패턴 추출을 이용한 LUT형 FPGA 합성

장준영*, 이귀상**

* 한국전자통신연구원 멀티미디어연구부, ** 전남대학교 전산학과

Logic Synthesis for LUT-Type FPGA Using Pattern Extraction

June Young Chang*, Guee Sang Lee**

* Multimedia Technology Department, ETRI

jychang@computer.etri.re.kr

** Department of Computer Science, Chonnam National University

gslee@cs.chonnam.ac.kr

요 약

In this paper, we presents a method for multi-level logic minimization which is suitable for the minimization of look-up table type FPGAs. A pattern extraction algorithm is minimized AND/XOR multi-level circuits. The circuits apply to Roth-Karp decomposition which is most commonly used technique in the FPGA technology mapping.

We tested the FPGA synthesis method using pattern extraction on a set of benchmark. The proposed method achieved reductions on the number of LUTs in mapping soultion as compared with MISII(or SIS) or previous results[5].

1. 서 론

FPGA는 ASIC 설계에서 기존의 표준 셀이나 MPGA(Mask Programmed Gate Arrays)를 급속히 대치해 가고 있다. FPGA는 임의의 조합 또는 순차 회로를 구현하는 일련의 논리 블럭들을 갖고 있으며 이러한 블럭들을 연결하는 프로그램 가능한 라우팅를 갖고 있다. 여기서 각각의 블럭의 기능 뿐만아니라 그 연결까지도 사용자가 프로그램할 수 있으며, 이로 인하여 빠르고 값싸게 초기 모형을 만들어 볼 수 있다. 최근 들어 FPGA가 빠른 시간 안에 원하는 논리 기능을 구현할 수 있다는 장점과 함께 회로 설계의 필수적인 단위로 등장하고 있다[1]. FPGA의 이러한 장점을 이용하기 위해서는 기존의 논리합성과는 다른 형태의 합성 방법이 요구된다. FPGA에서 사용되는 기본 모듈의 종류는 k-입력 LUT[1], 멀티플렉서[3], Cellular 구조[4] 등이 있다.

FPGA 합성 도구를 구현하는데 있어서 지금까지는 SOP(Sum-Of-Products)를 이용한 FPGA 합성 방법들이 많이 연구되어졌으나 패리티 타입 함수(산술, 통신, 에러 제어, 테스팅회로 등)의 구현에 이용되는 XOP(eXclusive-Of-Product) 표현을 이용한 FPGA 합성 방법에 관한 연구는 시작 단계이다. AND/XOR에 의한 표현의 단점은 그 구현이 XOR 게이트를 이용하므로 비용이 높다는 점이다. 그러나, FPGA에서는 XOR 게이트와 관계없이 구현되므로 이러한 문제점을 해결한다.

현재 SOP에 의한 FPGA 합성에 관한 연구는 1980년대 중반부터 본격적으로 활성화되어 최근에는 합성 결과에 대한 비교 기준으로 게이트의 수, 항의 수, 리터럴의 수 대신에 FPGA를 구성하는 기본 셀의 수를 이용하고 있다[1]. 최근에 발표되는 대부분의 논리합성의 결과는 FPGA 셀의 수를 기준으로 하고 있으며 이는 실제 회로를 구현하는데 직접적인 영향을 미친다. 지금까지 다양한 FPGA의 종류와 합성 방법이 발표되고 있고 이에 따른 실험 결과가 보고되고 있다[1,2,5,6]. 그러나 XOP를 이용한 FPGA 합성 방법은 최근 들어 연구가 진행되어지고 있으나 알고리즘의 빈약으로 인해 실제적으로 동작되는 합성 도구는 거의 없는 상태이다.

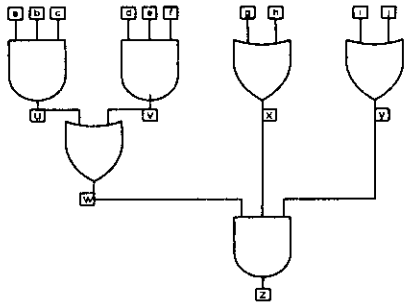
본 논문에서는 XOP 최소화를 위한 패턴 추출을 이용한 LUT형 FPGA 합성 방법을 제시한다. 패턴 추출에 의해 최소화된 다단계 회로를 구성하고 이를 Roth-Karp 분할에 적용하여 LUT형 FPGA에 매핑한다. 이 방법에 의한 실험 결과를 기존의 SOP를 이용한 FPGA 합성 방법의 결과와 비교한다.

다음 2장에서는 LUT형 FPGA 합성에 대해서 설명하고 3장에서는 다단계 회로의 최소화를 위한 패턴 추출 과정과 k-입력 LUT형 FPGA에 매핑하는 과정에 대해서 설명하고 4장에서는 벤치마크회로에 대한 실험 결과를 제시한다.

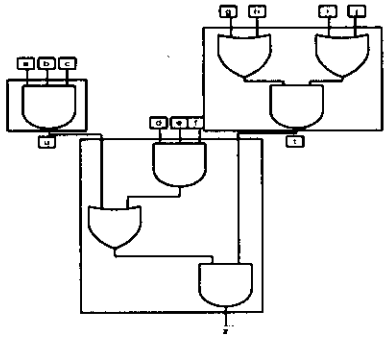
2. LUT에 의한 FPGA 합성

LUT(Look Up Table)에 의한 FPGA 합성은 입력의 개수가 k 개인 임의의 논리함수를 구현할 수 있는 논리 블록(Configuration Logic Block)을 가지고 회로를 구성하는 것이다. LUT 회로 최적화

의 목적은 회로를 구성하는 LUT 수와 LUT 레벨의 최소화이다. LUT 수를 최소화하는 것은 칩의 면적을 줄이며, LUT 레벨의 최소화는 회로의 전체적인 지연시간의 감소를 의미한다. LUT에 의한 합성 과정을 예를 들어서 설명하면 다음과 같다. 그림 1.(a)는 논리함수 z=((abc)+(def)) (g+h)(i+j)를 표현한 불리언 네트워크이고 이를 5-입력 LUT로 표시하면 그림 1.(b)와 같다. 이때 각 노드는 LUT를 나타내며 z-LUT에 의해 구현되는 함수는 z=(u+(def))t이다.



(a) 불리언 네트워크



(b) 5-입력 LUT

그림 1. 불리언 네트워크와 5-입력 LUT

기본적으로 주어진 회로를 k-입력 LUT에 매핑시킬 수 있도록 분할하며, 최종적으로 최소의 LUT로 회로가 구성될 수 있도록 기술적 매핑을 실행한다.

기존의 SOP에 의한 최소화가 산술, 통신, 에러제어 회로 등에서 좋은 결과를 얻지 못하고 또한 많은 회로에서 SOP에 의한 최소화보다 XOP에 의한 최소화가 더 좋은 결과를 얻는다는 것이 발표되었다[8]. 따라서 XOP에 의한 최소화의 결과를 FPGA 매핑에 가장 많이 사용되는 Roth-Karp 분할에 적용한 XOP에 의한 FPGA 합성 방법이 필요하게 되었다. 본 논문에서는 최소의 XOP 패턴을 추출하는 최소화 알고리즘을 제시하고 이를 LUT형 FPGA에 매핑하기위해 Roth-Karp 분할에 적용하는 과정을 제시한다.

3. 패턴 추출에 의한 LUT형 FPGA 합성

XOP에 의한 FPGA 합성 방법은 최소의 XOP 패턴을 구하기 위해 패턴 추출 알고리즘을 사용한다. 논리함수를 표현하는 PLA 화일을 읽어서 XOP 패턴 추출에 의한 추출된 패턴으로 다단계 AND/XOR 회로를 생성하고, 이를 텍스트 형태의 BLIF(Berkeley Logic Interchange Format)형태로 변환한 다음 Roth-Karp 분할에 적용하여 FPGA의 기본 셀인 k-입력 LUT에 매핑한다. 먼저, 패턴 추출 기법에 의해 최소화된 XOP 패턴으로 구성된 다단계 AND/XOR 회로를 구하는 최소화하는 방법에 대해서 설명하고 이를 입력으로하여 LUT에 매핑하는 과정에 대해서 설명한다. 먼저, 패턴 추출 알고리즘에 사용되는 용어에 대해서 설명한다.

[정의 1] 피벗 원소(Pivot element)
진리표 형태의 맵으로 표현된 논리 함수에서 패턴을 추출하기 위해 기준이 되는 원소를 피벗 원소라 한다.

[정의 2] 패턴(Pattern)
열 벡터와 행 벡터의 곱으로 구성되는 행렬을 패턴이라 한다.

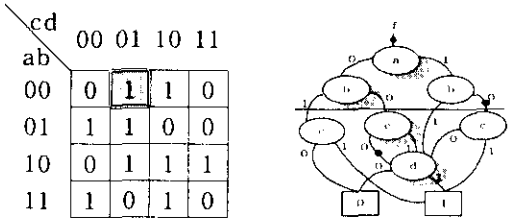
3.1 패턴 추출에 의한 최소화

최소화된 다단계 AND/XOR 회로를 생성하는 패턴 추출 알고리즘과 XOP 패턴 추출에 의해 회로가 최소화되어 가는 과정을 설명한다. BDD(Binary Decision Diagram)로 표현된 논리함수를 입력으로 하여 최소의 패턴의 구성된 다단계 AND/XOR 회로를 구성하는 패턴 추출 알고리즘에 대해서 설명한다.

```
/* 입력 : 논리함수를 나타내는 PLA 화일 */
/* 출력 : 최소화된 회로의 BLIF 화일 */
Pattern_extraction_synthesis(F) {
  for ( each output i )  {
    make_BDD(Fi)
      /* 논리함수를 표현하는 BDD생성 */
    input_partition
      /* 입력 분할 생성 */
    for ( BDD(Fi) = 0 )  {
      select_pivot_element
        /* 패턴을 생성하기위한 피벗 선택 */
      extract_pattern : BDD(Li), BDD(Ri)
        /* AND, XOR 연산에 의한
           좌, 우 패턴 추출 */
      BDD(Xi) = BDD(Li) * BDD(Ri)
      BDD(Fi+1) = BDD(Fi) ⊕ BDD(Xi))

    }
  }
  compaction_pattern(Li, Ri)
    /* 동일 패턴 제거 */
  make_blif_file(Li, Ri)
    /* blif_file 생성 */
  BDD_single_based_synthesis(Li)
  BDD_single_based_synthesis(Ri)
}
```

먼저, 주어진 논리함수가 F(a,b,c,d)라 하면 정상 입력 순서(a>b>c>d)에 의해서 BDD를 구성하고(그림 2), 입력된 BDD에 사용되는 입력 변수를 분할한다. 여기서 사용하는 입력 분할은 입력을 n/2 (입력 수: n)으로 나누는 간단한 방법을 이용한다. BDD에 의해 패턴 추출의 기준이 되는 피벗 원소를 선택하고 이를 이용하여 각 출력에 대해 독립적인 좌, 우 패턴을 생성한다(그림 3, 4). 다음으로 생성된 좌,우 패턴을 AND 연산(*) 하여 XOR 패턴을 생성하고(그림 5), 이를 원래 함수에 XOR 연산(F' = F ⊕ XOR 패턴)을 하여 원래 함수를 변화한다(그림 6). 원래 함수 F가 0이 될 때까지 반복하면서 좌,우 패턴을 생성한다. 각 단계에서 추출되는 패턴들에 의해 최종적으로 트리 형태의 최소화된 다단계 AND/XOR 회로가 합성된다.



(a) 맵 표현 (b) BDD 표현

그림 2. BDD에 의한 논리함수 표현

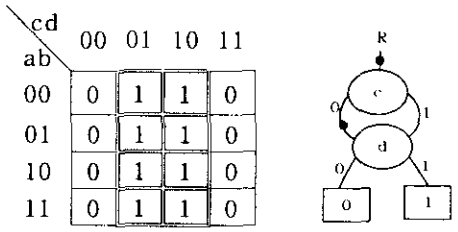


그림 3. cd 패턴 추출

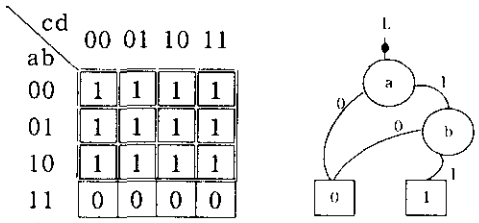


그림 4. ab 패턴 추출

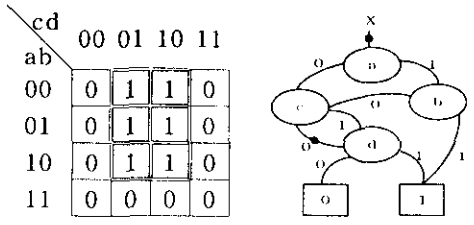


그림 5. XOR 패턴

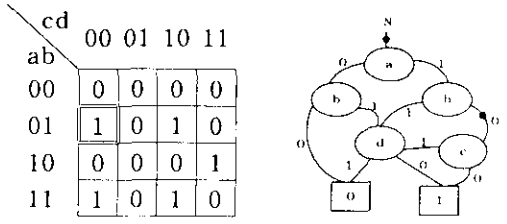
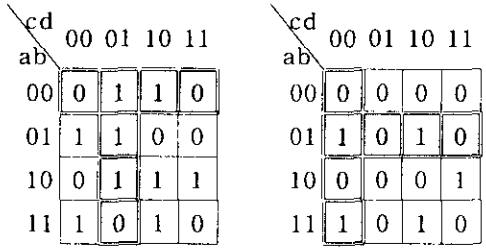
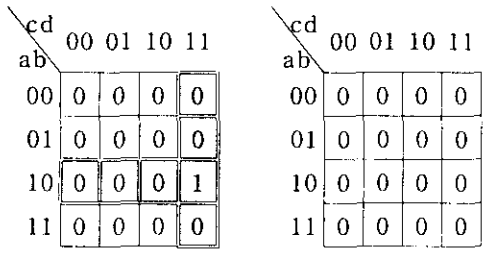


그림 6. XOR 연산에 의한 함수의 변화

따라서 일련의 피벗 원소의 선택에 의해서 논리함수를 구성하는 패턴이 추출되고 이에 따라 함수가 변화되어 간다. 이 연속적인 과정을 맵으로 표현하면 그림 7.과 같다. 그림 7.의 (a)에서 피벗 원소 a'b'c'd를 선택하고 피벗 원소와 관련된 ab 패턴은 a'b'⊕a'b⊕ab' = 1⊕ab = (ab)'이고, cd 패턴은 c'd ⊕ cd' = c⊕d이다. 이 두 개의 패턴을 AND(·) 연산하면 패턴 (ab)'·(c⊕d)가 추출된다.



(a) (ab)'·(c⊕d) 패턴 (b) b·d 패턴



(c) ab'·cd 패턴

그림 7. 패턴 추출과 논리 함수의 변화

이 추출된 패턴을 원래 함수 F에 XOR(⊕) 연산하면 그림 7.의 (b)가 된다. 따라서 추출된 3 개의 패턴에 의해서 함수가 구성되고 패턴 추출에 의한 최소화된 결과는 F = (ab)' (c ⊕ d) ⊕ bd' ⊕ ab'cd 이다.

3.2 Roth-Karp 분할을 이용한 FPGA 매핑

패턴 추출에 의해 구해진 XOR 패턴으로 구성된 다단계 AND/XOR 회로를 LUT형 FPGA 매핑의 가장 일반적인 분할 방법인 Roth-Karp 분할에 적용한다. 여기서 기본이 되는 이론은 Roth-Karp 분할이다.

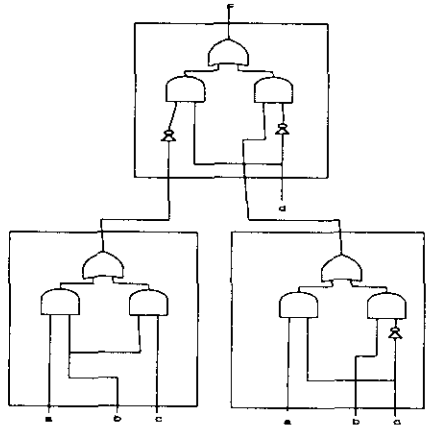


그림 8. 5-입력 LUT에 매핑

이 이론은 대규모회로를 좀 더 효과적으로 최적화될 수 있는 작은 부분회로로 분할하고 이를 FPGA의 기본 모듈인 LUT에 매핑할 때 사용되는 분할 방법이다. Roth-Karp 분할은 논리함수를 행과 열 패턴으로 분할하여 하위 단계와의 연결선의 수를 줄임으로서 LUT의 갯수와 함수를 구성하는 레벨의 수를 최소화하여 LUT에 매핑하는 방법이다. 위의 XOP 패턴 추출에 의한 결과를 최소화의

결과를 MISII의 FPGA 매핑 과정의 입력으로 사용되는 BLIF 파일로 변환하고 이를 Roth-Karp 분할 과정에 적용하여 3-입력 LUT에 매핑한 결과는 그림 8.과 같다. 따라서 3-입력 LUT 3개에 매핑됨을 알 수 있다. XOP 최소화의 결과로 구성된 블리언 네트워크를 입력으로하여 SIS에서 사용하는 xl_k_decomp -n 3를 이용하여 3-입력 LUT에 매핑한 결과이다.

4. 실 험 결 과

본 논문에서 제시한 패턴 추출에 의한 최소화 알고리즘은 IBM 586 PC의 LINUX 시스템에서 C 언어로 구현되었으며 FPGA 합성에 이용되는 Roth-Karp에 의한 분할 방법은 UC Berkely에서 개발한 MISII(or SIS) 사용하는 명령어를 이용하였다. SOP에 의한 FPGA 합성 방법에서는 최소화에 의해 구해진 논리함수를 구성하는 SOP 큐브를 생성하고 이를 Roth-Karp 분할에 적용한 반면 XOP에 의한 FPGA 합성 방법에서는 제시된 최소화 알고리즘에 의해 구해진 XOP 패턴을 Roth-Karp 분할에 적용한 결과이다.

XOP에 의한 최소화 결과를 SIS의 Roth-Karp분할에 적용한 실험 결과를 표 1. 에 제시한다. FPGA 합성 결과의 비교 기준으로는 Xilinx XC3000[7]의 기본 논리 블럭으로 사용되는 5-입력 CLB의 수를 이용한다. [5]에서는 휴리스틱 방법으로 최선의 입력 분할을 선택하여 최소화된 SOP 형태의 회로를 생성하고 이를 Roth-Karp 분할에 적용하여 CLB의 수를 최소화하는 하는 방법을 사용한다. 다음과 같은 SIS의 명령들을 사용하여 CLB의 수(노드의 수)를 계산한 결과를 제시한다.

```
xl_k_decomp -n 5
xl_partition -n 5 -tm
xl_cover
```

이때 네트워크에 있는 노드의 수는 일대일로 CLB에 직접 매핑이 가능하다. XOP_RK_decomp의 결과는 XOP의 최소화 결과를 SIS에서 제공하는 명령을 이용하여 CLB의 수를 계산한 결과이다. 실험 결과, XOP 최소화를 이용한 FPGA 합성 결과가 SOP에 의한 FPGA 합성 방법을 사용하는 SIS_RK_decomp나 [5]_RK_decomp의 결과보다 개선된 결과를 보인다. 특히, 산술, 통신, 에러 제어 회로등에 사용되는 패리티 함수(9sym, rd73, rd84)와 같은 회로에서는 매우 좋은 결과를 보인다.

따라서 XOP로 최소화된 회로의 구조가 SOP에 의한 회로의 구조보다 FPGA 매핑에서 개선된 결과를 보임을 알 수 있다. 또한 제시된 합성 방법이 합성 시간에서 매우 효과적이다. 벤치마크 회로를 합성하는데 걸리는 시간이 SIS나 [5]에서 사용하는 CPU 시간보다 매우 빠른 시간에 처리된다.

벤치마크 회로	PI/PO	SIS RK_decomp		[5] RK_decomp		XOP RK_decomp	
		#CLB	time (sec)	#CLB	time (sec)	#CLB	time (sec)
5xp1	7/10	26	6.7	26	4.9	*19	5.0
9sym	9/1	111	48.3	74	40.6	*10	5.8
clip	9/5	41	16.4	32	13.4	40	6.4
f51m	7/2	27	12.8	27	9.1	*17	0.2
misex1	8/8	19	3.5	18	2.7	*11	0.9
rd53	5/3	3	5.8	3	2.2	3	1.1
rd73	7/3	55	28.2	30	29.8	*8	1.3
rd84	8/4	129	72.4	75	35.2	*11	1.5
z4ml	7/4	5	0.4	5	0.4	10	2.0
bw	5/28	49	35.4	56	27.8	*28	2.5
sao2	10/4	72	26.2	51	20.2	*10	3.6
vg2	25/8	28	8.1	23	7.4	46	5.1
duke2	22/29	177	111.3	125	55.9	*43	11.4
misex2	25/18	32	6.5	32	6.5	*26	15.4

표 1. 벤치마크회로에 의한 실험 결과

결 론

본 논문에서는 패턴 추출을 이용한 FGA 합성 방법을 제시하였다. 패턴 추출에 의해 최소화된 XOP 패턴을 구하고 이를 Roth-Karp 분할에 적용하여 k-입력 LUT형 FPGA에 기본 셀인 CLB에 매핑한다. 실험 결과 XOP 최소화에 의한 FPGA 합성 방법이 기존의 SOP에 의한 FPGA 합성 방법보다 개선된 결과를 보인다. 특히 산술, 통신, 에러 제어등에 사용되는 패리티 형태의 회로에서 제시된 방법이 매우 좋은 결과를 보인다.

앞으로, 최선의 입력 분할을 선택하여 XOP를 최소화한 결과를 Roth-Karp 분할 방법에 적용하는 방법을 통하여 FPGA 합성 결과를 개선하고자 하며, 또한 XOP에 의한 LUT형 FPGA 합성 도구를 개발하고자 한다.

참 고 문 헌

[1] R.J.Francis., "A Tutorial on Logic Synthesis for Lookup -Table Based FPGAs," ICCAD, 1992, pp. 40-47.

[2] R.Murgai, Y.Nishizaki, N.Shenoy, R.K.Brayton and A.Sangiov anni-Vincentelli., "Logic Synthesis for Programmable Gate Ar rays," Proceedings of Design Automation Conference, Jun. 1990, pp. 620-625.

[3] R.P.Jacobi and A.M.Trullemans., "A New Logic Minimization Methods for Multilpexor Based FPGA Syntehsis," Proceedings of European Design Automation Conference, 1993, pp. 312-317.

[4] I.Schafer, M.A.Perkowski and H.Wu., "Multilevel Logic Synthesis for Cellular FPGAs Based on Orthogonal Expansion," IFIP Workshop on Applications of the Reed Muller Expansion in Circuit Design, Hamburg, Germany, Sep. 1993, pp. 42-51.

[5] Wen-Zen Shen, Juinn-Dar Huang, and Shih-Min Chao., "Lambda Set Selection in Roth-Karp Decomposition for LUT-Based FPGA technology Mapping," Design Aautomati on Conference, 1995, pp. 65-69.

[6] Ellen M. Sentovich, Kanwar Jit Singh, Luciano Lavagno, Cho Moon, Rajeev Murgai, Alexander Saldanha, Hamid Savoj, Paul R. Stephan, Robert K. Brayton, Albert Sangiovanni-Vincentelli, "SIS: A System for Sequential Circuit Synthesis", Electronics Research Laboratory, 1992.

[7] Xilinx Inc., 2069, Hamilton Ave. San Jose, CA-95125, The Programmable Gate Array Data Book.

[8] 상준영, 이귀상, "BDD를 이용한 다단계 리드 뮬러회로 합성", 정보처리응용학회 논문지, 제 3권, 3호, 1996, pp.640-654.