

FPGA를 이용한 logic tester의 test sequence control chip 설계 및 검증

*강창현, *최인규, *최창, **한혜진, *박종식
*경북대학교 전자공학과, **울산기능대 정보통신설비과

Test sequence control chip design of logic test using FPGA

*Chang-hun Kang, *In-kyu Choi, *Chang Choi, **Hye-jin Han, *Jong-sik Park
*Department of Electronics Engineering, Kyungpook National University
**Department of information and data communication, Ulsan polytechnic college

Abstract - In this paper, I design the control chip that controls inner test sequence of Logic Tester to test chip. Logic tester has the thirteen inner instructions to control test sequence in test. And these instructions are saved in memory with test pattern data. Control chip generates address and control signal such as read, write signal of memory.

Before testing, necessary data such as start address, end address, etc. are written to inner register of control chip. When test started, control chip receives the instruction in start address and executes, and generates address and control signals to access tester's inner memory. So whole test sequence is controlled by making the address and control signal in tester's inner memory.

Control chip designs instruction's execution blocks, respectively. So if inner instruction is added from now on, a revision is easy. The control chip will be made using FPGA of Xilinx Co. in future.

1. 서 론

집적회로(IC)의 복잡도와 집적도가 높아짐에 따라 제조된 chip의 기능 테스트(Function test)와 제조 공정상의 오류로 인한 고장 테스트를 하는 장비의 중요성이 증가하였다. Logic tester는 chip test 장비로써 SSI 급으로부터 VLSI에 이르기까지 다양한 디지털 IC의 성능을 테스트하여 생산자가 소비자에게 보증하여 주는 제품사양을 만족하는가의 여부를 판정하고 합격, 불합격 및 IC의 성능별 등급여부를 판정하여 준다.

패키지 IC에 대해서 테스트를 통하여 고장을 알아낼 수 있는 유일한 방법은 주입력(primary input)(패키지 상태의 소자입력)들에 테스트 신호(test pattern)들을 인가하고 주출력(primary output)(패키지 상태의 소자출력)들을 관찰하는 것이다. 그리고 순차논리회로를 가진 IC의 테스트는 먼저 알려진 상태(known state)에 놓여 있어야 한다.

본 논문에서는 logic tester에서의 테스트 순서를 제어하는 chip을 설계하고 그 기능을 검증하는 것에 대하여 다루고 있다. Logic tester는 내부에 13 종류의 명령어가 저장된 instruction data 메모리와 test vector에 해당되는 pattern data 메모리, 그리고 테스트 결과를 저장하는 Pass/Fail data 메모리가 있다. 본 논문에서 설계한 제어 칩은 logic tester의 내부 명령어를 수행하여 test vector의 순서를 제어함으로써 좀더 효율적인 테스트를 가능하게 하고 Counter와 같은 순차 논리회로를 가진 IC 테스트의 경우 원하는 출력이 나올 때까지 테스트 시작을 지

연시켜 테스트를 할 수 있도록 한다.

2. 본 론

2.1 Logic Tester의 전체 개요

본 논문에서 제시한 test sequential control chip의 구성 및 동작에 대한 이해를 돕기 위해서 control chip이 포함될 Logic tester의 전체 구성 및 동작을 살펴보자.

2.1.1 Logic tester의 특징 및 구성

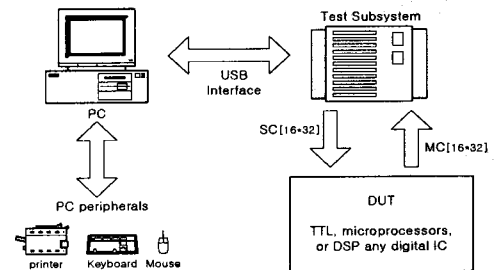


그림 2-1 Logic Tester의 개요도

Logic tester는 PC와 test subsystem 구성된다. PC는 windows환경에서 작동되며 test subsystem의 감시 및 제어를 하는 것과 동시에 test vector를 test subsystem을 통하여 DUT(Device Under Test)에 인가하고 이로부터 측정된 결과를 분석하는 기능을 수행한다. Test subsystem은 최대 10장의 test board를 가질 수 있으며 board 1장당 16개의 테스트 가능한 I/O pin 제공한다. 그리고 최대 12.5MHz에서 최소 0.19Hz까지 test clock(MTG)과 50MHz의 FPGA 자체 clock이 사용되어 20ns의 timing resolution을 가진다. 두 개의 testing mode가 있으며 13개의 내부 명령어가 있다.

각 테스트 board는 testing chip과 memory, control chip, 그리고 mux&decoder block으로 구성되어 있다.(그림 2-2참조) Memory는 동시에 access하는 64k depth의 instruction data memory(IDM), expect data memory (EDM), pattern data memory(PDM) 3개와 256k depth of the test result저장하는 Pass/Fail data memory(P/FDM)로 구성되어 있다.

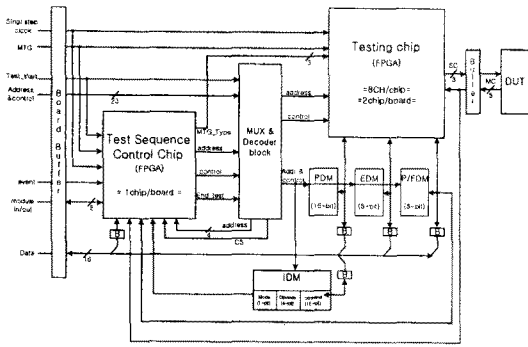


그림 2-2 Testing board의 내부구성도

2.1.2 Logic tester의 동작

테스트 전에 필요한 데이터를 PC로부터 입력받아 메모리와 각 chip의 내부 레지스터에 저장한다. 테스트가 시작되면 testing chip은 test sequence control chip에서 생성된 address와 control signal에 따라 memory로부터 data를 읽어들이어 test 하고자 하는 DUT에 source signal (test vector)을 인가하고, DUT가 출력하는 monitor signal을 받아들이어 오동작 여부를 분석해서 memory에 저장한다.

테스트 시작과 동시에 address와 control 신호를 만들어 내는 Test sequence control chip은 내부 레지스터에 테스트 전에 CPU로부터 data를 입력받아 저장하기 위해 4bits의 address bus와 16bits의 data bus를 가지고 있다. 그리고 매 cycle 마다 필요한 instruction data를 받기 위한 21bits의 전용 data bus(Instruction data bus)를 가지고 있다. 다른 depth를 가진 메모리를 access 하기 위해 두 개의 address bus(16bits, 18bits)와 control signal을 출력하며, 명령어 수행 시 board간의 데이터를 주고받기 위한 I/O pin을 가지고 있다.

MUX&Decoding Block은 test board와 CPU 사이를 인터페이스 하는 역할을 한다. 먼저 decode block은 테스트가 시작되지 않았을 때는 CPU로부터 address를 받아 decoding 한 뒤에 chip selection signal을 출력하게 된다. 그러나 test가 시작되면, CPU의 address와는 상관없이 모든 memory의 chip selection signal을 enable 시킨다. Muxing Block은 CPU와 control chip으로부터 생성되는 control signal과 address를 muxing하는 역할을 한다.

2.2 Test sequence control chip의 설계

Test sequence control chip은 IDM에서 instruction을 읽어 와서 decoding 하여 address와 control signal을 만들어 낸다. 이 때 만들어진 Address와 control signal로 메모리를 access 하여 logic tester의 테스트 순서를 제어 하는 기능을 한다

2.2.1 Test sequence control chip 전체 구조

Instruction은 fetch와 execution부분을 pipe line 형태로 하기 위해 instruction buffer는 2단으로 구성하였다. Instruction은 이러한 이중 버퍼를 거쳐 generator block에서 실행한다. 내부는 test mode에 따라 clock을 만들어 내는 clock decoder와 예러 발생 여부 및 외부 이벤트 발생을 감지하는 event detector, 테스트하기 전 필요한 데이터를 저장하는 special register, 서브루틴 호출이나 루프를 수행할 때 현재의 data를 저장하는 stack memory 부분, 그리고 실제 address 및 control 신호를 만들어 내는 generator 부분으로 구성하였다.(그림 2-3참조) 이들 각 부분들의 구조 및 동작에 대해서 자세히 설명하겠다.

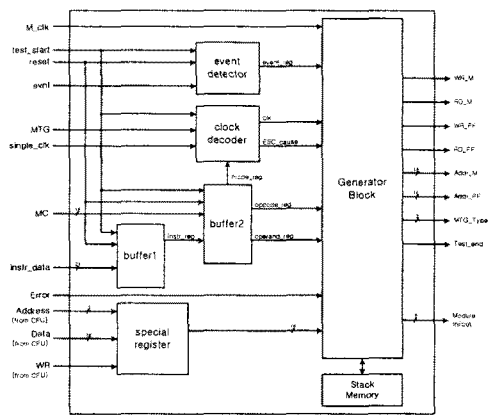


그림 2-3 Control chip의 전체 블록도

2.2.1 Instruction buffer

Buffer는 이단 구조로 clock decoder의 출력신호인 clock에 동기화 하였다. 또한 instruction 수행 중에 각 buffer를 제어하기 위해서 enable 신호를 추가하였다.(그림 2-4참조)

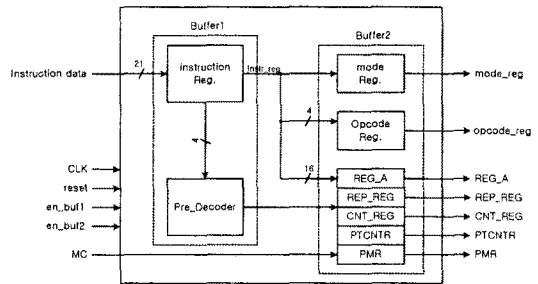


그림 2-4 Instruction buffer module

첫 번째 버퍼에서 opcode에 해당되는 4bit를 먼저 decoding 하여 다음 버퍼에서의 register에 selection 신호를 보낸다.(표2-1참조) 두 번째 버퍼에서는 instruction data를 각 기능별로 register에 저장시켜 generator block으로 보내는 역할을 한다. Instruction의 MSB(instr_reg[20])는 test mode를 나타내는 것으로 clock decoder 부분으로 보내어진다. 이렇게 하는 이유는 test mode를 처음에 고정시키는 것이 아니라 원하는 시점에서 mode를 선택할 수 있게 하기 위해서이다.

표 2-1 Pre-decoder에 의한 buffer2의 register 설정

Instruction	instr_reg [19-16]	Buffer2의 register
Repeat	0110	REP_REG <= instr_reg[15-0]
Loop	0111	CNT_REG <= instr_reg[15-0]
Pattern match	1011	PMR <= MC
Pulse train pattern match	1100	PMR <= MC PTCNTR <= instr_reg[15-0]
others	others value	REG_A <= instr_reg[15-0]

2.2.2 Generator block

이 블록에서는 버퍼의 출력 opcode와 operand, testing clock과 FPGA 자체 clock, 그리고 내부 레지스터에 저장된 값을 이용하여 instruction을 수행하여 외부 메모리를 제어하기 위한 address 신호와 control 신호를 만들어

낸다. 테스트 시작과 끝에 필요한 동작을 위한 블록과 디코딩 블록, 그리고 13개의 instruction을 수행하는 블록으로 구성되어 있다.(그림2-5참조)

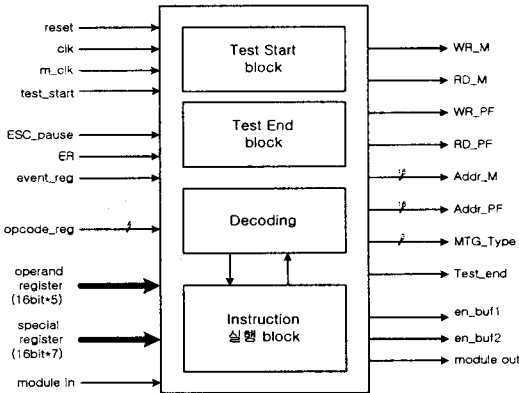


그림 2-5 Generator module

먼저 테스트 시작 블록은 테스트가 시작되었을 때 start address가 2단 버퍼를 거쳐 이 블록에 입력되기까지 테스트를 보류하는 역할을 한다(instruction buffering). Test end 블록은 현재 출력되는 address값이 end address와 같은지 확인하여 end_test 신호를 만들어 테스트를 종료시키는 역할을 한다. 그러나 Pass/Fail 값은 테스트 결과 값이므로 1 clock후에 memory에 저장되므로 1 clock delay 시켜 end_test 신호를 만들어야 한다.

Decoding 블록은 opcode를 decoding하여 명령어 수행 블록에 enable 신호를 인가하고 명령어 수행이 끝났음을 알리는 신호를 받아들여 다음 instruction을 decoding한다. 마지막으로 명령어 수행 블록은 총 13개의 독립된 블록으로 구성되어 있으며 각각 하나의 instruction을 수행한다. 이들 블록은 많은 clock이 소비되는 복잡한 명령어 수행을 위해서 testing clock 뿐만 아니라 FPGA 자체 clock(50MHz)을 사용한다.

2.2.1 기타 블록

Clock decoder는 mode에 따른 clock 신호와 pause 상태를 벗어나는 ESC_pause 신호를 발생시킨다. Mode 결정을 하는 data인 instruction의 MSB가 버퍼 2단을 지나서 입력되어 지므로 초기 clock을 MTG로 잡았다.

표 2-2 Clock decoder의 출력 모드

Reset	mode_reg	mode	Output
0	0	Sequential mode	CLK <= MTG
	1	Interrupt mode	ESC_pause <= single_clk
1	0	Sequential mode	CLK <= MTG ESC_pause <= single_clk
	1	Interrupt mode	CLK, ESC_pause <= single_clk MTG signal : don't care

Testing mode는 외부에서 한 개의 clock 펄스(single_clk신호)를 만들어 주어 이 clock에 맞추어 testing 하는 interrupt mode와 연속적인 clock 신호를 만들어 계속적인 테스트를 하는 sequential mode로 나뉜다. Interrupt mode에서는 user가 test 한 cycle마다의 결과를 볼 수 있어 칩 디버깅에 편리하도록 하였다.

Event detector는 외부에서 발생하는 interrupt 신호등의 유무를 저장하는 register와 테스트 결과 여러 발생 유무를 저장하는 register로 구성한다. 저장된 값은 관련 명령어 수행 후 다시 초기화된다.

테스터하기 전에 instruction 수행에 필요한 데이터를

CPU로부터 입력받아 저장하기 위한 것이 special register module 이다. Starting address와 end address 그리고 match instruction 필요한 데이터를 저장하기 위한 register와 decoding 블록으로 구성되어 있으며 3bits의 address와 16bits의 data bus를 통하여 WR의 falling edge에서 데이터를 각 register에 저장한다.(표 2-3 참조)

표 2-3 Special register의 address mapping
Table 2-3 Special register의 address mapping

Address	data => register
000	SAR(Start Address Reg.)
001	EAR(End Address Reg.)
010	MER(Match Expect dat Reg.)
011	MCR(Match Care bit data Reg.)
100	MCNTR(Matching Count Reg.)
101	BAR(Branch Address Reg.)
110	MPR(Matched Pin data Reg.)

서브루틴 호출과 같은 명령어 수행 중 address 및 data를 저장하기 위해서 칩 내부에 stack memory를 만들어 사용하였다. 크기는 여유롭게 16bits×32로 하였다.

2.3 명령어

테스트의 흐름을 제어하기 instruction은 13개 있으며 decoding 하기 위해서 opcode의 크기를 4bit로 하였다. 각각의 instruction data는 21bits의 크기로 test mode(MSB,I₂₀), opcode(I₁₉-I₁₆), operand(I₁₅-I₀)로 구성되어 있다. (표2-4 참조)

표 2-4 Opcode에 따른 명령어 decoding

Opcode	Instruction	Comment
0000	NOP	NO operation
0001	Pause	Pause test
0010	CH_MTG	Change MTG type
0011	JMP	Unconditional jump
0100	JMPE	Error conditional jump
0101	JMPV	Event conditional jump
0110	RPT	Repeat
0111	LOOP	Start loop
1000	END LOOP	End loop
1001	CALL	Call subroutine
1010	RTN	Return and end subroutine
1011	PM	Pattern match
1100	PTPM	Pulse train pattern match

명령어를 decoding하여 address와 control 신호를 만들어 낸다. 테스트가 실시되고 있는 중에는 P/FDM은 항상 결과를 기록하고 다른 메모리는 저장된 데이터를 읽어온다. 그리고 jump 명령어와 같이 address가 분기되는 경우에는 2clock 동안 buffering 과정을 거친다. 명령어가 메모리에 필요한 부분에만 존재하고 다른 메모리와 동시에 access 되어야 하므로 address만 증가시키는 NOP를 추가하였다. CH_MTG는 테스트 clock을 테스트 수행 중 변화를 주기 위한 명령어이다. 그리고 두 개의 match 명령어가 있는데 match 기능에 대해서 잠시 설명하겠다. DUT의 출력이 임의의 정해진 값과 동일할 때까지 계속 loop를 수행하고 일치하면 loop를 끝내 테스트를 시작하는 기능이 match 기능으로 순차논리회로의 테스트 경우나 테스트 시작시점에서 필요로 한다. Match 명령어는 DUT의 모든 output과 미리 정한 expected pattern을 match시키는 명령어와 DUT의 어느 하나의 pin에서의output pulse train(shift register 값)과 미리 정한 expected pattern train(MED)을 match시키는 pulse train pattern match 명령어가 있다.

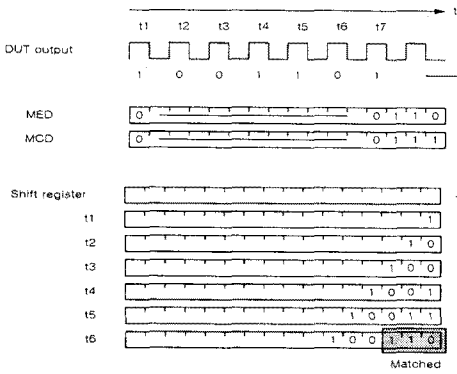


그림 2-6 Pulse train pattern match의 예

위에 그림에서 MCD는 match care data로 '1'이 있는 값에서만 match가 되는지 판단할 수 있게 한 것이다.

2.3 설계된 control chip의 검증

본 논문에서 설계한 chip을 검증하기 위해 Active-HDL을 사용하여 simulation을 하였다. 그림2-7은 테스트 칩 내부 레지스터에 필요한 데이터를 입력하는 부분이다.(test_start='1')

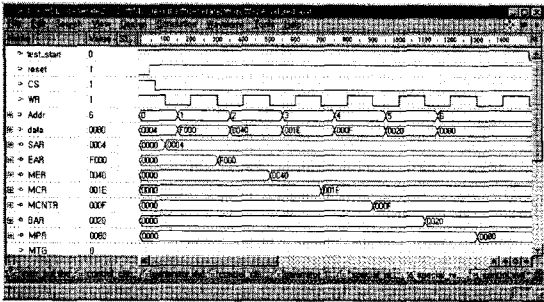


그림 2-7 Simulation 결과(1)

그리고 테스트가 시작된 후 명령어에 따른 address와 control 신호를 검증한 결과를 그림2-9,2-9에 나타내었다. 지면관계로 출력을 시간단위로 나누어 표현하였다. 시작메모리의 address는 4000h이고 jmp 명령어가 처음 실행되고 있음을 보여준다.

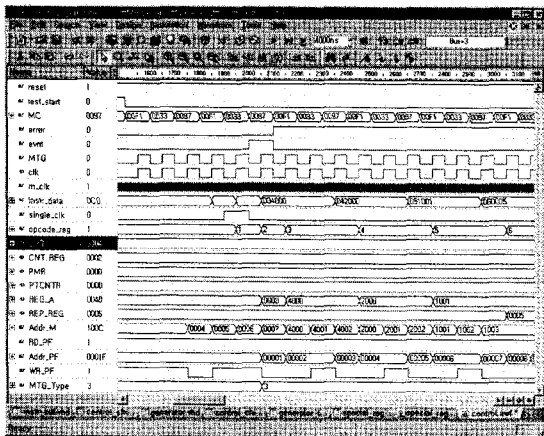
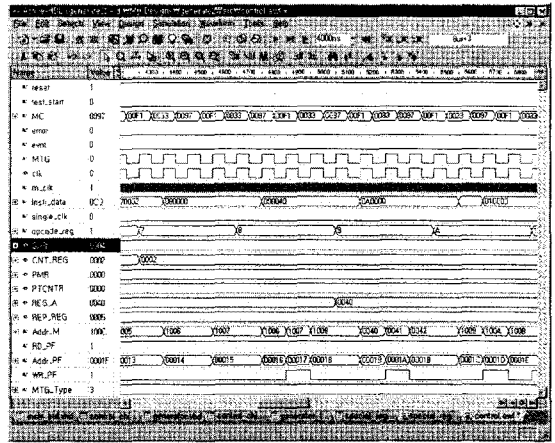


그림 8 Simulation 결과(2)



3. 결 론

본 논문은 logic tester의 내부 명령어에 따른 테스트 흐름을 제어할 수 있는 칩을 설계하고 검증하였다. 설계한 칩은 logic tester의 functional test를 단순히 메모리에 저장된 test vector의 순서대로 하는 것 보다 내부에 명령어를 추가하여 test vector의 순서를 제어하여 좀더 효율적인 테스트를 가능하게 하였다. 그리고 Counter와 같은 순차 논리회로를 가진 IC 테스트를 수행할 수 있으며 각각의 명령어에 따라 명령어 수행 부분을 독립된 block으로 설계하여 향후 logic tester 내부 명령어를 추가할 경우 수정이 용이하도록 하였다.

본 논문에서 설계한 Chip은 VHDL로 코딩하였으며 active-HDL tool을 사용하여 function 시뮬레이션을 하였다. 향후 설계된 칩은 Xilinx社의 Spartan-II FPGA를 사용하여 구현할 것이다. 따라서 시뮬레이션은 칩을 실제 구현할 경우 동작이 확실히 되도록 하기 위해 Xilinx社 자체 프로그램인 Foundation series를 이용하여 gate delay까지 고려한 timing simulation을 할 예정이다. 그리고 Logic tester의 완전한 control chip을 구현하기 위해서는 FPGA의 입출력 pin수의 제한 때문에 빠진 MUX &Decode 부분을 control chip의 내부에 들어갈 수 있도록 하는 연구가 필요하다.

[참 고 문 헌]

- [1] V. P. Nelsin, H. T. Nagle, B. D. Carroll, and J. D. Irwin, "Digital Logic Circuit Analysis and Design," Prentice-Hall International, Inc. 1995.
- [2] N. H. E. Weste and K. Eshraghian, "Principles of CMOS VLSI Design ; A Systems Perspective," Addison-wesley Publishing Company, 1993.
- [3] A. W. Shaw, "Logic Circuit Design," Saunders College Publishing, 1997.
- [4] R. J. Feugate, JR. Steven, M. McIntyre, "Introduction to VLSI Testing," Prentice Hall, 1988.
- [5] G. Perry, "The Fundamentals of Digital Semiconductor Testing," Soft Test Inc. 1996.
- [6] 박용수, "반도체 VLSI 테스트 기초 및 활용," 히드텍, 2000
- [7] Y.W. Yoon, J.S. Moon, B.M. Moon, S.K. Ahn, and D.K. Jeong, "A study on the Architecture and Logic Block Design of FPGA," 전자공학회 논문지, pp.140-151, 1996. 11