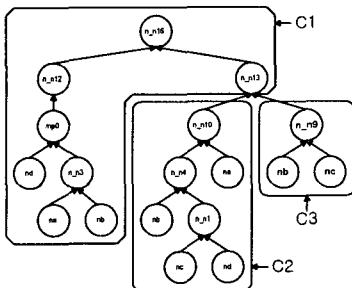
따라서 큰 회로를 하나의 CPLD로 구성하기 위해서는 2단 구조의 최소화를 수행하여야 한다. 또한 면적을 줄이기 위해 다단(multi-level)구조의 합성을 고려하여야 한다.

3. 문제 정의

주어진 조합 논리 회로는 DAG의 형태로 $G=(V, E)$ 로 표현된다. 각각의 노드 $v \in V$ 는 하나의 출력을 가진 논리 함수로 구성된다. 노드 $u \in V$ 의 출력이 되는 에지 $(u, v) \in E$ 는 노드 $v \in V$ 의 입력을 의미한다. V 를 구성하고 있는 노드의 집합(set)들 중에서 출력 에지(out-degree)가 0인 노드를 최종 출력(PO:primary output)이라 정의하고, 입력 에지의 수가 0인 노드를 최초 입력(PI:primary input)라 정의한다.

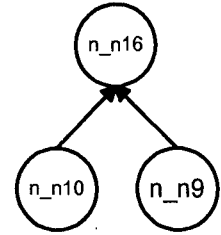
서브 그래프(subgraph)는 $C=(V', E')$ 로 정의하고, $V' \subset V$ 이고 $E' \subset E$ 이다. C 는 클러스터(cluster)라고 정의한다. V' 안의 모든 노드들을 병합(merge)하여 생성된 논리 함수를 클러스터 함수(cluster function)라 정의한다. 하나의 노드가 만약 CPLD의 기본 블록인 CLB(Common Logic Block)에 매핑(mapping)될 수 있다면 이 노드는 매핑 가능(feasible) 노드로 정의한다. 불린 네트워크를 구성하고 있는 각각의 노드가 매핑 가능 노드이면 이러한 불린 네트워크는 매핑 가능 네트워크라고 정의한다. 또한, 생성된 클러스터가 하나의 CLB에 매핑 가능하면 이 클러스터는 매핑 가능 클러스터(FC : feasible cluster)라 정의한다.

클러스터 C 가 만약 하나의 노드만을 가지고 있다면 이러한 클러스터를 트리비얼(trivial)이라 정의한다. 각각의 노드들이 가지고 있는 OR팀의 수를 초기 비용이라 정의한다. 또한, 매핑 가능 클러스터의 전체 OR 팀수를 전체 비용으로 정의한다.



(a) 모든 매핑 가능 클러스터

$$\begin{aligned}
 n_{n16} &= n_{n12} + n_{n13} \\
 n_{n12} &= np0 \\
 np0 &= nd * n_{n3} \\
 n_{n3} &= na + nb \\
 n_{n13} &= n_{n10} + n_{n9} \\
 n_{n10} &= n_{n4} + na \\
 n_{n4} &= nb + n_{n1} \\
 n_{n1} &= nc + nd \\
 n_{n9} &= nb + nc
 \end{aligned}$$



(b) 노드와 클러스터 함수 (c) 매핑 결과

그림 1. 본 논문의 기술 매핑에 대한 예제

Fig. 1 An example of technology mapping by our approach

예를 들어 그림 1에서 C1, C2, C3는 매핑 가능 클러스터이다. 매핑 가능 클러스터의 생성은 CLB의 OR 팀수를 초과하지 않는 범위내에서 가장 긴 경로(critical path)를 우선으로 생성하게 된다.

그림 1의 (c)의 결과는 (a)에서 생성된 클러스터를 매핑한 결과이다.

따라서 주어진 불린 네트워크를 주어진 시간 제약 조건하에서 수행 시간을 개선하기 위한 조건과 정의는 다음과 같다.

정의 1 : 매핑 가능한 클러스터의 집합 $S = \{C_1, \dots, C_n\}$ 는 주어진 불린 네트워크 $G = (V, E)$ 에서 다음의 2가지 조건을 고려하여 생성하여야 한다.

다단의 수 제한(number of multilevel constraint) : 다단의 수는 주어진 CPLD에서 시간 제약 조건과 지연 시간을 고려하여 생성한다. 다단의 수를 구하기 위한 수식은 (식 1)에 나타내었다.

$$\text{다단의 수}(mn) = \frac{\text{시간제약 조건}}{\text{소자의 지연시간}} \quad (\text{식.1})$$

$$\begin{aligned}
 \text{소자의 지연 시간} &= \\
 &T_{IO} + T_{CO} + T_{GA} + T_{RO}
 \end{aligned}$$

- T_{IO} : 입출력 핀에서 CLB까지의 지연 시간
- T_{CO} : 조합 논리 회로 출력까지의 지연 시간
- T_{GA} : 클럭 스큐(skew)
- T_{RO} : 레지스터 출력의 지연 시간

OR 팀수 제한 : 회로 구현을 위해 지정한 CPLD를 구성하고 있는 CLB의 OR 팀수를 초과하지 않는 범위내에서 클러스터를 생성하여야 한다.

본 논문은 주어진 불린 네트워크를 시간 제약 조건을 고려하여 매핑 가능한 클러스터들을 생성하여 매핑하는 것이다. 따라서 본 논문에서 제안한 알고리즘은 여러 단계로 나누어 수행된다. 첫 번째 단계로 매핑 가능한 클러스터를 생성하고, 커버링(covering)을 수행하는 그래프 커버링 단계이고, 두 번째 단계는 CPLD의 구조에 맞도록 PLA 형태의 부울식을 생성하는 콜랩싱을 수행한다. 마지막 세 번째 단계는 주어진 CPLD의 기본 블록인 CLB의 구조에 맞도록 분할하고 매핑하는 빈-패킹 단계를 수행한다.

4. 수행 시간을 개선한 CPLD 기술 매핑

4.1 단계 I : 그래프 커버링

불린 네트워크가 주어지면 첫 번째 단계로 그래프 커버링을 수행한다. 그래프 커버링은 매핑 가능한 클러스터 생성과 커버링으로 분류하여 수행된다. 불린 네트워크이 주어지면 우선 출력의 수가 2이상인 노드들을 검색한다. 검색된 노드들을 최종 출력으로 선정하여 CLB의 최대 OR터수인 k를 초과하지 않는 전체 비용을 갖는 매핑 가능 클러스터를 생성한다.

전체 비용을 계산 하기위한 수식은 (식 2)에 나타내었다.

$$Cost(Node) = \begin{matrix} k(Child_Node_Cost(Node)) & Node_Cost=1 \\ \sum Child_Node_Cost(Node) & Node_Cost \geq 2 \end{matrix} \quad (식 2)$$

매핑 가능 클러스터가 생성되면 주어진 시간 제약 조건에 맞도록 불린 네트워크를 커버링한다.

본 논문에서 제안한 알고리즘은 그림 2에 나타내었다.

Procedure BN_cover(N, k, mn, FC)

```

N : (V, E)
k, mn : 상수
FC : 매핑 가능한 클러스터들의 집합
begin
    Subgraph_Level = 0;
    foreach v ∈ PI of N do
        Re(v) = { }, PO(v) = { }
    end
    put PO(v) ⊆ N(|n_out| ≥ 2)
    foreach v ∈ V in PO(v) do
        while((mn-2) ≥ Subgraph_Level){
            while(k > FC_cost){
                Re(v) = ∪ FC(v)
            }
        }
    }
end BN_cover

```

```

}
Subgraph_Level ++;
}
end
end BN_cover

```

그림 2. 그래프 커버링 알고리즘
Fig. 2 Graph covering algorithm

그림 2에서 $Re(v)$ 는 불린 네트워크 커버링의 결과이다. 주어진 k 값보다 작거나 같은 비용을 가지고 있는 FC들을 $Re(v)$ 에 저장한다.

그러나 주어진 시간 조건하에서 가장 좋은 결과를 나타낼 수 있는 FC를 생성하기 위해 자식 모드를 포함할 수 있는 클러스터를 생성하여 매핑 한다. “다단의 수-2”만큼의 단수를 제한하는 것은 단수의 수가 많을 경우 주어진 시간 제약 조건에 맞도록 회로를 구성하기 위해 PO를 포함하고 있는 여러 단의 FC를 2단의 단수로 재구성해야 하기 때문이다.

4.2 단계 II : 콜랩싱(Collapsing)

단계 I의 그래프 커버링에서 선택된 FC는 부울식으로 변환하여야 한다. 생성된 클러스터는 CPLD가 PLA 또는 PLD 형태로 구성되어 있으므로 이러한 형태에 맞도록 AND-OR의 형태로 부울식을 생성한다. “다단의 수-2”의 조건 안에 생성된 FC는 FC가 가지고 있는 함수에 맞도록 부울식을 생성하고, 그 외에 PO를 포함한 FC와 나머지 FC는 모두 합하여 AND-OR 형태의 부울식을 생성한다.

4.3 단계 III : 빈 패킹(Bin-packing)

마지막 단계는 콜랩싱에서 생성된 부울식을 주어진 CPLD의 CLB에 맞도록 재구성하는 단계이다. “다단의 수-2”의 조건안에서 생성된 FC는 하나의 CLB로 구현 가능하므로 생성된 부울식을 직접 매핑하면 된다. 그러나 PO를 포함한 나머지 FC에서 생성된 부울식은 하나의 CLB에 구현될 수 없으므로 CLB의 내부 구조에 맞도록 재구성 되어야 한다. 따라서 PO를 포함한 나머지 FC들에서 생성된 부울식을 단수가 2가 되도록 재구성하는 과정을 단계 III인 빈 패킹에서 수행한다.

본 논문에서는 기존에 제안된 CPLD 기술 매핑 알고리즘들과 비교를 위해 같은 소자인 MACH4를 구현 대상 소자로 선정하였다. MACH4의 OR 터수는 최대 20개이다.

5. 실험 결과

본 논문에서 제안한 알고리즘을 MCNC 벤치마크에 적용한 결과 기존에 제안된 DDMAP과 TEMPLA보다 수행 시간과 면적이 감소된 결과를 나타내었다.[7][8] 또한 TMCPLD에 비해 면적은 증가되었으나 수행 시간이 감소된 결과를 나타내었다. TMCPLD-II는 TMCPLD에 비해 수행 시간이 느리므로 비교 대상에서 제외하였다.

실험 결과는 표 2에 나타내었다. 표 2에서 첫 번째 행은 벤치마크 회로의 이름을 나타내고, 두 번째 행은 TEMPLA를 이용하여 구현된 CLB의 수와 수행 시간을 나타내었다. 세 번째 행은 TMCPLD를 이용한 결과로서 CLB의 수와 수행시간을 나타내었다. 마지막 네 번째 행은 본 논문에서 제안한 TMCPLD-III의 결과를 나타내었다.

실험 결과 TEMPLA에 비해 CLB의 수는 9.13% 감소되었고 수행 시간은 8.03% 감소되었다. TMCPLD에 비해서는 CLB의 수가 8.63% 증가되었지만 수행 시간은 14.08% 감소된 결과를 나타내었다.

표 2. TMCPLD-III와 기존의 기술 매핑 알고리즘과의 결과 비교

Table 2. Result comprison of the existing technology mapping and TMCPLD-III

	TEMPLA		TMCPLD		TMCPLD-III	
	block	running time (sec)	block	running time (sec)	block	running time (sec)
alu4	155	29.3	81	33	85	28.7
cps	120	18.3	119	22.1	121	18.2
apex4	193	30.2	139	50.6	152	28.1
misex3	154	27.8	147	32.8	153	27.6
ex5p	132	18.8	132	33.5	136	18.9
s38417	603	495.2	479	502.7	538	432.5
seq	229	52.8	219	54.1	227	51.3
fir	249	123.3	199	128.1	231	118.9
fsm8_8_13	49	5.5	49	5.8	52	5.2
pmac	237	126.6	232	131.3	232	124.7
psdes	151	37	119	39.4	136	34.3
sort	138	29.8	101	31.2	127	26.3

6. 결론 및 추후 연구

본 논문은 수행 시간을 개선한 CPLD 기술 매핑 알고리즘을 제안하였다. 이번 CPLD 기술 매핑 알고리즘은 그래프 커버링, 콜랩싱, 빈 패키징의 3단계로 구성되어있다.

MCNC 벤치 마크에 적용한 실험 결과 기존에 제안된 TEMPLA와 TMCPLD의 기술 매핑 알고리즘

에 비해 수행 시간이 감소되는 결과를 나타내었다.

앞으로 본 논문에서 제안한 알고리즘을 다른 종류의 CPLD에 적용할 수 있는 범용적인 형태의 알고리즘을 개발하고자 한다.

참고문헌

[1] *The Altera Data Book*, Altera Corporation, 1996

[2] *ACT 1 Series FPGAs Data Sheet*, Actel Corporation, 1996

[3] *The MACH 4 Family Data Sheet*, Advanced Micro Devices, 1996

[4] J. Cong and Y. Ding, "FlowMap : An 'Optimal Technology Mapping Algorithm for Delay Optimization in Lookup-Table Based FPGA Designs", *IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems*, Vol. 13, No. 1, January 1994, pp. 1-11

[5] R.J Francis, J. Rose and Z. Vranestic, "Chortle-crf : Fast Technology Mapping for Lookup Table-Based FPGAs", *28th ACM/IEEE Design Automation Conference*, June 1991, pp.227-233.

[6] R.J Francis, J. Rose and Z. Vranestic, "Technology Mapping of Lookup Table-Based FPGAs for Performance", *1991 IEEE Conference on Computer Aided Design*, pp. 568-571

[7] E. M. Sentovice et al., "SIS : A system for sequential Circuit Synthesis", *Technical Report UCM/ERL M92/41*, Electronics Research Laboratory, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1992

[8] Jason Helge Anderson, Stephen Dean Brown, "Technology Mapping for Large Complex PLDs", *Design Automation Conference*, 1998, pp. 698-703