論文96-33A-11-16

FPGA 구조 및 로직 블록의 설계에 관한 연구

(A study on the Architecture and Logic Block Design of FPGA)

尹 余 煥 *, 文 重 錫 **, 文 炳 模 ***, 安 成 根 ***, 鄭 德 均 ***

(Y.W. Yoon, J.S. Moon, B.M. Moon, S.K. Ahn, and D.K. Jeong)

요 약

본 연구에서는 SRAM(Static RAM) 셀 프로그래밍 방식을 사용하는 대칭 어레이(symmetrical array) 구조의 FPGA(Field Programmable Gate Array)의 배선 구조 및 논리 블록의 회로를 설계하였다. 설계한 배선 구조는 스위치 매트릭스(switch matrix), 배선 채널 및 입출력 블록으로 구성되어 있으며 배선 채널은 다시 단일 길이 채널, 이중 길이 채널 및 글로벌 길이 채널로 구성되어 있다. 채널 사이의 연결은 SRAM 셀에 의해 제어되는 패스 트랜지스터를 통해 이루어지는데, 신호의 지연 시간(delay time)를 줄이기 위해 게이트 전압을 7V로 올리는 방식을 제안하였다. 설계한 SRAM 셀은 시프트 레지스터(shift register) 기능을 자체에 가지고 있어 별도의 시프트 레지스터가 필요 없고, LUT(Look-Up Table)의 SRAM 셀은 쓰기 동작이 클럭에 동기되어 이루어지도록 설계하였다. 논리 블록(LFU:Logic Function Unit)은 4-입력 LUT 4개와 기타 플립플롭 및 논리 게이트로 이루어져 있으며, LUT들은 메모리로도 사용될 수 있도록 설계하였고, 또 LFU에는 4비트의 덧셈기를 구현할 수 있도록 별도의 캐리 로직(carry logic)을 두고 있다. 본 연구의 FPGA는 0.6㎛ 의 CMOS 공정으로 설계하였으며, 4 비트 카운터를 구현하였을 때 시뮬레이션에서 100MHz에서 동작하는 것을 확인하였다.

Abstract

In this study, we designed the routing structure and logic block of a SRAM cell-based FPGA with symmetrical-array architecture. The designed routing structure is composed of switch matrices, routing channels and I/O blocks, and the routing channels can be subdivided into single length channels, double length channels and global length channels. The interconnection between wires is made through SRAM cell-controlled pass transistors. To reduce the signal delay in pass transistors, we proposed a scheme raising the gate-control voltage to 7V. The designed SRAM cells have built-in shift register capability, so there is no need for separate shift registers. We designed SRAM cells in the LUTs (Look-Up Tables) to enable the write operations to be performed synchronously with the clock for ease of system application. Each logic block (LFU) has four 4-input LUTs, flip-flops and other gates, and the LUTs can be used as SRAM memory. The LFU also has a dedicated carry logic, so a 4-bit adder can be implemented in one LFU. We designed our FPGA using 0.6㎛ CMOS technology, and simulation shows proper operation of a 4 bit counter at 100MHz.

* 正會員, 멀티데이터

** 正會員, Univ. of Southern California S.C.

*** 正會員, 서울大學校 電氣工學部

(School of Elec. Eng., Seoul Nat'l Univ.)

接受日字:1995年5月10日, 수정완료일:1996年11月12日

Ⅰ. 서 론

FPGA는 사용자가 현장에서 직접 회로를 구현하여 사용할 수 있는 집적회로로서 칩 상에 미리 제작된 배선과 논리 블럭들을 프로그래밍함으로써 원하는 기능

을 구현한다. FPGA를 이용하면 사용자가 논리 회로를 설계할 때 칩의 제조 공정을 거치지 않고 설계한 즉시 시험할 수 있기 때문에 칩의 설계 및 원형(prototype) 제작 과정에서 많은 시간과 비용을 절약할 수 있다[1]. FPGA는 프로그래밍 가능한 논리 블럭, 논리 블럭들 사이의 연결 형태를 지정할 수 있는 배선 채널 및 외부와의 입출력을 담당하는 입출력 블럭으로 구성되어 있고, 프로그래밍에서는 논리 블럭들의 기능을 지정해 주고 배선 채널 및 입출력 블럭들의 연결 상태를 지정해주게 된다. 현재까지 상품화된 FPGA 구조는 <그림 1>에서와 같이 각 블럭들의 배치 형태에 따라 대칭 어레이, row-based, sea-of-gates, 계층적 PLD 등으로 나눌 수 있으며, 프로그래밍 방식에도 <표 1>에서와 같이 SRAM 방식, 앤티 퓨즈(anti-fuse) 방식, EPROM 방식, EEPROM 방식 등의 여러 가지가 있다[1].

표 1. 각 프로그래밍 기술의 특성
Table 1. Characteristics of programming techniques.

프로그래밍 기술	정보의 휘발성	재프로그래밍	면 적	접점 저항 (Ω)	용량 (fF)
SRAM 셀	○	○ (회로 내)	크다	1-2k	10-20
anti-fuse	×	×	anti-fuse:작다 프로그래밍 트랜지스터:크다	수십-수백	1-5
EPROM	×	○ (회로 밖)	작다	2-4k	10-20
EEPROM	×	○ (회로 내)	EPROM의 2배	2-4k	10-20

본 연구에서는 재프로그래밍이 가능하고, 표준 CMOS 공정을 사용하여 제작할 수 있는 이점이 있는 SRAM 프로그래밍 방식[1]과 대칭 어레이 구조를 사용하여 논리 블럭들이 2차원상에 대칭적으로 배치되어 있고 그 사이에 스위치 매트릭스 및 수직, 수평의 배선 채널이 있는 구조에 대하여 연구하였다. 배선 회로에서의 패스 트랜지스터의 게이트 전압은 7V로 올려서 접속 저항 및 신호의 손상을 줄이는 방식을 제안하였으며, 논리 블럭은 SRAM 셀 방식의 이점을 살릴 수 있는 LUT(Look-Up Table)를 논리 함수 발생의 기본 소자로 사용하고 SRAM 셀은 프로그래밍 데이타의 로딩(loading)를 위한 시프트 레지스터(shift register) 기능을 내부에 포함하고 있고, LUT의 셀은 메모리 동작시의 쓰기 동작(write operation)의 타이밍 완화를 위해 쓰기가 클럭에 동기되어 이루어지는 동기화된 셀로 설계하여 기존의 방식에 비하여 시스템 구현이 용이하고 메모리의 대역폭은 최대로 이용할 수 있도록 하였다.

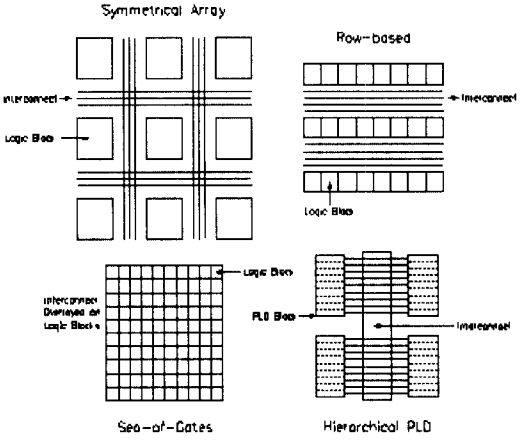


그림 1. FPGA 구조의 일반적인 형태
Fig. 1. General architecture of FPGAs.

표 2. 상용 및 본 연구의 FPGA 특성 요약
Table 2. Summary of commercial FPGAs and our design.

회사	FPGA 구조	논리 블럭 종류	프로그래밍 기술
Xilinx	대칭 어레이	LUT 기반	SRAM
Actel	Row-based	MUX 기반	anti-fuse
Altera	계층적 PLD	PLD 블럭	EPROM
Plessey	Sea-of-gates	NAND 게이트	SRAM
Plus	계층적 PLD	PLD 블럭	EPROM
AMD	계층적 PLD	PLD 블럭	EEPROM
QuickLogic	대칭 어레이	MUX 기반	anti-fuse
Algotronix	Sea-of-gates	MUX 및 기본 게이트	SRAM
Concurrent	Sea-of-gates	MUX 및 기본 게이트	SRAM
Crosspoint	Row-based	트랜지스터 쌍 및 MUX	anti-fuse
본 연구	대칭 어레이	LUT 기반	SRAM

II. FPGA 배선 구조

본 연구에서 설계한 FPGA에서의 배선 회로는 크게 배선 채널과 스위치 매트릭스로 구분되고, 배선 채널은 다시 길이에 따라 단일 길이 채널(SLC:Single Length Channel), 이중 길이 채널(DLC:Double Length Channel) 및 글로벌 길이 채널(GLC:Global Length Channel)로 구분할 수 있다. 배선 채널들은 PCP(programmable contact point)를 통해서 LFU의 포트에 직접 연결할 수 있다.

1. 스위치 매트릭스(Switch matrix)

스위치 매트릭스는 수직, 수평의 SLC 및 DLC가 교차하는 지점에 위치하여 이들을 연결하는 역할을 한다. <그림 2>에 나타난 것 처럼 스위치 매트릭스는 4방향에 각각 12개의 선에 연결되고, 내부에는 12x12의 격자 구조를 이루는 선들의 교차점 중 대각선 상에 위치하는 곳에 연결점이 있다. 하나의 연결점에는 SRAM 셀에 의해 제어되는 6개의 패스 트랜지스터가 있는데, 이들을 이용하면 하나의 연결점에 달려 있는 4개의 선 사이에는 자유롭게 연결할 수 있다. 그러나 연결은 연결점이 있는 곳에서만 가능하기 때문에 스위치 매트릭스에 인입되는 하나의 선은 최대 3개의 다른 선과 연결될 수 있다. 연결 가능한 선의 갯수를 3으로 제한하는 것은 연결점에서의 패스 트랜지스터에 의한 용량이 커지는 것을 막기 위해서이다.

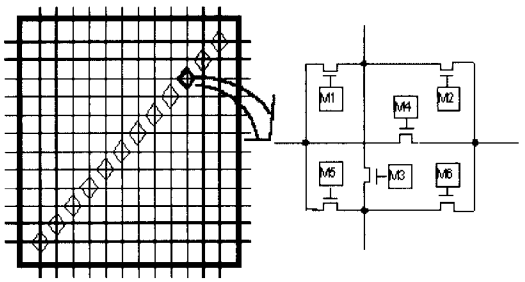


그림 2. 스위치 매트리스의 구조도
Fig. 2. Structure of the swtich matix.

스위치 매트릭스를 비롯하여 SRAM 셀에 의해 제어되는 모든 패스 트랜지스터는 7V 정도의 전압으로 동작하도록 설계하여 패스 트랜지스터에 의한 접속 저항을 줄일 수 있도록 설계하였다. 0.6μm CMOS 공정에 대한 시뮬레이션의 결과에서 게이트 전압이 5V인 경우에는 패스 트랜지스터를 통과한 신호의 전압폭이 약 3.2V인데 반해 7V의 경우에는 5V의 풀 스윙(full swing) 신호를 얻을 수 있었다. 또한 폭 5μm의 패스 트랜지스터의 연결 저항은 5V의 경우 약 2.26kΩ인데 반해 7V의 경우 약 1.88kΩ로서 약 16%정도 줄어든다. 본 연구에서 PCP에 전원을 공급하는 방식을 프로그래밍이 끝난 후 FPGA가 동작할 때에만 전압을 7V으로 올리도록 설계하였다. PCP의 SRAM의 상태는 동작 중에는 변하지 않으므로 거의 모든 동적 전력은 전압을 올리지 않은 프로그래밍 상태에서 소모되고 전압이 7V인 상태에서는 정적인 전력 소모만 발생하여 7V의 전압을 사용하는 것에 의한 추가 전력 소모는 크지 않다. 7V의 전원은 5V의 공급전원으로부터 차지 펌프(charge pump) 회로에 의해 발생시키고 있다. <그림 3>은 PCP의 회로를 나타낸 것으로, 패스 트랜지스터와 SRAM 셀을 결합한 회로이다.

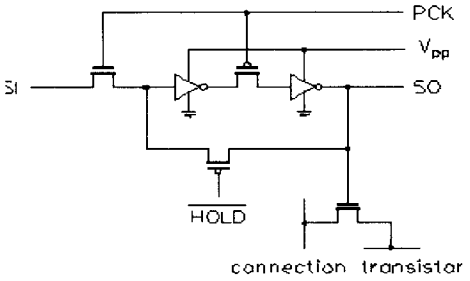


그림 3. PCP의 회로도
Fig. 3. Circuit of the PCP.

2. 단일 길이 채널(SLC:Single Length Channel)

SLC는 인접한 블럭들 사이를 연결하기 위해서 제공되는 배선 자원으로서 인접한 2개의 스위치 매트릭스 사이를 연결하고 있고, LFU의 포트에 연결할 수 있도록 되어 있다. SLC를 통한 배선에는 스위치 매트릭스를 거쳐야 할 경우가 많은데, 스위치 매트릭스에서의 연결 자유도가 3으로 제한되어 있으므로 다양한 배선 경로를 지원하기 위해서는 많은 선이 필요하다. 따라서 SLC의 수를 10개로 설계하였다. LFU의 한쪽 면의 포트수는 6개이므로 10개의 SLC와 2개의 DLC를 가지고 인접한 2개의 LFU의 포트에 서로 다른 배선 경로를 제공하는 것이 가능하다.

3. 이중 길이 채널(DLC:Double Length Channel)

DLC는 SLC의 2배의 길이를 갖는 배선자원으로서

같은 방향으로 연속된 2개의 SLC를 사용해야 할 경우에 사용된다. SLC를 2개 사용하는 경우에는 스위치 매트릭스에서 패스 트랜지스터를 하나 거치는 데에 반해 DLC를 사용하면 패스 트랜지스터를 거치지 않고 배선을 할 수 있다. DLC도 SLC와 마찬가지로 스위치 매트릭스에 연결되어 있는데, 2개의 DLC가 번갈아 가면서 스위치 매트릭스를 거치게 된다. 그러나 DLC는 스위치 매트릭스에서 DLC끼리만 연결할 수 있으므로 SLC와 혼합하여 사용할 수 없다. <그림 5>는 DLC의 구조도이다.

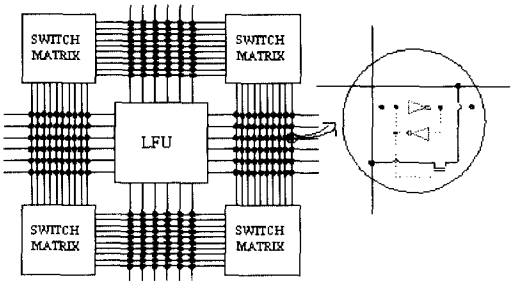


그림 4. 단일 길이 채널의 구조도
Fig. 4. Structure of single length channel.

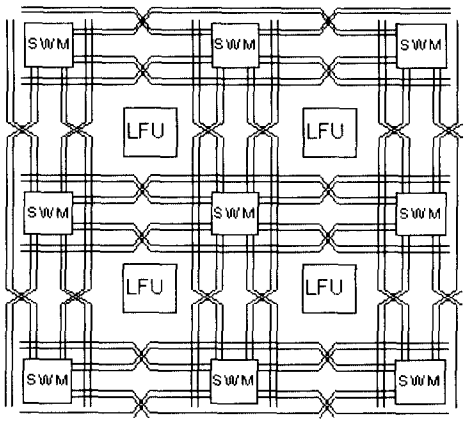


그림 5. 이중 길이 채널의 구조도
Fig. 5. Structure of double length channel.

3. 글로벌 길이 채널(GLC:Global Length Channel)

GLC는 글로벌 신호(global signal)를 위한 배선 자원으로서 스위치 매트릭스를 거치지 않고 상하 또는 좌우의 전 배선 영역에 신호를 전파시킬 수 있도록 설계되어 있다. 이를 이용하면 SLC를 사용하는 것에 비해 패스 트랜지스터를 적게 거치면서 글로벌 신호를 배선할 수 있으므로 스큐(skew) 문제나 지연 문제를 감소시킬 수 있다. GLC는 방향에 따라 수평 GLC와 수직 GLC로 구분할 수 있다. 수평 GLC는 8개의 선으로 구성되어 있고, 수직 GLC에는 8개의 선에 추가로 6개의 선을 더 두고 있다. <그림 6>에는 GLC를 나타냈고, <그림 7>에는 SLC, DLC, GLC 및 LFU의 연결 형태를 나타내었다.

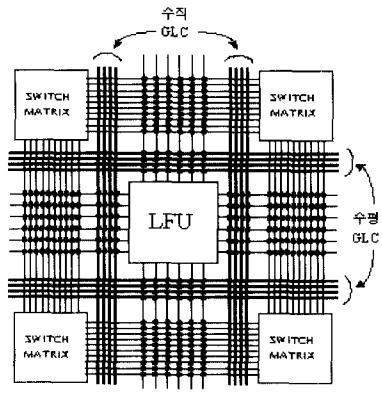


그림 6. 글로벌 길이 채널의 구조도
Fig. 6. Structure of global length channel.

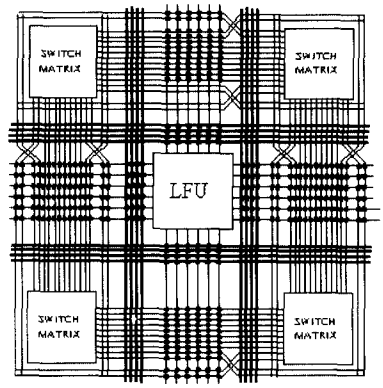
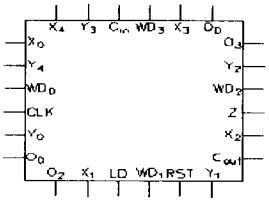


그림 7. LFU의 주변 의 배선 구조
Fig. 7. Interconnection structure near LFU.

Ⅲ. Logic Function Unit(LFU)

1. LFU의 구조 및 기능

본 FPGA의 논리 블록(Logic Function Unit 또는 LFU)은 하나의 블록에 많은 기능을 가지도록 설계하였다. 기본 구조는 4개의 입력을 갖는 LUT(Look-Up Table) 4개와 기타 논리 게이트, MUX 및 플립플롭으로 이루어져 있다. <그림 8>은 LFU의 포트 구성을 나타낸 것으로, 각 포트에 대한 설명은 다음과 같다.



X_0 - X_4 , Y_0 - Y_4 : 논리 입력
WD_0 - WD_3 : 외부 Data 입력
CLK : 클럭 입력
LD : 플립 플립의 Load Enable
RST : 플립플립의 동기 Reset
Cin : 캐리 로직의 캐리 입력
Cout : 캐리 로직의 캐리 출력
Z : 부가적인 로직 입력
O_0 - O_3 : LFU 출력

그림 8. LFU의 포트 구성
Fig. 8. Port view of the LFU.

본 연구에서는 LFU를 크게 하여 하나의 논리 블록(LFU)에 많은 기능을 구현할 수 있도록 설계하여 하나의 LFU에 많은 논리 회로를 구현할 수 있고, 밀접한 연결을 갖는 여러 개의 논리 회로 블록을 하나의 LFU에 통합시킴으로써 LFU간 배선 문제를 완화시킬 수 있도록 하였다. 연구 결과 기술 매핑을 효율적으로 하면 논리 블록(LFU)이 클 경우에는 작은 논리 블록 여러 개를 사용하는 것에 비해 좋은 결과를 나타내고 있다[1)][19].

기본적인 논리함수는 LUT를 통해서 발생시킨다. 연구 결과[11] 4 개의 입력을 갖는 LUT를 사용할 경우가 면적상 가장 효율적이 것으로 나타나고, 또 분해 가능한 LUT의 경우 4 개의 출력을 갖는 경우가 면적상 가장 효율적인 것으로 나타난 것을 참고로 하여 본 연구에서는 4 개의 4-입력 LUT를 사용하였다. LUT의 입력은 X(X_0 ~ X_4)와 Y(Y_0 ~ Y_4)의 2가지 조합이 있는데, LUT 중에서 2개는 입력이 X 입력의 조합으로 고정되어 있고 나머지 2개의 LUT의 입력은 X와 Y 중에서 선택할 수 있게 되어 있다. LFU의 출력들은 외부로 출력되기 전에 플립플롭을 거칠 수 있고, 플립플롭의 입력에는 직접 LFU 외부로부터 데이타를 가할 수 있다. 또한, 플립플롭에는 동기 리세트(synchronous reset) 기능이 있다.

LUT의 출력들은 외부 출력단으로 연결되기 전에 여러 논리 게이트 및 MUX를 거치는데, 이를 이용하면 여러 가지 복잡한 논리 함수를 구현할 수 있다. 또한 LFU에는 별도의 캐리 로직(carry logic)이 있어 4 비트의 덧셈기를 하나의 LFU에 구현할 수 있도록 설계하였다. <그림 9>에는 LFU의 내부 구조를 나타내었다.

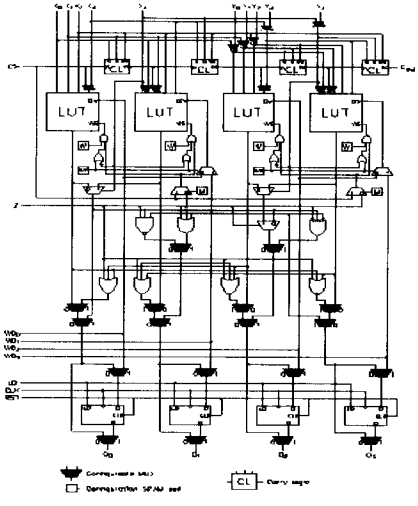


그림 9. LFU의 내부 구조
Fig. 9. Internal structure of the LFU.

1) 논리 함수 발생기

LFU의 LUT들은 4 개의 입력을 갖고 있으므로 이들은 기본적으로 4 개의 완전한 4 변수 함수를 구현할 수 있고, X 및 Y의 2개의 입력 조합에 대한 논리함수를 독립적으로 구현할 수 있다. 또, LUT의 출력들을 MUX에 의해 선택하면 5 변수나 6 변수의 논리함수를 구현할 수 있다. LUT 중 같은 입력 조합을 가질 수 있는 것들끼리 2개씩 묶고 (LUT1과 LUT2, LUT3과 LUT4) LUT의 입력으로는 X_0 ~ X_3, Y_0 ~ Y_3의 4 개만 사용하고 5번째 입력인 X_4나 Y_4는 LUT의 출력들을 선택하는 MUX의 제어신호로 사용할 경우 임의의 5 변수의 구현이 가능하고, X 및 Y의 입력 조합을 사용하면 2개의 독립된 5 변수 함수를 구현할 수 있다. 한편 5 변수 함수 발생기의 구성에서 2 개의 5 변수 함수 발생 부분의 입력 조합을 같게 하고 2개의 출력을 다른 입력 Z를 가지고 MUX에 의해 선택하면 임의

1) 로직 파티션닝(logic partitioning)시에 XINIX사의 5-LUT를 가지는 CLB(Configuration Logic Block)구조를 사용한 경우가 본 논문에서 제안한 6-LUT를 가지는 LFU 구조를 사용한 경우에 비해 LUT수가 약 17% 더 크게 맵핑된다는 실험 결과가 있다.

의 6 변수 논리 함수를 구현할 수 있다. 이밖에 LUT의 출력들과 외부 입력 Z 사이에 NAND, OR 및 XOR 연산이 가능하며, 또 OR 게이트에 의해서 LUT 출력들 사이에 곱의 합(sum of minterms) 형태의 함수 구현이 가능하다. 이와 같은 기능을 이용하면 여러 가지 복잡한 기능을 효과적으로 구현할 수 있다.

2) 메모리 기능

LFU가 메모리로 사용될 때에는 LUT의 입력 X, Y가 주소가 되어 이에 해당하는 위치의 LUT 비트에 대해 읽기/쓰기를 하는 것이 기본 동작이 된다. LUT들의 주소의 조합과 쓰기 인에이블(WE:write enable) 신호의 선택에 따라 여러 가지의 메모리 구성이 가능하며, 크게는 한 메모리 블록에 LUT를 2개만 사용하는 모드와 4개 모두를 사용하는 모드로 나눌 수 있다. 한 메모리 블록에 LUT를 2개만 사용하는 모드(16x2, 32x1)에서는 LUT들을 2개씩 2조로 묶어서 한 조에는 X 입력, 다른 한 조에는 Y 입력을 연결하여 각 조에 다른 주소 입력을 줄 수 있고, WE도 다르게 줄 수 있도록 되어 있다. 따라서 각각의 조를 서로 다른 메모리 블록으로 사용할 수 있다. 또 LUT 별로 메모리 모드의 사용 여부를 따로 지정할 수 있어 2개조 중 하나만을 메모리로 사용하고 다른 하나는 보통의 함수 발생기로 사용할 수 있어 한 LFU에 메모리 및 논리 블록을 함께 구현할 수 있다. 한 조 내에서 LUT를 참조하는 방법에 따라 다른 메모리 구성 방법이 가능하다. 2개의 LUT를 동시에 참조할 경우(WE를 동시에 가함) 2개의 LUT의 데이타가 한 워드를 이루게 되어 2 비트의 워드를 사용하는 16x2 모드로 동작한다. 한편, 한 번에 하나의 LUT만 참조하는 경우 X_4나 Y_4의 값에 따라 참조하는 LUT를 선택하게 되는데, 이 비트은 5번째 주소 비트로 작용하여 LUT 2개가 32 비트의 메모리 블럭 하나로 사용되는 32x1 모드로 동작한다. <그림 10>과 <그림 11>은 각각 16x2, 32x1 모드에서의 LFU 내의 연결도를 나타낸 것이다

LUT 4개를 모두 사용하는 메모리 모드(16x4, 32x2)에서는 2개의 조의 주소 입력을 X 조합으로 동일하게 가하여 2 조를 동시에 참조하는데, 여기서도 한 조 내에서 LUT를 참조하는 방법에 따라 2개의 다른 메모리 구성 방법이 가능하다. 각 조 내에서 2개의 LUT를 16x2의 메모리로 참조할 경우 전체 메모리 구성은 16x4가 되고, LUT 2개를 32x1의 메모리 참조할 경우 전체 메모리 구성은 32x2가 된다. 이때, 16x4 모드에서는 4 비트, 32x2 모드에서는 2 비트 워드 단위로 메모리 참조가 이루어지는데, 16x4나 32x2 모드에서는 LUT 4개를 전부 사용하므로 같은 LFU상에 다른 논리 회로를 구현할 수 없다.

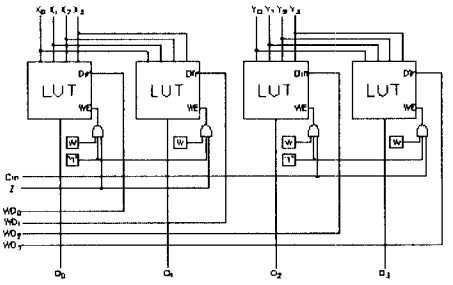


그림 10. 16x2 메모리 모드
Fig. 10. 16x2 memory mode.

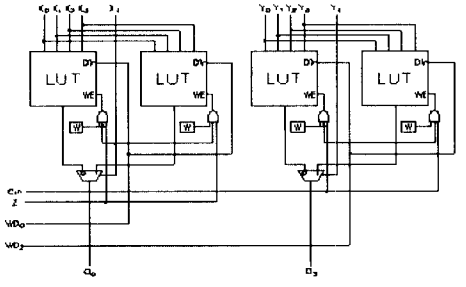


그림 11. 32x1 메모리 모드
Fig. 11. 32x1 memory mode.

2. 캐리 로직(Carry logic)

LFU의 캐리 로직 회로는 캐리가 C_{in} 입력으로부터 C_{out} 출력까지 캐리 로직 블럭을 따라 계속 전파되는 구조로 구성되어 LFU 내에서의 캐리 입력(carry in)으로부터 캐리 출력(carry out)까지의 지연 시간이 최

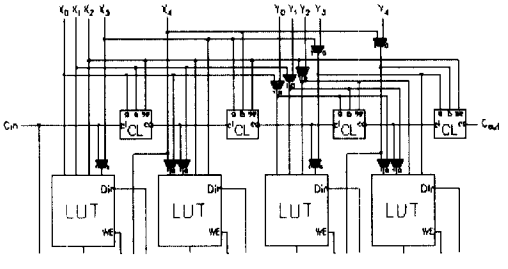


그림 12. 캐리 로직의 구조
Fig. 12. Structure of the carry logic.

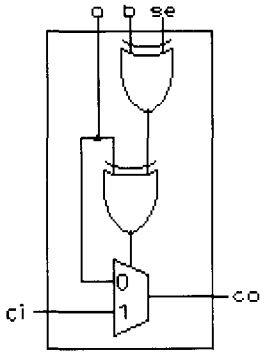


그림 13. 캐리 발생기
Fig. 13. Carry generator.

소가 되도록 설계하였다. 각 캐리 로직 블럭에서는 더하려는 2 비트의 값에 따라 캐리의 전달, 발생을 판단하여 이의 결과로 MUX를 통해 앞 단에서 올라오는 캐리와 발생되는 캐리중에서 선택하여 다음 단으로 캐리를 보내는 동작을 한다. <그림 12>과 <그림 13>은 각각 캐리 회로의 구조와 캐리 발생기의 회로를 나타낸 것으로 카운터나 덧셈기를 구현할 때 빠른 동작 속도를 낼 수 있다.

3. LUT의 회로

<그림 14>에 나타난 것 처럼 LUT는 4x4 구조의 SRAM 셀 어레이와 행/열 디코더, 클럭 제어 회로 및 입력/출력 버퍼로 구성되어 있다.

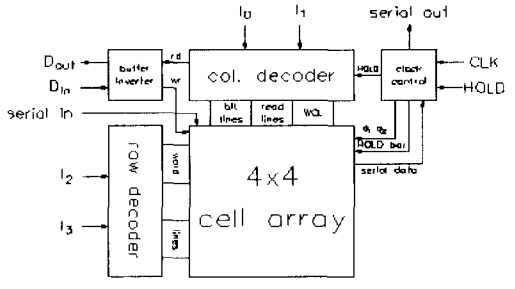


그림 14. LUT의 내부 구성
Fig. 14. Internal structure of the LUT.

1) 시프트 레지스터 기능을 갖는 SRAM 셀

SRAM 셀은 프로그래밍시의 직렬 데이타의 로드를 위한 별도의 시프트 레지스터가 필요하지 않도록 셀 내에 시프트 레지스터 기능을 포함시켰다. 시프트 레지스터 기능을 갖는 SRAM 셀의 기본적인 구조는 2개의 인버터로 구성된 루프와 이를 끊을 수 있는 여러 개(최소 3개)의 패스 트랜지스터로 이루어져 있다. <그림 15>은 본 설계에서 사용하는 기본 SRAM 셀의 회로도를 나타낸 것이다.

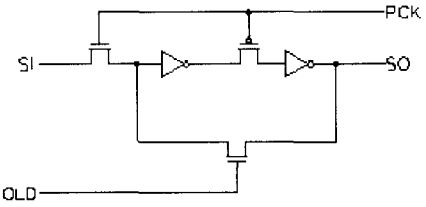


그림 15. 시프트 레지스터 기능을 갖는 SRAM 셀
Fig. 15. SRAM cell with built-in shift register.

패스 트랜지스터들은 HOLD와 클럭에 의해 제어되는데, 프로그래밍시에는 HOLD가 0이 되어 PMOS에 의해 인버터 루프가 끊어지고 나머지 패스 트랜지스터들은 클럭에 의해 인버터와 함께 동적 시프트 레지스터처럼 동작하여 프로그래밍 데이타는 직렬 방식으로 SRAM 셀들을 차례로 통과하여 프로그래밍이 이루어진다. 프로그래밍이 끝나면 HOLD가 1이 되고 클럭이 멈추어서 패스 트랜지스터들이 인버터 루프를 닫고 시프트 레지스터의 직렬 연결을 끊게 되어 적재된 값을 유지하게 된다. 이러한 SRAM 셀은 LFU의 LUT를 제외한 모든 SRAM 셀(PCP 포함)의 기본 구조를 이루고 있다.

2) LUT의 SRAM 셀

LUT에 사용되는 SRAM 셀들은 메모리 동작시 읽기/쓰기를 가능하게 하기 위해 기본 SRAM 셀을 변형시킨 것으로서, 읽기 포트, 쓰기 포트, 셀 선택 및 쓰기 인이에이블(WE:write enable) 제어를 위한 부가적인 트랜지스터들이 있다. LUT SRAM 셀은 기본 SRAM 셀과는 달리 프로그래밍 중이 아니라도 계속 클럭에 의한 동적인 동작이 가능한데, 이때 셀에 저장된 데이타는 인버터 루프에서 계속 재생된다. <그림 16>에는 LUT의 SRAM 셀의 회로도를 나타내었다.

읽기 포트는 두번째 인버터의 출력에 연결된 NMOS로 구성되는데, 워드 라인(WL)이 1이 되면 NMOS가 on 되어 저장된 값이 읽기 라인(RD)로 출력된다. 쓰기 포트는 워드 라인와 WCL(write column line)에 의해 첫번째 인버터의 입력이 선택되는 구조로 되어 있는데, 워드 라인과 WCL이 동시에 1이면 쓰기 라인

(WD)을 선택하고 기타 경우에는 두번째 인버터의 출력을 선택하게 되어 워드 라인과 WCL이 동시에 1일 때에만 쓰기가 이루어진다. 메모리 모드에서는 프로그래밍 중이 아니라도 클럭(Φ1, Φ2)에 의한 동적인 동작이 이루어지므로 메모리 쓰기 동작도 직렬 데이타의 로드 동작 처럼 클럭에 동기되어 이루어진다.

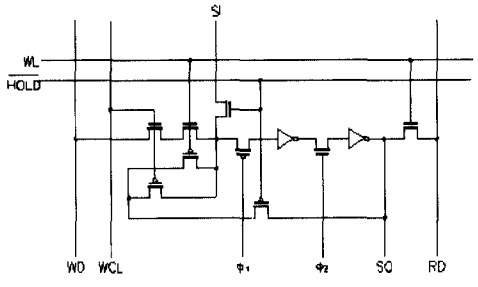


그림 16. LUT의 SRAM 셀 (동기화된 셀)
Fig. 16. SRAM cell (synchronous cell) of LUTs.

한편, LUT가 메모리로 사용되지 않을 경우 프로그래밍 중 이외에는 클럭에 의한 동적인 동작이 필요가 없는데, 이럴 때 Φ_1, Φ_2의 상태를 고정시켜 인버터 루프가 항상 닫히도록 한다. 이것에 대한 자세한 설명은 클럭 제어 회로를 다룰 때 하기로 한다.

3) LUT의 SRAM 셀 구성

<그림 17>에 나타난 것 처럼 LUT의 셀 어레이에서는 16개의 SRAM 셀들이 4x4 형태로 배열되어 있고, 각 행 마다 워드 라인(WL_0~WL_3)이 있고 각 열에는 읽기 라인, 쓰기 라인및 WCL이 있어서 해당 행이나 열의 모든 셀들과 연결되어 있다. 워드 라인들은 행 선택의 기능을 갖는데, 이들 중 LUT 입력(주소)에 해당하는 행의 워드 라인만이 하이(high)가 되어 해당 행의 셀에 대해서만 읽기/쓰기를 가능하게 한다.

각 읽기 라인는 비트 라인(BL_0~BL_3)에 의해 제어되는 패스 트랜지스터를 거쳐서 하나의 선에 연결되어 버퍼를 통해 외부 출력으로 나간다. 비트 라인들은 열 선택의 기능을 갖는데, LUT의 주소 입력에 해당하는 열의 비트 라인만이 1이 되어, 읽기 동작시에는 해당 열의 읽기 라인만이 Dout 출력으로 연결되고, 쓰기 동작시에는 해당 열의 WCL만이 쓰기가 가능하므로 이것과 워드 라인의 조합에 해당하는 하나의 셀만이 쓰여질 수 있다.

WCL 신호는 비트 라인, WE(write enable) 및 HOLD를 AND시켜서 발생하고 있는데, 각 WCL 신호는 HOLD가 1(프로그래밍 중이 아닐 때)이고 해당 비트 라인이 1(해당 열이 선택)인 상태에서 WE이 1(쓰기 동작)일 때에만 1이 되어 셀의 쓰기를 가능하게 한다. 이때, 셀로 쓰여질 데이타의 로드는 직렬 데이타 로드와 같은 방법으로 이루어지므로 실제 쓰기는 클럭의 상승 엣지에 동기되어 일어난다. 쓰기 동작시의 열 선택은 WCL에 의해 수행되므로 열 패스 트랜지스터 등의 필요 없이 데이타의 입력은 버퍼를 거쳐 각 쓰기 라인과 직접 연결되어 있다. WCL를 발생시키는 AND 게이트에서의 HOLD는 프로그래밍시 직렬 데이타의 로드 이외의 쓰기를 불가능하게 하고 있다.

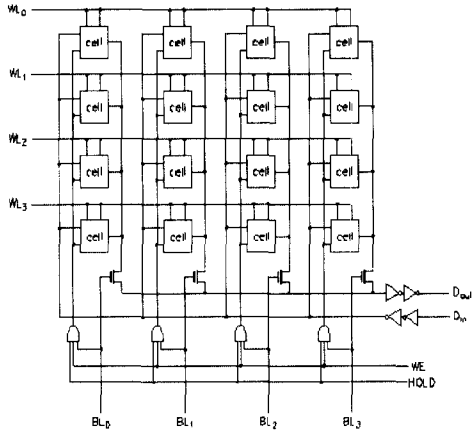


그림 17. LUT 셀의 구성
Fig. 17. Organization of the LUT cells.

4) 클럭 제어 회로

LFU에 공급되는 클럭는 3가지로서, 플립플롭에 공급되는 클럭(CLK), LUT 셀에 공급되는 클럭(Φ_1, Φ_2), 그리고 LUT 이외의 SRAM 셀에 공급되는 프로그래밍 클럭(PCK)이다. <그림 18>에 나타난 것처럼 플립플롭의 클럭는 시스템 클럭(CLK)를 그대로 사용하고, 프로그래밍시에는 CLK를 직렬 데이타 로드에 사용하기 위해 CLK로부터 Φ_1, Φ_2 및 PCK를 만들고 있다. 이와 같은 클럭을 제어하는 회로는 <그림 19>에 나타나 있다.

Φ_1, Φ_2는 LUT가 메모리로 사용될 경우와 그렇지 않을 경우에 그 값이 달라지는데, 메모리로 사용될 경우에는 프로그래밍 중의 여부에 관계 없이 항상 CLK와 같게 된다. 메모리로 사용되지 않을 경우에는 프로그래

밍 중일 때에만 CLK과 같고 프로그래밍이 끝나면 Φ_1은 0, Φ_2는 1로 고정되어 인버터 루프는 항상 닫히게 된다. 두 가지 모드 사이의 선택은 프로그래밍 데이타(SRAM 셀)에 의해 이루어진다. 각각의 모드에서 HOLD 신호의 상태에 따른 Φ_1, Φ_2 및 PCK의 값은 <표 3>과 같다.

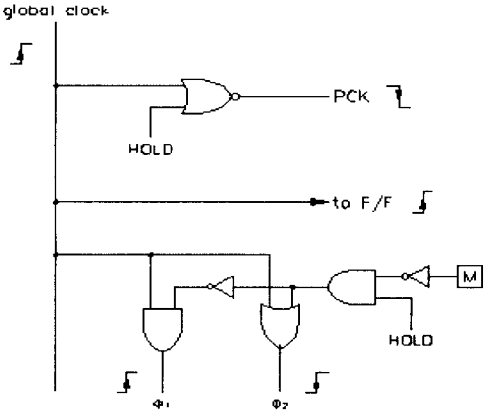
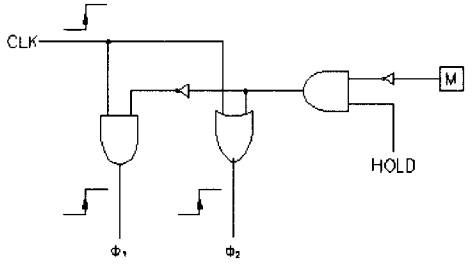


그림 18. 클럭의 공급
Fig. 18. Clock distribution.



M = 1 : LUT를 메모리로 사용
M = 0 : LUT를 메모리로 사용하지 않음

그림 19. LUT의 클럭 제어 회로
Fig. 19. Clock control circuit of the LUT.

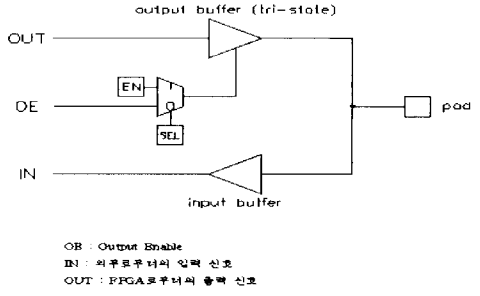
표 4. 각 모드에서의 Φ1, Φ2 및 PCK의 값
Table 4. Values of Φ1, Φ2 and PCK.

LUT의 모드	동작 상태	Φ_1	Φ_2	PCK
메모리	프로그래밍 중	CLK	CLK	CLK_bar
메모리	정상 동작	CLK	CLK	0
논리 회로	프로그래밍 중	CLK	CLK	CLK_bar
논리 회로	정상 동작	0	1	0

PCK는 프로그래밍 중일 때에만 필요하므로 HOLD가 0(프로그래밍 중)일 때에는 CLK의 반전과 같고, HOLD가 1(정상 동작)일 때에는 로직'0'로 고정된다. 이때, CLK를 그대로 사용하지 않고 반전시켜서 사용하는 것은, LUT의 셀들은 클럭의 상승 엣지(rising edge)에서 직렬 데이타가 출력되지만, 일반 SRAM 셀(기본 SRAM 셀)들은 하강 엣지(falling edge)에서 출력되기 때문이다.

Ⅳ. 입출력 블록 (IOB 또는 Input/Output Block)

입출력 블록은 FPGA를 외부와 연결시키는 블럭으로서 입출력 핀을 이루고 있다. 본 연구에서는 여러 개의 FPGA를 연결하여 사용하는 시스템을 고려하여 핀 수가 많은 FPGA를 만들기 위하여 기본적인 기능, 즉 입력, 출력, 하이 임피던스(high impedance 또는 tri-state) 제어를 지원하는 회로를 설계하여 회로를 단순화시키고 칩 크기를 줄일 수 있도록 하였다. <그림 20>에는 입출력의 구조도를 나타내었다.



OE : Output Enable
IN : 외부로부터의 입력 신호
OUT : FPGA로부터의 출력 신호

그림 20. 입출력 블록의 회로도
Fig. 20. Circuit of the IOB.

Ⅴ. FPGA의 프로그래밍

모든 프로그래밍 가능한 SRAM 셀들은 하나의 직렬 경로 상에 위치해 있으므로, 모든 프로그래밍 데이터는 하나의 비트열로 이루어지게 되며, 외부 프로그래밍 데이터의 입력은 하나의 포트를 통해 이루어지고, <그림 21>과 같고, SRAM 셀들은 LFU 및 채널의 PCP를 나타낸다. <그림 22>에서는 프로그래밍 데이터는 시

프트 레지스터 기능을 갖는 SRAM 셀 들을 통해서 직렬로 입력되다가 모든 프로그래밍 데이터가 입력되는 순간 Porg 신호는 로우(low)로 떨어지고 Hold 신호가 하이(high)가 되는 모든 SRAM 셀 들이 프로그래밍 된다. 다음은 프로그래밍시 사용되는 모든 신호들의 타이밍 다이어그램을 나타낸 것이다. 직렬 경로 상에서의 프로그래밍 데이터의 이동은 프로그래밍 클럭(PCK)의 상승엣지에 따라 이동하게 되고 프로그래밍 데이터가 모두 입력되는 순간, Hold 신호는 로우(low)에서 하이(high)가 되면서 SRAM 셀이 프로그래밍되게 된다.

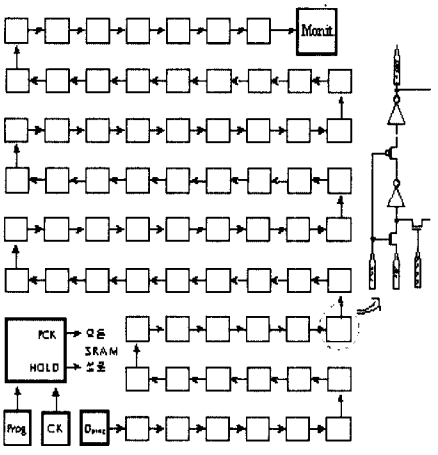


그림 21. FPGA의 프로그래밍
Fig. 21. Programming of FPGA.

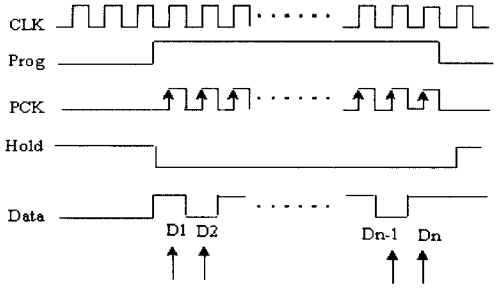


그림 22. 프로그래밍의 타이밍 다이어그램
Fig. 22. Timing diagram of programming.

Ⅵ. 회로 설계 및 성능 평가

본 연구에서 제안하는 FPGA는 0.6μm CMOS 공정으로 설계하였으며, LFU의 크기는 약 0.6x0.6mm^{2}로서 8x8의 어레이 구조의 FPGA를 제작할 경우 칩 크기는 약 10x10mm^{2}가 되고, 구현 가능한 게이트 수는 약 3200개가 된다. <그림 23>에는 LFU의 레이아웃을 나타내었다.

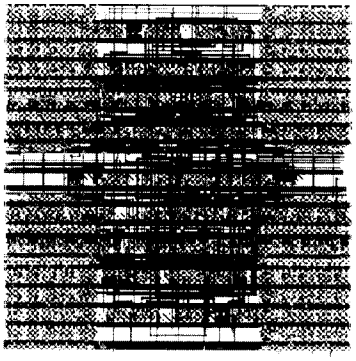


그림 23. LFU의 레이아웃
Fig. 23. Layoug of the LFU.

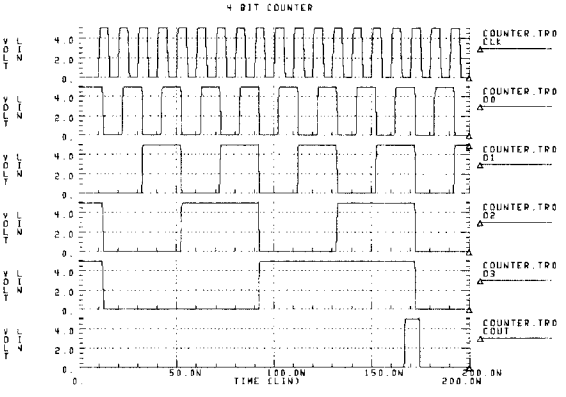


그림 24. 4 비트 카운터의 HSPICE 시뮬레이션 결과
Fig. 24. HSPICE simulation of 4-bit counter.

설계한 LFU의 캐리 로직을 이용하여 구현한 4 비트 카운터에 대한 HSPICE 시뮬레이션을 한 결과 카운터의 최대 동작 속도는 약 100MHz로 나타났다. <그림 24>에는 시뮬레이션 결과를 나타내었다.

Ⅶ. 결 론

본 연구에서는 SRAM 프로그래밍 기술을 사용하는 대칭 어레이 형태를 가지는 FPGA 구조를 제안하고 이를 회로로 설계하였다. SRAM 셀은 프로그래밍시 별도의 쉬프트 레지스터의 필요가 없도록 내부에 쉬프트 레지스터 기능이 있는 구조로 되어 있으며, LUT의 SRAM 셀 들은 메모리 동작시 쓰기가 클럭에 동기 되어 일어나도록 동기 화된 구조로 되어 있어 시스템에

서의 응용이 매우 쉽도록 설계되었다.

논리 블록인 LFU는 2개의 임의의 형태의 4입력 또는 5입력 논리회로나 1개의 임의의 형태의 6입력 논리회로를 구현할 수 있을 뿐만 아니라 16x2, 32x1, 16x4, 32x2의 형태의 SRAM으로 구성할 수 있고, 카운터나 ALU를 용이하게 구현하기 위해 별도의 캐리 로직을 두는 비교적 큰 구조로 설계하여 여러 개의 작은 논리 블록을 사용하는 경우에 생길 수 있는 블럭간 배선 문제를 줄이려고 시도하였다.

배선 회로에서는 비교적 간단한 스위치 매트릭스를 채택하여 하나의 선이 나머지 3개의 선과 임의로 연결할 수 있는 구조로 설계하였으며, 배선 채널은 단일 길이 채널(SLC), 이중 길이 채널(DLC), 글로벌 길이 채널(GLC) 등으로 구별하여 배선에 의한 지연 시간을 줄일 수 있도록 하였다. 또한, 선들의 연결에 사용되는 PCP에서는 패스 트랜지스터의 게이트 전압을 약 7V로 올려 패스 트랜지스터의 접속 저항을 줄일 수 있는 방식을 제안하였다. 이는 CMOS 로직 특성이 갖은 다이나믹 전력 소모가 조금 커질수 있지만, 연결 선(interconnection wire)의 RC 지연 시간을 줄임으로써 FPGA 전체 동작 속도를 향상 시켰다.

본 연구에서 제안하는 FPGA는 0.6㎛ CMOS 공정으로 설계하였으며, 8x8의 FPGA의 경우 칩 크기는 약 10x10mm^{2}가 되며, 구현 가능한 게이트 수는 약 3200[2]개가된다. 시뮬레이션 결과 4 비트 카운터로 프로그래밍 하였을 때 약 100MHz[3]의 동작 속도를 얻을 수 있었으며 이는 기존의 작은 구조의 LFU보다 큰 구조를 가진 LFU의 성능이 좋음을 나타낸다.

2) 구현 가능 게이트 수 산출 : (LFU 당 50 게이트) x (8x8 array) = 3200 LFU 당 50 게이트는 Xilinx 4000의 high count chip의 (전체 게이트 수) / (CLB 수) 에서 CLB당 게이트 수를 얻은 다음, LUT 개수에 따라 스케일링하고 (Xilinx : 우리 = 2 :4) 게이트를 추가하여 나온 결과임.

3) Xilinx 4000의 시스템이 클럭이 최대 50Mhz까지 지원함.

참 고 문 헌

[1] Stephen D. Brown, Robert J. Franscis Rose, Zvonko G. Vranesic, *Field Programmable Gate Arrays*, Kluwer Academic Publishers, 1992.

[2] *The Programmable Logic Data Book*, Xilinx, 1993.

[3] Hung-Cheng Hsieh, *5-Transistor Memory Cell with Known State on Power-Up*, U.S. Patent 4,821,233,1989.

[4] Jonathan Rose, Abbas El Gamal, and Alberto Sangiovanni Vincenteli, *"Architecture of Field-Programmable Gate Arrays"*, Proceedings of the IEEE, Vol.81, N0. 7, pp. 1013~1029 July 1993.

[5] H. Shin and C. Kim, "Performance-Oriented Technology Mapping for LUT-Based FPGAs," IEEE Transactions on VLSI systems, Vol.3, No. 2, pp.323-327, June, 1995.

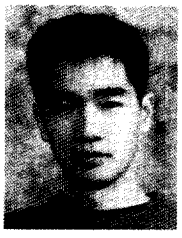
[6] Jonathan Rose, Robert J. Francis, Paul Chow, David Lewis, *"The Effect of Logic Block Complexity on Area of Programmable Gate Arrays"*, IEEE CICC, pp.5.3.1~5.3.5, 1989.

[7] Barry K. Britton, Dwight D. Hill, WilliamOswald, Nam-Sung Woo, Satwant Singh, *"Optimized Reconfigurable Cell Array Architecture for High Performance Field Programmable Gate Arrays"*, IEEE CICC, pp.7.2.1~7.2.5 1993.

[8] Hung-Cheng Hsieh, William S. Carter, Jason Ja, Ed Cheung, Steve Schreifels, Chuck Erickeson, Philip Freidin, Liane Tinkey, and Roy Kanazawa, *"Third-Generation Architecture Boosts Speed and Density of Field-Programmable Gate Array"*, IEEE CICC, pp. 31.2.1~31.2.7 1990.

[9] 신현철, *FPGA를 위한 Partitioning 및 Technology Mapping Tool 개발*, ISRC 연구결과요약집, 서울대학교 반도체공동연구소, 1996

저 자 소 개



尹 余 煥(正會員)

1971년 10월 3일생. 1994년 2월 서울대학교 전자공학과 졸업(B.S). 1996년 2월 서울대학교 전자공학과 졸업(M.S). 1996년 2월 ~ 현재 멀티 데이터

文 炳 模(正會員)

1972년 7월 25일생. 1995년 2월 서울산업대학교 전자공학과 졸업(B.S). 1995년 2월 ~ 현재 서울대학교 전자공학과 석사과정

鄭 德 均(正會員)

1958년 8월 29일생. 1981년 2월 서울대학교 전자공학과 졸업(B.S). 1984년 8월 서울대학교 전자공학과 졸업(M.S). 1985년 5월 U.C Berkeley 전기/컴퓨터 공학(Ph.D). 1989년 5월 ~ 1991년 8월 Texas Instruments 연구원. 1991년 8월 ~ 현재 서울대학교 전기공학부 교수

文 重 錫(正會員)

1969년 7월 30일생. 1992년 2월 서울대학교 전자공학과 졸업(B.S). 1995년 2월 서울대학교 전자공학과 졸업(M.S). 1995년 2월 ~ 1996년 2월 삼성전자 연구원. 1996년 3월 ~ 현재 Univ. S.C. 석사과정.

安 成 根(正會員)

1968년 4월 10일생. 1996년 2월 서울대학교 전기공학부 졸업(B.S). 1996년 2월 ~ 현재 서울대학교 전자공학부 석사과정