

# FPGA를 고속으로 동작시키기 위한 지연시간 최적화 알고리즘

김남우\*, 허창우\*, 최익성\*\*, 이범철\*\*  
목원대학교 대학원\*, 한국 전자통신연구원\*\*

(Delay optimization algorithm for the high speed operation of FPGAs)

Nam-Woo Kim\*, Chang-Wu Hur\*, Ick-Sung Choi\*\*, Bum Chul Lee\*\*,  
\*Mokwon university, \*\*Electronics and Telecommunications Research Institute.

E-mail : gotree94@mokwon.ac.kr

## 요 약

본 논문에서는 고속 FPGA 설계를 위한 논리 수준의 조합회로 합성 알고리즘을 제안한다. 제안된 알고리즘은 회로의 지연시간을 줄이기 위해 critical path를 분할한 후 분할된 회로를 동시에 수행하는 구조의 회로를 생성한다. MCNC 표준 테스트 회로에 대한 실험에서 제안된 지연시간 최적화 알고리즘이 기존 알고리즘에 비해 지연시간이 평균 33.3 % 감소된 회로를 생성함을 보였다.

## 1. 서 론

FPGA(Field Programmable Gate Array)는 VLSI ASIC 설계시 공장에서 별도의 공정 과정을 거치지 않고도 손쉽게 연구실이나 현장에서 저가로 회로를 구현할 수 있으므로 개발 기간이 짧고 회로의 수정에 따른 추가 비용이 적은 장점이 있다. 최근 공정과 설계 기술의 발달에 따라 FPGA는 prototype system 개발시 초기 개발 비용이 적고 디자인의 변경이 용이성 등의 많은 장점이 있으므로 그 수요가 점점 늘고 있는 추세이다. FPGA chip은 논리 소자(logic element)와 이를 연결하는 wire로 구성되어 있으며, 논리 소자는 K-input 진리표(lookup table)를 구현할 수 있는 program이 가능한 logic block 들로 구성되어 있다. 일반적으로 logic block의 입력의 수가 적으면 FPGA 논리 회로에서 사용되는 logic block의 효율성이 증가하지만, 회로의 critical path를 구성하는 logic block의 depth가 증가하여 회로의 지연시간이 길어지는 현상이 발생한다. 최근의 FPGA 소자들은 지연시간과 logic block 사용 효율을 개선하기 위하여 하나의 논리 소자가 다수개의 logic block들을 포함하는 복잡한 구조로 설계되어 있다.

FPGA는 회로 구현과 변경이 용이한 장점이 있지만, ASIC에 비해 3 ~ 10 배 지연시간이 길어 100 Mhz 이상의 고속으로 동작하는 시스템의 구현이 어렵고 ASIC에 비해 상대적으로 집적도가 떨어지는 단점이 있다. 이러한 단점을 개선하기 위하여 technology mapping을 적절히 수행하거나 routing과 회로를 적절히 재구성하여 critical path를 구성하는 logic block의 level 수를 줄임으로써

FPGA 회로의 지연시간을 개선하기 위한 방법들이 제안되었다 [1][2].

FPGA는 technology mapping의 목적은 면적과 지연시간을 최소화될 수 있도록 logic element들을 구성하고 연결하는 것이다. FlowMap은 최적의 logic block depth로 구성될 수 있도록 technology mapping을 수행하는 polynomial time 알고리즘이다 [3]. 이 알고리즘은 K개의 입력을 가지는 logic block을 이용하여 구성할 수 있는 최소 depth 한계를 구하여 critical path를 구성한 후 회로의 min-cut과 OBDD를 이용하여 조합회로 최적화 과정을 수행한다. [3]

제안된 논문에서는 FlowMap 등에서 제시한 지연시간 한계보다 지연시간을 개선시킬 수 있는 논리수준의 합성 알고리즘에 대하여 기술한다. 제안된 알고리즘은 주어진 회로를 직렬로 분할하고, 분할된 회로를 동시에 수행하는 구조의 회로를 합성함으로써 지연시간을 개선하였다. 제안된 알고리즘은 회로의 지연시간을 결정하는 critical path를 절단할 수 있도록 회로를 직렬로 분할한 후, 분할된 회로들이 동시에 수행되는 구조의 회로로 변환하여 지연시간을 개선한다. 회로 분할시 cut-set에 포함된 line의 수가 많으면 면적이 방대하게 증가하는 현상이 발생하므로, ROBDD를 이용하여 critical path의 cut-set이 최소화될 수 있도록 회로를 재구성하는 전처리 과정을 거친다. MCNC benchmark 회로를 통한 실험을 통하여 기존의 이론적인 최적 지연시간보다 지연시간을 현저하게 개선할 수 있음을 보였다.[4][5]

## II. FPGA 회로의 지연시간 최적화 알고리즘

### 1. FPGA

FPGA(Field Programmable Gate Array)는 전자회로 설계자가 현장에서 집적회로를 구현할 수 있는 새로운 형태의 반주문형 반도체(ASIC : Application Specific Integrated Circuit)이다. FPGA는 범용 회로의 구현이 어렵고 용량이 적어 응용범위가 제한된 PLD(Programmable Logic Devices)의 단점과 제작 시간과 비용(Non-Recurring Engineering Cost)이 많이 드는 ASIC의 단점을 절충할 수 있는 장점이 있는 소자이지만, 복잡한 배선 구조로 인해 routing 효율성과 동작 속도가 주문형 반도체에 비해 떨어지는 단점이 있다. 현재 FPGA가 주로 사용되고 있는 응용 분야는 칩의 주기(life-time)가 짧은 제품의 ASIC의 대용품, 제품의 변경이 용이한 초기 시제품 개발에 자주 사용되고 있다. 현재 가장 널리 사용되고 있는 상용 FPGA는 Xilinx SRAM-based FPGAs, Actel antifuse-based FPGAs, Altera EEPROM-based FPGA 등을 예로 들 수 있으며, 이러한 FPGA 제품들은 성능과 집적도를 향상시키기 위하여 각 회사마다 고유한 설계 구조를 가지고 있다. 최근의 고성능 고집적 제품인 SRAM-based FPGA는 기본적으로 CLB(Configurable Logic Block)과 IOB(Input Output Block), 그리고 이를 연결해 주는 interconnection wire로 구성된다. SRAM-based FPGA는 내부 구성을 SRAM을 이용하여 설정하며, 대부분 내부에 on-chip 메모리를 가지고 있거나 메모리 소자로 사용 가능한 CLB를 가지고 있는 특징이 있다. 최근 이러한 고성능 고집적 FPGA가 발표되고 있지만, ASIC에 비해 지연시간이 상대적으로 큰 단점을 극복하여 고속으로 동작하는 회로를 구현하기 위해서 설계된 회로가 FPGA 구조에 효율적으로 적용될 수 있도록 설계할 수 있는 효율적인 합성 알고리즘이 필수적으로 요구된다. [6]

### 2. FlowMap 최적 Technology mapping 알고리즘

FPGA 지연시간을 줄이기 위해 technology mapping 과정에서의 최적화 알고리즘에 대해 활발하게 연구가 진행되어 왔으며, 이 중 FlowMap 알고리즘은  $O(N^2)$  시간 내에 최적의 technology mapping을 수행이 가능하다. FlowSyn algorithm은 depth를 최소화하기 위해 network에서 빠른 network flow computation을 먼저 찾은 후 고정된 height 값을 가지는 최소 cut들을 연속적으로 계산함으로써 이루어진다. Boolean optimization은 functional decomposition을 위해 효율적인 OBDD를 이용하여 수행한다. FlowMap의 전체 알고리즘은 두 과정으로 이루어져 있으며, 첫째 과정에서 각 노드에서 최적의 mapping이 되었을 경우 K-LUT의 레벨을 나타내는 label을 계산하고, 두

번째 과정에서는 첫째 과정에서 생성된 각 노드의 label을 이용하여 PO로부터 입력 방향으로 가능한 모든 K-LUT들을 생성하여 technology mapping을 수행한다. FlowMap은 K개 이하의 입력을 가지는 gate로 구성된 K-bounded network를 입력받아 CLB level이 최소화될 수 있도록 technology mapping을 수행하므로, 일반적인 unbounded network를 bounded network으로 변환하는 전처리 과정을 거쳐야 하며, 생성된 K-bounded network에 따라 technology mapping의 결과가 영향을 받는 단점이 있다. [7][8]

FlowMap 최적 알고리즘을 이용하여 16개의 입력을 가지는 조합회로를 4개의 입력을 CLB를 이용하여 technology mapping을 수행할 경우 그림 \*와 같이 최소 2단의 CLB level을 거치게 된다.

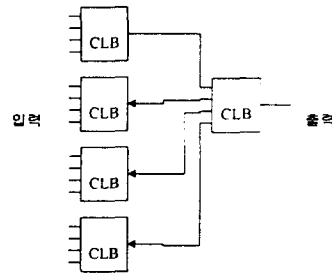


그림 1 예제 조합 회로

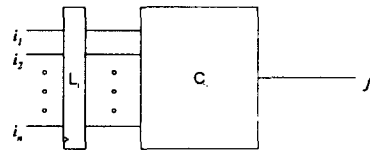


그림 2 게이트 레벨의 기본 회로

## III. 제안된 FPGA 지연시간 최적화 알고리즘

제안된 FPGA 지연시간 최적화 알고리즘은 주어진 회로를 직렬로 분할하여 critical path를 분할하고, 분할된 회로를 동시에 수행함으로써 지연시간이 현저히 개선된 회로를 생성한다. 제안된 방법을 이용하여 기존의 조합회로 최적화와 technology mapping 과정을 통해 얻을 수 있다고 알려진 지연시간 한계보다 지연시간이 개선된 회로를 설계하였다 [9]. 회로 분할시 cut-set에 포함되는 wire의 수가 많을 경우 회로의 크기가 방대하게 증가하는 단점이 있으므로 이를 개선하기 위하여 제안된 알고리즘은 주어진 회로를 cut-set이 적은 회로로 변환하는 전 처리 과정을 거친

후 회로 분할을 수행한다. 제안된 알고리즘은 회로 분할 시 Shannon expansion을 이용하여 두 cofactor 회로로 분할하며, cofactor 회로는 원래 회로보다 지연시간이 크지 않으며, 적절한 divisor를 이용할 경우 주어진 회로보다 지연시간이 감소하는 특징이 있다 [5].

그림3 (a)는 일반적인 조합회로를 나타낸 그림이다. 제안된 알고리즘은 critical path를 분할하기 위하여 주어진 그림3 (a) 회로를 그림3 (b)와 같이 직렬로 회로를 분할한다. 분할된 회로 중 출력 쪽에 가까운 부분을 A, 입력 쪽에 가까운 회로를 B라고 할 때, A와 B 회로를 자른 단면을 지나는 line을 cut-set line이라고 한다. 만일 그림3 (b)에서 cut set의 line 수가 1개일 경우 그림3 (b)의 회로는 그림4 와 같이 변환이 가능하다. 그림4의 Aon과 Aoff은 그림4의 A 회로의 B 함수에 대한 cofactor 회로로 A 회로의 입력 line인 cut-set line에 가능한 모든 값인 1과 0 값을 대입함으로써 각각 변환할 수 있다.

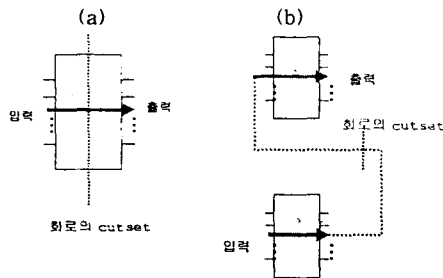


그림 3. Critical path Folding.

만일 cut-set에 포함된 line의 수가 N 개인 경우 N개의 라인값이 대입될 수 있는 경우의 수는  $2^N$ 가지가 되므로 그림5 와 같이  $2^N$ 개의 A 회로가 변환된 회로가 생성된다. 이 경우 면적이 방대하게 증가하여 구현이 어려우므로 제안된 알고리즘은 cut-set에 포함된 line의 수가 1 개가 되도록 회로를 분할한다.

제안된 알고리즘은 critical path상의 line 함수의 커널을 이용하여 회로를 분할한다. 커널은 회로의 공통되는 부분을 나타내는 기본단위로 커널에 사용된 어떠한 변수로 나누어도 나누어 떨어지지 않는 cube를 나타낸다. 임의의 주어진 함수  $f$  는 커널  $B$  에 대하여 다음과 같이 이용하여  $f_B$  와  $f_B^-$  cofactor 회로로 분할이 가능하다. 이때  $A_{on}$  회로는  $f_B$ ,  $A_{off}$  회로는  $f_B^-$ 에 대응된다.

$$f = B \cdot f_B + \bar{B} \cdot f_B^-$$

회로를 분할하는 공통 인수를 어떻게 선택하느냐에 따라 면적, 전력과 속도에 영향을 미치므

로, 커널의 선택에 사용되는 비용 함수를 적절히 설계해야 한다.

$$\text{비용함수} = \max(\text{delay}(f_B), \text{delay}(f_B^-), \text{delay}(K))$$

$$\text{단 } \text{delay}(B) < \max(\text{delay}(f_B), \text{delay}(f_B^-))$$

본 논문에서 제안한 커널 선택 알고리즘은 회로 최적화를 위한 커널을 선택하여 Shannon expansion을 이용하여 회로를 분할하고, 분할된  $f_B$ ,  $f_B^-$ , B 세부 회로 내에서 중복되어 존재하는 공통 회로는 공유하여 사용이 가능하다.

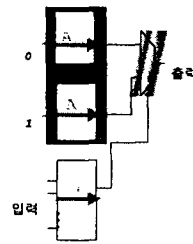


그림 4. Cut-set line이 1개인 경우 그림 \*(b) 회로를 변환한 회로.

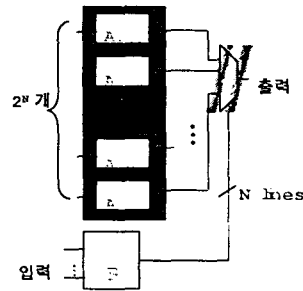


그림 5. Cut-set line의 수가 N개인 경우 변환된 회로.

최근에 사용되고 있는 FPGA는 CLB내에 MUX 나 tristate buffer를 포함하고 있으므로 제안된 구조를 구현시 출력단의 MUX는 추가 CLB나 지연시간 overhead가 없이 FPGA에서 구현이 가능하다. 보통 FPGA에서 mux를 내부 tri-state buffer로 구현할 경우 MUX selection signal에 대응되는 tri-state buffer의 control signal로부터 출력 line까지 지연시간이 증가하는 경우가 발생할 수 있으므로,  $f_B$  와  $f_B^-$  회로의 출력이 계산되지 전에 MUX의 selection signal인 커널 회로 B의 값이 계산되어야 한다. 이를 위해 제안된 알고리즘은 항상 B의 회로의 지연시간이  $f_B$  와

$f_B$  회로의 지연시간보다 적은 조건을 만족하도록 커널을 선택하여 회로를 합성한다.

주어진 회로에 대해 최적의 커널을 선택하고 이 커널을 이용하여 분할한 회로 구조를 그림 6에 나타내었다. 그림 6에서 I는 각각 회로의 전체 입력 집합을 나타내고, f는 함수의 출력 집합을 나타낸다.  $f_K$ 와  $f_{\bar{K}}$ 는 선택된 커널을 이용하여 Shannon expansion을 통해 생성된 cofactor 회로이다. 회로의 MUX selection 신호는 커널을 합성한 회로(K)의 출력 값을 나타내며, 커널을 합성한 회로의 출력값에 따라 두 세부 회로  $f_K$ ,  $f_{\bar{K}}$ 의 출력값 중 하나가 선택된다.

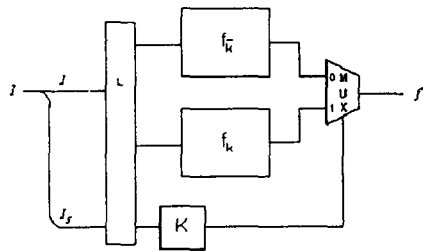
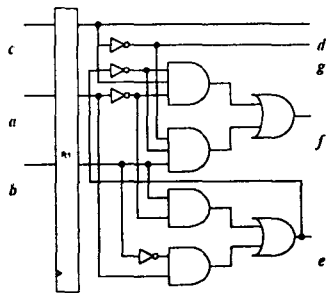
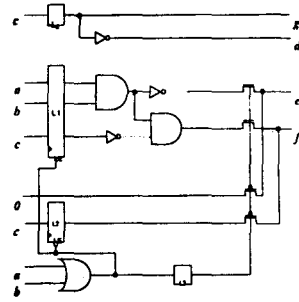


그림 6. 커널 선택 알고리즘을 이용한 회로

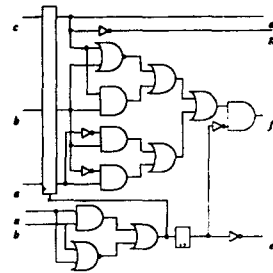
그림 6 (a)는 MCNC 벤치마크 회로 중 b1 회로 예제를 보인다. 이 회로는 FlowMap를 이용하여 최적화한 회로로 지연시간을 나타내는 CLB 레벨이 \*이며, 면적을 나타내는 CLB 개수는 \*이다. 이 예제를 그림 \*에 보인 제안된 구조로 합성할 경우 CLB 레벨이 \*이고, 사용된 CLB 개수가 \*이다. 따라서 예제에서 제안된 합성 알고리즘이 기존의 FlowMap 알고리즘에 비해 지연시간과 면적에서 효율적인 회로를 생성함을 알 수 있다.



(a)



(b)



(c)

그림 6. 예제 회로의 다양한 지연시간 최적화 설계  
 (a) 본래의 조합 논리 회로.  
 (b) FlowMap을 이용하여 생성한 회로.  
 (c) 커널 선택 알고리즘을 이용하여 생성한 회로.

#### IV. 실험 결과

제안한 커널 선택 알고리즘은 SUN UNIX 환경에서 C 언어로 구현되었다. 제안된 커널 선택 알고리즘은 기존의 FlowMap 지연시간 최적화 알고리즘과 결과를 비교하였다. 주어진 회로들은 sis 시스템 환경에서 최적화를 거친후 각 알고리즘에 적용되었으며, 생성된 결과는 각 방식의 Critical Path가 거치는 CLB의 단수와 전체 면적의 CLB개수를 통하여 분석하였다. CLB의 단수는 기존의 방식에 비해 20에서 최고 33.3%까지 줄었음을 알 수 있다. 이로써 CLB 한 단을 거치는데 드는 지연시간이 상당히 줄었음을 알 수 있다. 줄러 들지 않은 원인은 회로자체가 Kernel을 많이 포함하고 있을 경우에는 나누어도 거의 변화가 없다. 또한 나눌 때에 Kernel부분이 다른 회로에 비해 커지면 Technology Mapping 후에 지연시간이 개선되지 않거나 더 나쁜 결과를 얻을 수 있었다. 그러나 우려하였던 바와 같이 면적의 증가는 본회로의 90%정도로 나타났다.

이를 통하여 제안된 회로의 지연시간 최적화

알고리즘은 현재 가장 최적으로 알려진 FlowMap 방법에 비하여 30%의 지연시간의 단축을 가져왔으며, 그로 인해 생기는 면적 overhead면에서는 90%정도의 증가를 가짐을 알았다.

Circuits	#I/O	FlowMap		Proposed		Δ (%)	
		#Level	#CLB	#Level	#CLB	#Level	#CLB
alu4	3 / 4	9	271	6	688	33.3	-153.9
c8	4 / 10	3	21	2	28	33.3	-33.3
cm32a	6 / 8	1	5	1	6	0	-20
k2	21 / 1	6	251	6	537	0	-113
manully	14 / 5	1	1	1	2	0	-100
mixex2	16 / 4	4	20	3	27	25	-35
nxler8	35 / 16	3	20	3	44	0	-120
sao2	14 / 11	6	34	4	50	33.3	-47
sct	26 / 18	3	10	3	23	0	-130
lcon	22 / 29	1	4	1	9	0	-120
v4d	8 / 8	5	142	4	357	20	-151.4
v2	5 / 1	3	8	2	14	33.3	-75
Average	15 / 10	3.6	65.58	3.0	148.75	14.85	-91.55

표 1. 지연 시간 및 면적 비교

V. 결 론

본 논문에서는 커널을 선택하여 회로를 변환시켜 회로의 전력 소모 및 면적을 개선하는 논리 회로 합성 알고리즘을 제안하였다. 제안한 커널 선택 알고리즘은 모든 논리 회로에 적용할 수 있는 알고리즘으로 우선 주어진 논리 회로에 제안한 알고리즘을 적용하여 최적의 커널을 선택하고 이 커널을 중심으로 Shannon expansion을 수행하여 회로를 분할하고 분할된 세부 회로내에서 커널에 의해 영향을 받지 않는 공통 회로를 추출하는 방법으로 회로를 합성한다. 합성된 회로는 커널에 의해 영향을 받지 않는 공통 회로와 커널에 좌우되는 두 세부 회로로 분할되는데, 공통 회로는 늘 동작하나 회로 크기가 전반적으로 작고 나머지 세부 회로는 커널에 의해 한 세부 회로만 구동되고 다른 세부 회로는 disable되어 그만큼 세부 회로의 스위칭 활동은 감소되어 회로에서 발생하는 전력 소모 및 면적은 감소한다. 실험 결과에서 기존의 저전력 소모를 위한 precomputation 논리에 근거한 입력 선택 알고리즘과 비교할 때 커널 선택 알고리즘을 적용해 합성된 회로들이 전력 소모와 회로 면적면에서 보다 효율적임을 보인다. 커널 선택 알고리즘으로 생성된 회로의 출력단의 먹스 회로는 wired-OR 구조의 transmission 게이트로 대체함으로써 추가 회로에 의한 부담을 줄일 수 있다. 더 나아가 커널 선택 알고리즘을 사용하여 설계 작업을 수행할 경우 빠른 시간 내에 적절한 전력 소모 및 회로 면적을 갖는 회로를 합성할 수 있어 최적의 회로를 구현하기 위한 설계 자동화의 효율을 높일 수 있다.

참 고 문 헌

- [1] A. Chandrakasan, R. W. Brodersen, "Minimizing Power Consumption in Digital CMOS Circuits," Proceedings of the IEEE, Vol. 83, No. 4, pp. 498-523, April 1995.
- [2] L. Benini, P. Siegal, G. De Micheli, "Saving Power by Synthesizing Gated Clocks for Sequential Circuits," IEEE Design & Test of Computers, Vol. 11, No. 4, pp. 32-41, Oct. 1994.
- [3] A. Ghosh, S. Devadas, K. Keutzer, J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits," in Proc. 29th DAC, pp. 253-259, June 1992.
- [4] H. Savoj, R. Brayton, H. Touati, "Extracting Local Don't Cares for Network Optimization," in Proc. ICCAD, pp. 514-517, Nov. 1991.
- [5] K. Roy, S. Prasad, "Syclop: Synthesis of CMOS Logic for Low Power Application." in Proc. ICCD, pp. 464-467, Oct. 1992.
- [6] M. Alidina, J. Monteiro, S. Devadas, A. Ghosh, "Precomputation-Based Logic Optimization for Low Power," in Proc. ICCAD, pp. 74-81, Nov. 1994.
- [7] G. De Micheli, *Synthesis and Optimization of Digital Circuits*, McGraw-Hill, 1994.
- [8] 김 형, 최 익성, 서 동욱, 허 훈, 황 선영, "저전력 소모 조합 회로의 설계를 위한 효율적인 알고리즘," 한국통신학회 논문지, 제 21권 제 5호, 1996년 5월, pp. 1221-1229.
- [9] S. Iman, M. Pedram, "Logic Extraction and Factorization for Low Power," in Proc. 32nd DAC, pp. 248-253, June 1995.