

論文96-33A-5-21

# 안티퓨즈 FPGA의 배선지연시간을 고려한 VHDL 모델링

## (VHDL Modeling Considering Routing Delay in Antifuse-based FPGAs)

白瑛錫\*, 曹漢鎭\*, 朴仁學\*, 金景洙\*

(Young Seok Baek, Han Jin Cho, Inhag Park, and Kyung Soo Kim)

## 요 약

본 논문에서는 안티퓨즈 방식의 FPGA를 이용한 디지털 시스템 설계시 FPGA 칩 및 전용 CAD 시스템의 검증을 위한 VHDL 모델링 기법에 대하여 논한다. VHDL 모델은 원판의 프로그램 회로, 논리모듈, 배선선분, 안티퓨즈 및 입출력 패드 등을 표현한다. 배선경로의 지연시간은 프로그램된 안티퓨즈들의 저항과 프로그램되지 않은 안티퓨즈들과 배선선분의 정전용량에 의해 발생하는데, 복잡한 배선경로의 경우 배선지연시간은 논리 모듈의 지연시간의 최대 수배에 달하여 무시할 수 없다. 본 논문은 배선경로의 지연시간을 추출하기 위해, C 언어를 이용하여 프로그램된 안티퓨즈의 위치정보로부터 배선경로를 재구성한다. 재구성된 배선경로로부터 지연시간을 계산하여 VHDL 시뮬레이션에 피드백(feedback) 한다. 지연시간 모델에 따른 영향평가를 위해 SPICE, Elmore 및 Horowitz 등의 방식을 구현하여 실험하였다. 본 검증 기법은 ETRI FPGA 시스템에 구현되어 원판과 회로의 논리 및 타이밍검증을 수행하며, 또한 임계경로 탐색을 통해 구현된 회로의 성능을 예측하는 수단을 제공한다.

## Abstract

This paper describes a post-layout simulation method using VHDL and C for verifying the architecture of antifuse-based FPGAs and the dedicated CAD system. An antifuse-based FPGA consists of programming circuitry including decoding logic, logic modules, segmented tracks, antifuses and I/O pads. The VHDL model which includes all these elements is used for logic verification and programming verification of the implemented circuit by reconstructing the logic circuit from the bit-stream generated from layout tool. The implemented circuit comprises of logic modules and routing networks. Since the routing delay of the complex networks is comparable to the delay of the logic module in the FPGA, the accurate post-layout simulation is essential to the FPGA system. In this paper, the C program calculates the delay of the routing networks using SPICE, Elmore or Horowitz delay models and the results feedback to the VHDL simulation. Critical path can be found from this post-layout simulation results.

## I. 서 론

반도체 설계 기술의 발전으로 설계제품의 수명(life

cycle)이 점점 짧아짐에 따라 설계자의 설계요구를 빠르게 제품화할 수 있는 FPGA<sup>[1]</sup>의 사용이 증대되었으며, 이로 인하여 다양한 형태의 FPGA들이 연구 개발되고 있다.

FPGA의 개발에는 FPGA 원판 설계와 FPGA의 사용 시에 필요한 전용 CAD 환경의 개발이 병행되어 진행되어야 한다. FPGA 원판 설계는 프로그램 소자의 설계, Logic Module설계, 배선자원의 설계, 프로그램

\* 正會員, 韓國電子通信研究所

(Electronics and Telecommunications Research Institute)

接受日字: 1995年10月10日, 수정완료일: 1996年4月24日

회로의 설계 등이며, 전용 CAD 환경은 설계 입력 환경과 입력된 회로를 FPGA 논리 모듈에 효율적으로 분할하는 기술 맵핑(technology mapper)와 생성된 연결정보로부터 회로를 원판에 효과적으로 배치하고 배선하는 알고리즘이외에도 전압을 인가하여 스위치를 프로그램 하여야 하는 FPGA(antifuse-based 혹은 EPROM-based FPGA)의 특성상 배치 배선기에서 출력되어진 프로그램 될 스위치들의 위치를 원판의 디코더 구조에 맞게 입력 패턴을 생성하는 비트열(bit-stream) 생성기가 필요하다.

생성된 비트열은 하드웨어 프로그래머에 의해 최종적으로 FPGA 원판에 입력되어 원하는 안티퓨즈 스위치들이 프로그램 된다. 본 논문에서는 ETRI FPGA의 설계를 위하여 FPGA 칩의 제조전에 FPGA원판<sup>12</sup>을 검증하는 방법을 기술한다. FPGA 원판은 VHDL과 C 언어로 모델링하였으며, 이 모델링된 FPGA원판은 배치배선 알고리즘에 의하여 생성된 비트열을 입력으로 하여 프로그램 된 안티퓨즈의 위치 정보들과 그때의 구현된 회로의 논리 및 지연시간을 이용하여 FPGA 원판과 CAD 환경을 검증한다. 그림 1은 FPGA 검증 시스템의 구성도를 나타내며, VHDL 시뮬레이션의 결과 발생한 오류는 해당하는 부분으로 보내져 수정되며 원판, 배치 배선 알고리즘 및 하드웨어 프로그래머의 동작이 맞을때까지 반복되어 진다.

FPGA의 논리가 맞지않을 경우에는 생성된 비트열, 배치 배선알고리즘에대한 오류발생 가능성에대한 조사가 이루어지며 이의 오류가 없으면 원판의 설계에대한 오류발생 가능성의 조사가 이루어진다. 소자의 지연시간에의하여 논리 오류가 발생시에는 배치배선 알고리즘설계에서 배선시 통과하는 안티퓨즈의 수에대한 제한을 고려하여 배치배선 알고리즘을 재설계한다. 배치 배선 알고리즘의 재설계로 해결이 되지 않는 경우는 FPGA의 각 소자에대한 재설계로 해결한다.

상기의 배치배선 알고리즘의 검증시 배선되는 경로에 존재하는 안티퓨즈의 갯수와 배선 길이에 의하여 지연시간이 달라진다. 따라서 정확한 배치 배선 알고리즘을 검증하기 위하여는 이러한 배선자원의 지연시간에 대한 모델링이 필요하다. 구현된 회로의 지연시간의 모델은 FPGA 원판의 논리 구성 요소에 대한 모델과 안티퓨즈와 배선선분으로 이루어지는 배선경로의 모델로 나누어진다. FPGA의 원판 상에서 안티퓨즈의 단락 저항은 보통 50 ~ 500 ohm 정도이며, 기본 배선선분

의 정전용량 값이 200pF 정도이므로 복잡한 배선경로의 경우 배선부분의 지연시간이 FPGA상의 논리모듈의 지연시간과 비슷하게 된다. 따라서 이러한 배선경로의 지연시간은 배선경로의 지연시간은 프로그램된 안티퓨즈의 위치 및 구성에 따라서 변하게되므로 FPGA 원판의 모델링 시에 반드시 고려되어야 한다.

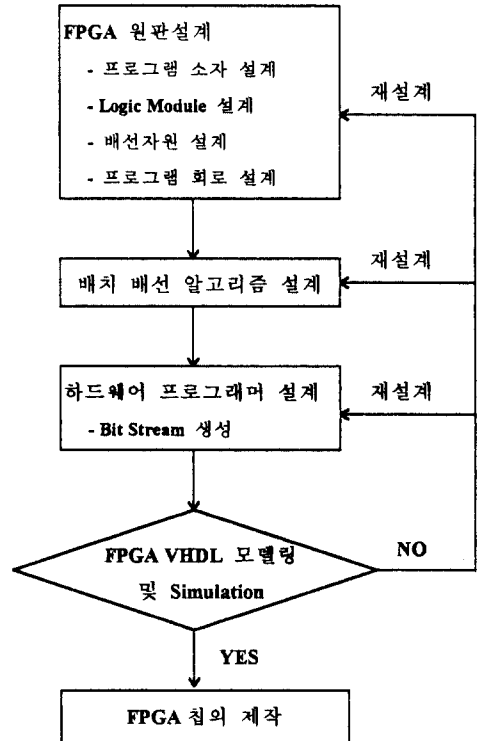


그림 1. FPGA 검증 시스템의 구조  
Fig. 1. Structure of FPGA Verification System.

본 논문에서는 ETRI FPGA의 설계를 위하여 FPGA 칩의 제조전에 FPGA원판을 검증하는 방법을 기술한다. FPGA 원판은 VHDL과 C 언어로 모델링하였으며, 이 모델링된 FPGA원판은 배치배선 알고리즘에 의하여 생성된 비트열을 입력으로 하여 프로그램 된 안티퓨즈의 위치 정보들과 그때의 구현된 회로의 논리 및 지연시간을 이용하여 FPGA 원판과 CAD 환경을 검증한다. 배선지연시간을 고려하기 위하여 안티퓨즈의 프로그램이 끝난 후에 프로그램된 안티퓨즈의 위치정보를 VHDL 모델로부터 받아 C 프로그램을 이용하여 RC 네트워크를 구성하고, 배선선분의 지연시간을 계산한다. VHDL 모델은 이 계산된 지연시간 정보를 참조하여 타이밍 시뮬레이션을 수행한다. 배선경로

의 지연시간 계산은 SPICE나 Elmore 지연시간 모델<sup>[4]</sup> 및 Horowitz 지연시간 모델<sup>[5]</sup> 등을 이용할 수 있다. 본 논문은 상기의 세가지 부분을 모두 구현하여 설계자가 검증시간과 정확도의 필요에 따라 배선경로에 대한 지연시간 모델을 선택하여 사용할 수 있게 한다. 또 시뮬레이션 결과로부터 임계 경로를 찾아 배치 배선기로 피드백 시킴으로 최적화된 회로를 구현할 수 있게 한다.

## II. FPGA의 배선지연시간 모델링

본 논문에서 모델링하는 FPGA 원판의 구조는 그림 2와 같다. FPGA의 원판 및 하드웨어 프로그래머는 VHDL로 모델링되어 있으며, 하드웨어 프로그래머에서 비트열을 입력으로 받아 FPGA 원판에 클럭과 동기하여 보낸다. FPGA 원판은 입력된 비트열을 이용하여 안티퓨즈를 프로그램 한다. 프로그램된 안티퓨즈의 정보는 화일에 저장되며, C 프로그램은 이 정보를 이용하여 RC 네트워크를 구성하고 원하는 지연시간 모델을 이용하여 배선경로의 지연시간을 계산한다. 계산된 지연시간은 화일형태로 VHDL 원판 모델에서 참조하여 고려하게 된다. 배선경로의 지연시간이 VHDL 모델에 입력되면 논리모듈과 입출력 패드를 포함한 타이밍 시뮬레이션을 행한다.

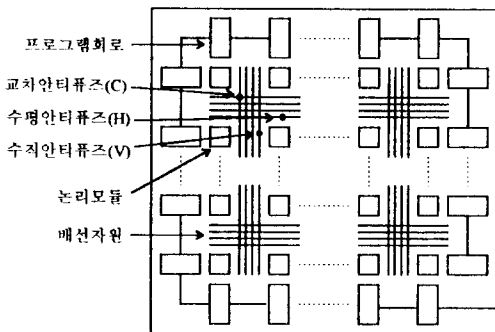


그림 2. ETRI FPGA 원판의 구조  
Fig. 2. Structure of ETRI FPGA Base Array.

그림 3은 임의의 회로에서 선택된 14개의 단락된 안티퓨즈를 지나는 배선경로에 대한 실험이다. 이는 안티퓨즈의 저항 변화에 따라 얻어지는 배선경로의 지연시간의 변화를 나타낸 것으로 수평축은 안티퓨즈의 단락 저항의 값을 나타내며, 수직축은 배선경로의 지연시간과 논리모듈의 지연시간의 비율을 나타낸다. 그림 3에

서 보는 바와 같이 단락 저항의 값이 커질수록 배선경로의 지연시간은 증가하며, 저항의 값이 250 ohm 일 때 배선경로의 지연시간은 논리모듈의 지연시간과 같다. 따라서, FPGA 칩 및 전용 CAD 시스템의 설계시 배선지연시간의 고려가 없이는 정확한 설계에 대한 검증이 이루어진다고 할 수 없으며, 배치배선 알고리즘에서 배선경로에서 설계되는 회로의 클럭 주파수에 대하여 사용가능한 안티퓨즈의 허용 갯수를 알 수 없다.

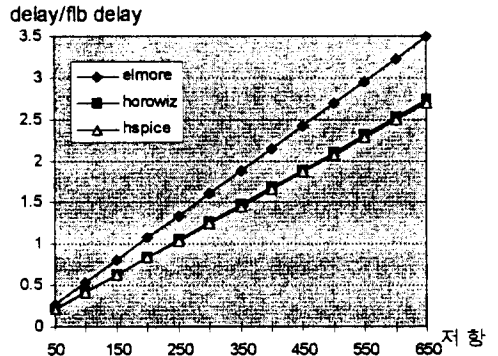


그림 3. 안티퓨즈의 저항변화에대한 임의의 배선경로의 지연시간  
Fig. 3. Delay Time of Routing Path for on-resistance variation of Anti-fuse.

본 논문에서는 배선경로의 지연시간을 고려하기 위하여 FPGA의 각 소자들을 VHDL로 모델링 하였으며<sup>[3]</sup>, 지연시간의 모델은 C 언어를 이용하여 구현 하였다. VHDL로 구현된 FPGA 원판은 프로그램된 안티퓨즈의 정보를 C 언어로 구현된 지연시간 계산함수에 전달하고, 계산된 지연시간을 C 함수로부터 받아 FPGA에대한 시뮬레이션을 행한다. VHDL과 C언어와의 정보 교환은 VHDL의 foreign attribute를 이용하여 행할 수 있다. 그림 4는 C언어와 인터페이스를 위한 VHDL 프로그램을 보여주며, 그림 5는 VHDL 프로그램과 인터페이스를 위한 C 프로그램을 보여준다. C 언어를 이용하여 구현하고자 하는 설계에 대하여 entity를 VHDL로 기술하고 architecture 상에서 foreign attribute를 이용하여 사용되는 프로그램을 명시한다. 그림 4의 entity에서 ready 입력 단자는 C 프로그램이 실행되도록 하는 신호선을 나타내며, done 출력 단자는 C 프로그램의 실행이 끝났음을 나타내는 신호선이다. 여기에서 cal\_delay.o는 지연시간을 계산하는 C 프로그램의 오브젝트 화일이며, cal\_delay\_init은 처음으로 불러지는 C 프로시저 이

름이다. 그림 4와 그림 5에서 나타나는 예는 Mentor Graphics의 QuickVHDL을 이용한 예이다.

```

library WORK;
use WORK.fpga_logic.all;
entity cal_delay is
    port(
        ready    : in bit;
        done     : out bit);
end;
architecture only of cal_delay is
    attribute foreign : string;
    attribute foreign of only : architecture is
        "cal_delay_init cal_delay.o";
begin
end;
    
```

그림 4. C 언어와 인터페이스를 위한 VHDL 프로그램  
Fig. 4. VHDL Program for Interfacing with C.

```

do_cal_delay(ip)
    inst_rec *ip;
{
    intvll;
    intresult;

    vall = mti_GetSignalValue(ip->ready);
    if (vall != 1)
        return;
    cal_fib_delay();
    result = 1;
    mti_ScheduleDriver(ip->done, result, 0,
        MTL_INERTIAL);
}
cal_delay_init(region, param, generics, ports)
    regionID region;
    char *param;
    interface_list *generics;
    interface_list *ports;
{
    inst_rec*ip;
    signalIDoutp;
    processIDproc;
    ip=(inst_rec*)MALLOC((unsigned)sizeof(inst_rec));
    mti_AddRestartCB(free, ip);
    ip->ready = mti_FindPort(ports, "ready");
    outp = mti_FindPort(ports, "done");
    ip->done = mti_CreateDriver(outp);
    proc=mti_CreateProcess("p1", do_cal_delay, ip);
    mti_Sensitize(proc, ip->ready, MTL_EVENT);
}
    
```

그림 5. VHDL과 인터페이스를 위한 C 프로그램  
Fig. 5. C Program for Interfacing with VHDL.

상용의 VHDL 시뮬레이터는 VHDL과 C언어와 같은 프로그래밍 언어와의 인터페이스를 위하여 전용의 인터페이스 함수를 제공한다. 이러한 함수는 특정문자

로 시작하며, 그림 5의 예에서는 "mti\_"로 시작하는 함수이며, 다른 것은 사용자가 정의한 함수이다. VHDL에서 불리어지는 함수 cal\_delay\_init()에서 사용하는 mti\_FindPort() 함수는 주어진 이름의 입출력 단자에 대한 정보를 얻는다. mti\_CreateDriver()는 입출력 단자를 출력용으로 만들어 준다. mti\_CreateProcess() 함수는 새로운 프로세스(process)를 생성하는 것으로 활성(active)될 때마다 부르는 함수(do\_cal\_delay())를 등록시킨다. mti\_Sensitize() 함수는 프로세스에 sensitivity signal을 등록시킨다. 프로세스에서 불리는 함수 de\_cal\_delay() 내에서 mti\_GetSignalValue() 함수는 해당 신호선의 값을 return한다.

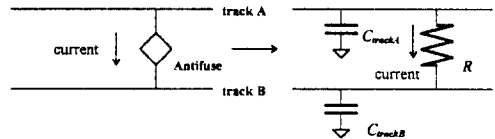


그림 6. 단락된 안티퓨즈와 배선의 RC 모델  
Fig. 6. Programmed Anti-fuse and RC Model of Routing Path.

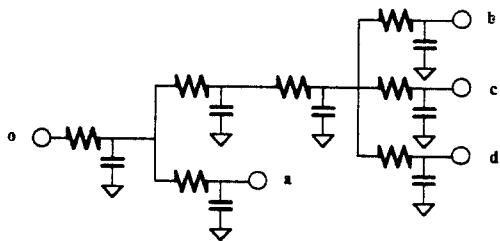
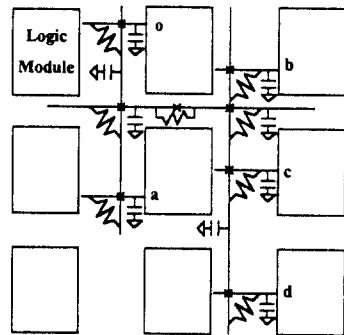


그림 7. 배선경로의 RC 네트워크  
Fig. 7. RC Network of Routing Path.

지연시간을 계산하기 위하여 배선경로는 RC 네트워크로 모델링이 된다. 그림 6과 같이 단락된 안티퓨즈는 저항으로 대체되며 선분은 정전용량으로 대체된다. 이

와 같은 방식으로 모든 패스에 대하여 배선경로를 RC 네트워크로 모델링할 수 있다. 이때 R과 C의 값은 FPGA의 안티퓨즈 소자의 구조나 배선구조에 따라 달라질 수 있다. 그림 7에 임의의 프로그램된 안티퓨즈에 의하여 생성된 배선경로를 RC 네트워크로 모델링한 예를 나타내었다. 모델링된 RC 네트워크의 입력에서 출력까지의 지연시간을 계산하기 위하여 본 논문에서는 Elmore, Horowitz, Spice 등의 지연시간 모델을 이용하였다.

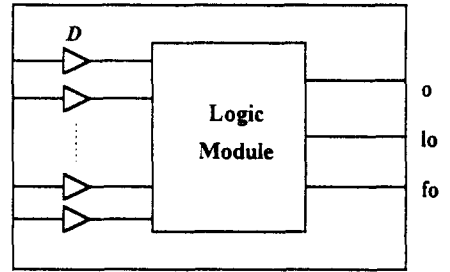


그림 8. 배선경로의 지연시간을 고려한 논리모듈  
Fig. 8. Logic Module including Delay of Routing Path.

```

use STD.textio.all;
library WORK;
use WORK.fpga_logic.all;
entity flb is
generic(i      : integer := 0;
        j      : integer := 0);
port ( a      : in fpga_logic := 'Z';
       b      : in fpga_logic := 'Z';
       .....
       set    : in fpga_logic := 'Z';
       sel    : in fpga_logic := 'Z';
       o      : out fpga_logic := 'Z';
       lo     : out fpga_logic := 'Z';
       fo     : out fpga_logic := 'Z';
       ready  : in fpga_logic := 'Z');
end flb;
architecture flb of flb is
constant fname : string := "delay/flb" & To_String(i) & "_" & To_String(j) & ".dly";
.....
begin
    ma <= transport a      after t_a;
    mb <= transport b      after t_b;
    .....
    mset <= transport set  after t_set;
    msel <= transport sel  after t_sel;

    P0: process (ready)
        file fff : text;
        variable VALUE : integer;
        variable IN_LINE : line;
    begin
        if (ready'event and ready='1') then
            FILE_OPEN(fff, fname, READ_MODE);
            while not(ENDFILE(fff)) loop
                readline(fff, IN_LINE);
                read(IN_LINE, VALUE); t_a <= VALUE * 1 ps;
                read(IN_LINE, VALUE); t_b <= VALUE * 1 ps;
                .....
                read(IN_LINE, VALUE); t_set <= VALUE * 1 ps;
                read(IN_LINE, VALUE); t_sel <= VALUE * 1 ps;
            end loop;
            FILE_CLOSE(fff);
        end if;
    end process P0;

    P1: process (ma,mb,mc,md,mdb,me,meb,mf,mg,mgb,mh,mhb,mdin,msel)
        -- 논리모듈에대한 기술
        .....
    end process P1;
end flb;
    
```

그림 9. 논리모듈의 VHDL 모델  
Fig. 9. VHDL Model of Logic Module.

지연시간 모델을 이용하여 계산된 배선경로에 대한 지연시간은 VHDL로 모델링된 FPGA원판에서 사용하기 위하여 그림 8에서와 같이 논리모듈의 각 입력 단자에 해당하는 배선경로의 지연시간을 가지는 버퍼를 부가함으로 고려할 수 있다. 버퍼의 지연시간을 논리모듈 내에서 타이밍 시뮬레이션을 행하기 전에 읽어 transport 지연시간 모델 형태로 고려하여 이용한다. 이에 대한 논리모듈의 VHDL 표현을 그림 9에 나타내었다.

### III. 실험 및 검증결과

ETRI에서 개발하는 FPGA는 8000 gates 급과 2000 gates 급의 두가지가 있다. 8000 gates 급 FPGA 는 16x32의 논리모듈을 가지며, 313264개의 안티퓨즈를 가진다. 2000 gates 급 FPGA는 8x16의 논리모듈을 가지며, 78368개의 안티퓨즈를 가진다. FPGA 원판의 VHDL 모델의 검증을 위하여 인버터 체인(inverter chain), 4 비트 BCD 카운터 등 여러 가지 회로에 대하여 실험을 수행 하였다. 한 예로 배치 배선기와 비트열 생성기를 통해 출력된 4 비트 BCD 카운터의 프로그램 할 안티퓨즈 수는 약 150개 정도이다. 그림 10은 비트열을 입력으로 하여 FPGA 원판의 안티퓨즈를 프로그램 한 결과로 안티퓨즈가 단락된 시간과 종류 및 위치정보를 출력한다. 이 정보는 배치 배선기에서 출력된 프로그램 될 예상 안티퓨즈 화일과 일대일 비교하여 검증된다. 안티퓨즈의 종류를 나타내는 'C'는 고차 안티퓨즈를, 'H'는 수평 안티퓨즈를, 'V'는 수직 안티퓨즈를 의미한다. ETRI FPGA를 설계하는 과정에서 본 VHDL 모델을 이용하여 FPGA의 검증과정을 수행하였으며, 이의 결과로 비트열 생성기 및 배치 배선 알고리즘 등의 재설계를 행하였다. 그림 11은 검증이 완료된 상태의 4 비트 BCD 카운터의 안티퓨즈를 프로그램 한 후 논리 시뮬레이션 결과이다. mode 의 값이 '1'인 상태에서는 안티퓨즈들을 프로그램 하는 상태를, '0'의 값은 보통의 논리동작 상태를 나타낸다. BCD 카운터 회로의 입력 및 출력으로 사용한 패드는 rapd(4), clka, tpad(6), lpad(8), bpad(7), rapd(7)이다. rapd(4)와 clka는 입력으로, 각각 reset 과 clock을 나타낸다. tpad(6), lpad(8), bpad(7), rapd(7)은 출력으로, 하위 비트로부터 상위 비트를 나타낸다.

1535 ns <C(4, 17, 2, 1)>  
 5345 ns <C(4, 0, 3, 2)>  
 9155 ns <C(9, 4, 2, 2)>  
 12965 ns <C(9, 4, 2, 1)>  
 16775 ns <C(0, 5, 2, 2)>  
 35825 ns <C(4, 11, 3, 16)>  
 39635 ns <C(4, 11, 18, 4)> <C(5, 11, 18, 8)>  
 43445 ns <C(4, 11, 18,13)> <C(5, 11, 18, 29)>

489215 ns <H(16, 37, 4, 0)>  
 493025 ns <H(16, 37, 5, 0)>  
 496835 ns <V(4, 12, 1, 0)>  
 569225 ns <V(6, 5, 4, 0)>  
 571500 ns Hardware Programmer Transmission Complete

그림 10. 안티퓨즈 프로그램의 결과  
 Fig. 10. Result of Programming Antifuses.

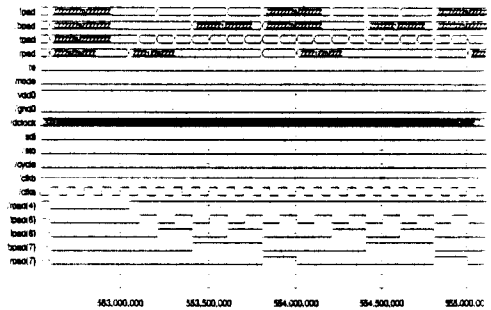


그림 11. FPGA원판의 VHDL 시뮬레이션  
 Fig. 11. VHDL Simulation Result of FPGA Base Array.

표 1은 FPGA 원판의 검증을 위하여 사용한 몇가지 회로에대한 실험결과를 보여준다. 여기에서 안티퓨즈의 수는 프로그램된 전체 안티퓨즈의 수를 나타낸다. 실행 시간은 SUN Sprac 10에서의 Mentor의 QuickVHDL을 이용하여 실행시 user time과 system time의 합을 나타낸 것이다. 이는 원판상에서 안티퓨즈를 프로그램하고 논리 및 타이밍 시뮬레이션을 마친 시간을 나타낸다.

표 1. 여러가지 회로에 대한 시뮬레이션 수행 시간  
 Table 1. Simulation Time for Several Circuits.

회로이름	안티퓨즈의 수	실행시간(sec)
BCD 카운터	150	3:51.0
인버터 체인	174	4:37.6
디코더	492	11:34.8
주파수 카운터	2010	43.42.8

본 논문에서 사용한 3가지 지연시간 모델에 대한 정확도를 그림 12에 나타내었다. 그림 12의 가로축은 임의의 회로에서 추출된 배선경로를 나타내며, 세로축은 배선경로에 대한 각 지연시간 모델을 적용한 배선지연시간을 나타낸다. 여기서 SPICE 시뮬레이션의 결과에 비교하여 Elmore 지연시간 모델은 항상 과대 평가의 결과를 나타내며, Horowitz의 결과는 SPICE 시뮬레이션의 결과에 근접한 결과를 나타내지만, 경우에 따라서 과소 평가의 결과를 나타내는 경우가 있다. 이러한 과소 평가의 결과가 나오는 경우 설계자의 오류가 나올 확률이 크다. 표 2는 그림 12에 나타난 각 배선경로에 대하여 단락된 안티퓨즈의 수를 나타내었다.

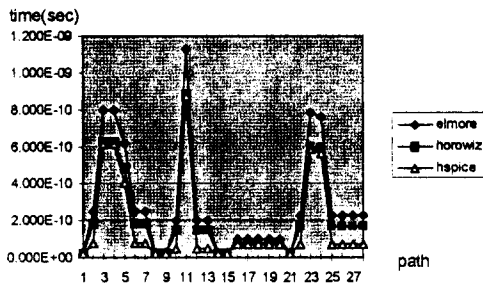


그림 12. 지연시간 모델의 정확도  
Fig. 12. Accuracy of Delay Models.

표 2. 배선경로에 대하여 단락된 안티퓨즈의 수

Table 2. Number of Programmed Antifuses in Routing Path.

Path	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Antifuse 수	2	2	7	7	5	2	2	2	2	2	14	2	2	2
Path	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Antifuse 수	2	2	2	2	2	2	2	2	2	7	7	2	2	2

IV. 결 론

본 논문에서는 안티퓨즈 방식의 FPGA를 이용한 디지털 시스템 설계시 FPGA 칩 및 전용 CAD 시스템의 검증에 위한 VHDL 모델링 기법에 대하여 논하였다. FPGA 원판에 구현된 논리회로는 논리모듈과 이들을 연결하는 배선경로들로 구성된다. 이 배선경로들은 프로그래밍된 안티퓨즈와 선분화된 트랙의 구성에 따라 배선경로의 지연시간이 달라진다. 본 논문은 FPGA 칩의 정확한 검증을 위하여 FPGA 원판의 동작은 VHDL을

이용하여 모델링하였고, FPGA의 배선경로의 지연시간은 SPICE, Elmore 지연시간 모델 및 Horowitz 지연시간 모델 등을 이용하여 고려하였다. FPGA 원판의 동작을 모델링하기 위하여 프로그램 회로, 논리모듈, 배선선분, 안티퓨즈 및 입출력 패드 등을 VHDL로 모델링 하였으며, 배선지연시간의 계산을 위하여 프로그램된 안티퓨즈의 위치정보로부터 배선경로를 재구성하고, 여러가지 지연시간 모델을 이용하여 배선경로의 지연시간을 계산하는 부분은 C 언어로 구현하였다. 구현된 FPGA 회로의 논리 및 타이밍검증을 위하여 추출한 지연시간을 VHDL 시뮬레이션 시에 사용할 수 있도록 하였다.

참 고 문 헌

- [1] S. D. Brown, et al., *Field-Programmable Gate Array*, Kluwer Academic Publishers, 1992.
- [2] Hanjin Cho, et al., A New Field Programmable Gate Array: Architecture and Implementation, ETRI Journal, vol. 17, No. 2, pp.2-30, July 1995.
- [3] 백영석, 조한진, 박인학, 김경수 안티퓨즈 FPGA의 VHDL 모델, 대한전자공학회 하계종합학술대회논문집 vol. 18, pp.320-323, 1995
- [4] W. C. Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wide-band Amplifiers", Journal of Applied Physics, vol. 19, no. 1, pp. 55-63, Jan. 1948.
- [5] J. Rubinstein, et al., "Signal Delay in RC Tree Network", IEEE Trans. on CAD, vol. 2, no. 3, pp. 202-211, July 1983.
- [6] Barry K. Britton, et al., Optimized Reconfigurable Cell Array Architecture for High-performance Field Programmable Gate Arrays, IEEE 1993 Custom Integrated Circuit Conference, Sandiego, CA, pp.7.2.1-7.2.5, 1993.
- [7] Abbas El Gamal, et al., An Architecture for Electrically Configurable Gate Arrays, IEEE J. of Solid State Circuits, vol. 24, no. 3, pp. 394-398, 1989.
- [8] David marple, An MPGA-like FPGA, IEEE

Design & Test of Computers, pp.51-60,  
Dec. 1992.

*Description Language*, Kluwer Academic  
Publishers, 1989.

[9] R. Lipsett, et al., *VHDL:Hardware*

저 자 소 개



白 瑛 錫(正會員)

1963년 1월 14일생. 1985년 2  
월 한양대학교 전자공학과 졸업.  
1987년 2월 한양대학교 대학원  
전자공학과 졸업(공학석사)  
1989년 9월 ~ 현재 한국전자통

신연구소 자동설계연구실 선임연구원. 주관심분야는  
VLSI CAD, VHDL 모델링, 상위수준 합성 등임.

曹 漢 鎮(正會員) 第 32卷 10號 A編 參照

朴 仁 學(正會員) 第 32卷 10號 A編 參照

金 景 洙(正會員) 第 33卷 1號 A編 參照