

論文95-32A-5-10

# 디지털 신호처리 프로세서의 테스터블 디자인 기법

## (Testable Design Technique for Digital Signal Processor)

金東奭\*, 金備鎔\*\*, 李起焮\*\*, 崔海旭\*\*\*

(Dong Seok Kim, Bo Kwan Kim, Ki Jun Lee, and Hae Wook Choi)

### 요약

테스터블 디자인 기법은 여러가지가 있으며, 이들 방법 중 Scan path를 이용한 기법과 BIST 기법이 주로 사용된다. 그러나, 이들 기법의 실제 칩에 적용하는 과정에서의 여러 가지 고려사항들에 대한 연구는 많지 않기 때문에, 본 논문에서는 테스터블 디자인 기법의 실제 칩에 구현하는 과정에서 효율성의 증대에 관한 연구를 하였다. 정상 동작시에는 상관 관계가 없는 블록을 테스트 할 때는 공동 사용하여 하드웨어의 증가 없이 테스트 시간을 줄이는 방법과 Scan path를 가장 짧게 구현하기 위해 MUX를 사용하는 방법 및 여러 블록들을 제어하기도 하고 관측하기도 하는 Scan 레지스터를 이용하여 하드웨어의 증가를 억제하는 방법등을 제시한다.

### Abstract

There are many testable design techniques, among which Scan path and BIST techniques are mainly used. In this paper, the increase of design effectiveness is discussed, when these techniques are applied to the practical implementation of chips. The following techniques are presented : 1) Blocks are commonly used to reduce test time without hardware increase, 2) MUX is used to implement the shortest Scan path, 3) Scan register is used which controls and/or observes several blocks to avoid the increase of hardware.

### I. 서론

최근 반도체 기술이 급격히 발전함에 따라 단일 칩 내에 집적할 수 있는 회로 소자의 수가 크게 증가하게 되었다.

회로 소자의 수가 많아짐에 따라 테스트를 위한 장비와 테스트에 소요되는 시간이 증가 하였다. 이러한 증가요인을 줄이기 위해 칩의 설계 단계에서 테스트를 고려하여 설계하는 기술을 테스터블 디자인이라 한다. 테스터블 디자인 기법의 가장 기본적인 요소는 칩의 공정 단계에서 발생할 수 있는 모든 오류를 검출할 수 있어야 한다. 공정상의 모든 오류를 검출하면서 테스트를 빠르게 할 수 있어야하고 테스트를 위한 하드웨어의 양이 적도록 설계하여야 한다. 공정상 발생하는 오류를 검출하기 위해서는 오류가 발생할 수 있는 회로에 테스트 벡터를 인가할 수 있도록 제어성이 높아야 하며 오류의 발생 여부를 알 수 있도록 관측성이 높아야 한다.

\* 正會員, 國防科學研究所

(Agency for Defense Development (ADD))

\*\* 正會員, 忠南大學教 電子工學科

(Dept. Elec. Eng., Chungnam Nat'l Univ.)

\*\*\* 正會員, 韓國電子通信研究所

(Elec. Tele. Research Institute (ETRI))

接受日字: 1994년6월30日, 수정완료일: 1995년5월1일

테스터블 디자인에 관한 연구는 최근 들어 활발하게 진행되었으며 많은 기법들이 발표되었다. 하지만 이론적인 기법들의 실제 칩에 적용하는 방법에 의한 효율의 증대에 대한 연구는 많지 않았다.

본 논문에서는 DSP 칩인 ADPCM Transcoder를 모델로 하여 실제 칩에 구현하는 과정에서 칩면적의 증가를 억제하면서 테스트 시간을 줄이는 몇 가지 방법을 제시한다. 모델로 사용한 칩은 일반적인 디지털 신호처리 프로세서에 사용되는 거의 모든 기능 블록들이 존재하기 때문에 이 칩에 존재하는 기능 블록들에 대한 테스터블 디자인은 다른 프로세서에도 비슷하게 적용할 수 있다. 또한 이 칩은 이미 테스터블 디자인 기법을 적용하여 설계되어 있기 때문에 칩 전체에 대한 테스터블 디자인 기법에 대해 연구하였고 더욱 효율이 높은 설계 기법들을 제시하는 것이다.<sup>[1]</sup>

본 논문에서 제안하는 새로운 테스터블 디자인 기법들은 이론적인 기법들을 실제 프로세서에 구현하는 과정에서 테스터블 디자인 효과를 높이기 위한 방법들이다. 정상 동작시에는 상관 관계가 없는 블록을 테스트 할 때는 공동사용하여 하드웨어의 증가 없이 테스트 시간을 줄이는 방법, Scan path를 가장 짧게 구현하기 위해 MUX를 사용하는 방법, 여러 블록들을 제어하기도하고 관측하기도하는 Scan 레지스터를 이용하여 하드웨어의 증가를 억제하는 방법등을 소개한다.

## II. 프로세서의 분석

모델로 사용된 칩은 ADPCM Transcoder로서 그림 1과 같이 입출력 레지스터와 10 MHz의 속도로 처리 가능한 디지털 신호처리 프로세서로 구성되어 있고 레지스터와 프로세서는 입출력 데이터 버스로 연결되어 있다. 디지털 신호처리 프로세서는 다시 크게 두 블록으로 나뉘어지는데 칩의 제어 기능을 하는 PCU (Program Control Unit) 블록과 데이터의 계산 역할을 하는 ACU (Arithmetic Control Unit) 블록이다.

이 칩에 구현된 테스터블 디자인 기법은 Scan path 기법과 BIST 기법이 모두 사용되었다. 프로세서의 설계시 각각 독립적인 역할을 하는 기능 블록들이 이미 분할되어 있기 때문에 이들을 따로따로 분리해서 테스트를 수행할 수 있도록 전체적인 테스터블 디자인의 구성은 Scan path 기법을 이용하였다. 독립된 기능 블록 중에서 테스트 시간이 많이 걸리는 블록은 부분적으로 BIST 기법을 사용하여 테스트 시간의 단축을 꾀하였다. Scan path 기법을 적용함에 있어서 테스트 시간의 단축을 위해 Multiple Scan Chain을 사용하

였으며 ACU 블록에는 Partial Scan path 기법을 사용하였다. 그림 2는 칩의 전체적인 테스터블 디자인에 의한 회로도이다. 그림 2에서 Macro ROM과 Micro ROM의 테스트를 위해 BIST 기법이 사용되었다.

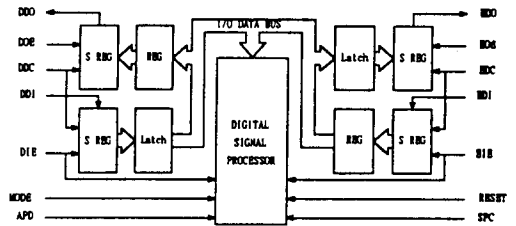


그림 1. 프로세서의 블록도  
Fig. 1. Block diagram of processor.

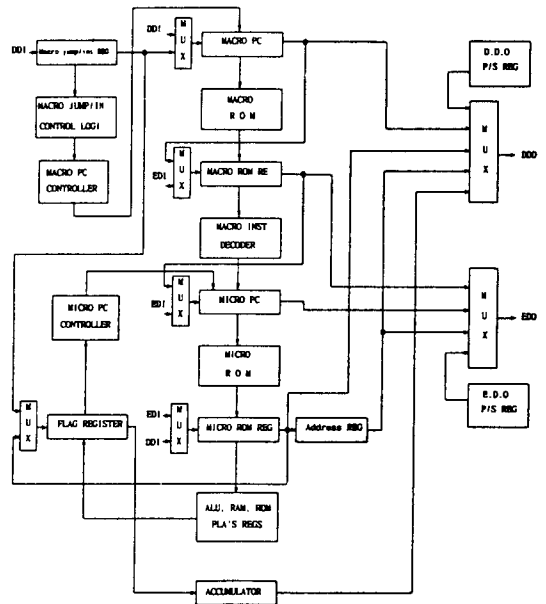


그림 2. Scan path의 구현  
Fig. 2. Structure using Scan path.

### 1. PCU 블록의 테스터블 디자인

PCU 블록은 데이터를 계산하는 ACU 블록을 제어하는 역할을 하는 블록으로 Macro ROM, Micro ROM, Macrocode Instruction Decoder, Micro PC Controller, Macro Jump/Interrupt Control Logic, Address Register의 6개의 작은 블록으로 나뉘어진다. 이중에서 Macrocode ROM과 Microcode ROM은 BIST 기법이 사용되었고 나머지 4개의 블록

은 Multiple Scan path 기법이 사용되었다. 그림 3에 PCU 블록의 테스트블 디자인을 이용한 회로도를 나타내었다. 굵은 선으로 표시된 연결선이 Scan path를 나타내는 것이다.

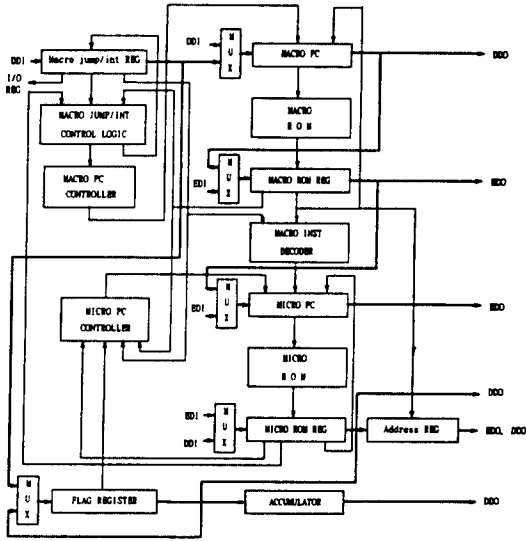


그림 3. PCU의 Scan path 구조  
Fig. 3. Scan path structure of PCU.

1) ROM에 사용된 BIST 기법

프로세서가 Split Control Store 구조를 가지고 있기 때문에 PCU 블록에는 두개의 ROM 이 존재한다. 이 두개의 ROM의 테스트는 3가지 모드로 나누어 테스트하는데 그 첫번째 모드는 진단 테스트에 사용되는 Memory Dump 테스트로 그림 3에 나타낸 Scan path를 통해 원하는 장소의 데이터를 한 번에 읽어내는 모드이다. Scan path를 통해 읽으자 하는 장소의 주소를 각각의 프로그램 카운터에 입력시키고 입력이 다되면 정상 동작을 하여 기록되어 있는 데이터를 읽어 명령어 레지스터에 저장시킨다. 저장이 끝난후에 Scan path를 통해 명령어 레지스터에 저장된 테스트 결과를 출력하고 동시에 다음번 주소를 입력시킨다. 두 번째, 세 번째 모드는 BIST 기법으로 그림 4에 나타낸 것과 같은 구조의 회로로 구현하였다. 두 번째 모드는 "LFSR/MISR" 모드로 LFSR은 테스트 벡터 생성기로 프로세서의 실행속도인 10MHz의 속도로 슈도 랜덤 테스트 벡터를 생성한다. LFSR에 의해 자동생성된 테스트 벡터가 ROM에 인가되어 테스트되어진 후에 테스트 결과는 MISR에 의해 데이터 압축을 하여 Signature를 발생시켜 매 클럭 사이클 마다 외부에서 관측이 가능하도록 출력된다. 세 번째 테스트 모드는

"INC/MISR" 테스트 모드로, 테스트 방법은 "LFSR/MISR" 테스트 모드와 같으며 단지 LFSR 대신 프로그램 카운터를 Incrementer로 사용한다는 것이 다를 뿐이다. 슈도 랜덤 테스트 벡터의 발생을 위해 설계된 두 개의 프로그램 카운터의 특성 방정식을 알아보면 Macrocode ROM의 LFSR의 길이는 7 비트이며 특성 방정식은  $x^7+x^4+1$ 이고, Microcode ROM의 LFSR의 길이는 10 비트이며 특성 방정식은  $x^{10}+x^6+1$ 이다. 테스트 결과의 압축을 위한 MISR의 경우는 Macrocode ROM의 MISR의 길이는 9 비트이고 특성 방정식은  $x^9+x^1+1$ 이며, Microcode ROM의 MISR의 길이는 27 비트이고 특성 방정식은  $x^{27}+x^3+x^2+x^1+1$ 이다.<sup>[5]</sup>

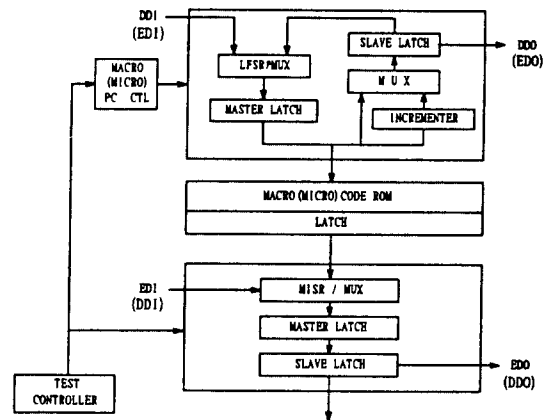


그림 4. ROM의 테스트를 위한 BIST 구조  
Fig. 4. BIST structure for testing ROM.

2) PCU 블록의 PLA의 테스트블 디자인

PCU 블록에서 테스트되어질 블록은 3개의 PLA와 1개의 레지스터가 있다. 이 블록들은 각각 독립적으로 분할된 블록들이기 때문에 Multiple Scan path를 이용하여 독립적인 테스트가 가능하다.<sup>[6]</sup> 그림 3에 나타낸 Scan path를 이용하여 블록들마다 독립적인 테스트를 수행한다.

Macro Jump/Interrupt Control Logic은 DDI 핀 -> Macro Jump/Interrupt Control Logic -> Macro PC -> Macrocode ROM Register -> EDO 핀, EDI 핀 -> Microcode ROM Register -> DDO 핀, Macro Instruction Decoder는 DDI 핀 -> Macro PC -> Macrocode ROM Register -> Micro PC -> EDO 핀, DDI 핀 -> Macro Jump/Interrupt Register, EDI 핀 -> Flag Register -> Accumulator, Micro PC Controller

는 DDI 핀 -> Macro PC -> Macro ROM Register -> Micro PC -> EDO 핀, EDI 핀 -> Micro ROM Register -> Flag Register -> Accumulator -> DDO 핀, Address Register는 EDI 핀 -> Macrocode ROM Register -> Micro PC -> EDO 핀, DDI 핀 -> Microcode ROM Register -> Address Register -> DDO 핀으로 연결되는 Scan path를 갖는다.

각 블록들의 Scan path 들은 대부분 테스트 벡터의 입력 경로로 사용되거나 테스트 결과의 출력 경로로 사용된다. 그러나 모두 데이터의 입출력을 위해 사용되는 것은 아니고 테스트 벡터의 입력이 끝나고 실행될 때 테스트되는 블록과 관계없는 블록들의 오동작을 막아주기 위해 사용되는 경로도 있다. 테스트하는 방법은 Scan path를 통해 외부에서 생성된 테스트 벡터를 입력시키고 정상 동작으로 전환하여 실행시킨다. 실행이 완료되면 테스트 결과의 비교를 위해 Scan path를 통해 외부로 출력된다. 동시에 다음 테스트 벡터가 Scan path를 통해 입력되어 진다.

이와 같이 테스트를 위해 Scan path를 길게 형성하여 모든 블록을 테스트하는 것이 아니고 각 블록의 입출력에 해당하는 레지스터만이 Scan path에 포함되도록 하여 Scan path의 길이를 짧게하여 테스트 제어장치의 구현을 위한 약간의 하드웨어의 증가를 감수하고 테스트 시간을 대폭 감소 시켰다.<sup>110) 111)</sup>

2. ACU의 테스터블 디자인

PCU 블록의 전체 구성은 Scan path 기법에 의한 테스트에 적합하게 파이프라인 레지스터 구조를 갖고 있는데 비하여 ACU 블록은 조합 회로와 조합 회로가 직접 연결되어 데이터의 이동이 이루어지고 있기 때문에 PCU 블록과 같은 테스트를 할 수가 없다. 따라서 각 블록의 가제어성과 가관측성을 높이기 위해 부가적인 회로가 필요하다. 이를 위해 그림 5와 같이 정상 동작시 전혀 쓰이지 않는 테스트 레지스터가 테스트 포인트로써 추가되어 설계 되었다. 테스트 레지스터는 ACU의 테스트시 이미 형성되어 있는 Scan path에 의해 관측할 수 없는 MUX, MULT PIA, SC Latch, MR Latch, Destination Decoder, Source Decoder의 상태를 관측할 수 있게 설계되었다.

ACU의 테스트시 scan path는 역시 multiple scan chain 기법을 이용하여 가장 짧은 scan path를 선택하여 테스트 벡터를 입력시키고 테스트 결과를 출력 시킨다. Scan path는 DDI 핀 -> Flag Register -> Accumulator -> DDO 핀, EDI 핀 -> Microcode ROM Register -> Address Register ->

EDO 핀으로 연결되는 두개의 Scan path가 같이 쓰이고 있다. 두 개의 Scan path를 이용하여 마이크로 명령어 레지스터에 각 블록을 제어하는 테스트 벡터를 입력시키고 테스트 제어장치는 Scan 모드를 실행 모드로 변환하여 테스트 목적 블록이 실행되게 하고 그 결과를 Flag Register와 Accumulator에서 받아 다시 Scan 모드로 바꾸어 Scan path를 통해 외부로 출력시킨다.

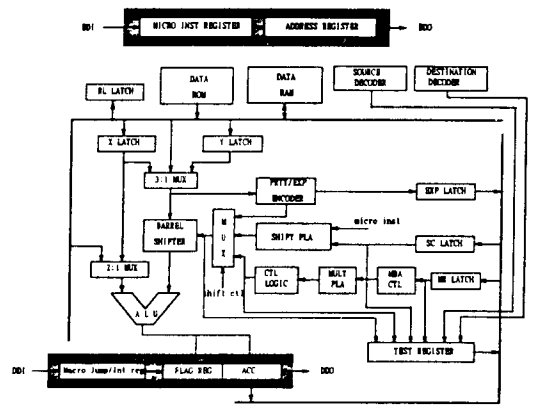


그림 5. ACU의 Scan path 구조  
Fig. 5. Scan path structure for ACU.

1) Data ROM의 테스트

Data ROM의 테스트는 마이크로 명령어 레지스터에 Data ROM의 데이터 값을 읽고 Accumulator에 저장하라는 명령어를 인가해주고 실행 시킨 후 Accumulator에 저장된 데이터 값이 정확한지 Scan path를 통해 외부로 출력하여 비교한다.

2) Data RAM의 테스트

RAM의 테스트는 두가지 방법으로 테스트를 행하는데 한가지는 Memory Array와 주소 디코더의 "Stuck-at-Fault"를 검사하기 위한 것이고 다른 한가지는 저장셀의 "Cell to Cell Leakage"를 검사하기 위한 것이다.<sup>141)</sup>

첫번째 테스트는 Data RAM에 슈도 랜덤 테스트 벡터를 쓰고 다시 그 데이터를 읽어 오류의 발생 여부를 검사한다. 이 때 슈도 랜덤 테스트 벡터를 외부에서 직접 입력시켜 주는 것이 아니고 ACU의 몇몇 블록들이 실행되어 여러 데이터를 만들어 주는 명령어를 테스트 벡터로 쓰고 있다. 이때 쓰이는 블록들은 ALU, Barrel Shifter, Accumulator, CTL Logic, 임시 저장 레지스터 등이다. 이들 블록은 따로 테스트 하는 것이 아니고 RAM을 테스트 하면서 동시에 테스트

되어지는 것이다.

두번째 테스트는 "Checkerboard Test"라 하며 셀과 셀의 누순 관계를 검사하는 테스트이다. 이 테스트 모드에서 테스트하는 방법은 여러 가지 "Checker Board Pattern"을 각 셀에 저장시킨후에 누순을 위한 약간의 시간이 지난 후에 읽어 내어 오류의 발생 여부를 검사한다.

3) PLA와 Decoder의 테스트

ACU 블록에는 표 1과 같이 2개의 PLA와 2개의 Decoder와 1개의 Encoder가 있다. 이들 5개의 블록들은 모두 마이크로 명령어를 해석하고 Barrel Shifter를 제어하는데 쓰인다.

표 1에서 열거한 블록들은 모두 테스트 결과를 Scan path를 통해 관측할 수 없기 때문에 16 비트의 테스트 레지스터를 부가적으로 추가해서 설계하여 테스트가 가능하게 하였다. 각 블록의 상태를 테스트 레지스터에 임시 저장시킨 다음 Scan path인 Accumulator와 Flag 레지스터로 이동시킨후 외부 핀으로 출력 시킨다. 각 블록을 테스트하는데는 3-4개의 마이크로 명령어의 집합으로 한번의 테스트를 할 수 있다. 따라서 작은 규모의 블록들임에도 불구하고 많은 테스트 시간이 걸린다.

표 1. ACU의 Decoder와 PLA  
Table 1. Decoder and PLA of ACU.

블록명	입력수	출력수	Minterm 수
Source Decoder	7	9	9
Destination Decoder	4	5	5
Prty/Exp Encoder	15	9	15
Shift PLA	6	5	22
Mult PLA	5	3	9

III. 새로운 테스터블 디자인 기법

본 논문에서 제안하는 테스터블 디자인 기법의 실제 칩에 구현하는 과정에서의 효용성 증대를 위한 방법들에 대해 자세히 기술한다.

1. Data ROM의 새로운 BIST 기법

모델로 사용된 프로세서에는 세계의 ROM이 존재한다. 이중 Macrocode ROM과 Microcode ROM은 PCU 블록에 속하며 이들은 기억 소자인 레지스터에 의해 데이터 입력을 공급받고 출력되는 데이터 값도 레지스터에 저장되기 때문에 scan path에 의해 제어 및 관측이 가능하다. 그러나 ACU 블록에 속해 있는

Data ROM은 전체적인 scan path에 의해 직접적으로 제어 및 관측할 수 없고 마이크로 명령어에 의해 제어되기 때문에 마이크로 명령어에 의해 테스트 데이터를 인가해 주고 출력도 외부로 출력시킬 수 있다. 때문에 한 주소에 저장되어 있는 데이터를 읽기 위해서는 마이크로 명령어와 주소의 scan-in과 실행 등이 반복되어야 한다.

PCU에 있는 ROM들은 scan path에 의해 주소를 입력시키고 부호화되었는 값을 출력시키는데 scan 레지스터들을 통해 시프트되어지기 때문에 테스트 시간이 많이 소비된다는 것을 감안하여 BIST 기법을 사용하고 있다. 프로그램 카운터를 변형하여 테스트 벡터를 자동 생성하고 명령어 레지스터를 변형하여 출력 데이터를 압축시켜 한 주소의 부호화된 값을 테스트하는데 1 클럭사이클만 소비되도록 설계하였다. 그래서 ROM의 테스트에 이용되는 시간을 1 / (scan path의 길이)로 줄일 수 있었다. 그러나 Data ROM의 경우는 프로그램 카운터나 명령어 레지스터와 연결되어 있지 않기 때문에 PCU의 ROM들과 같은 방법을 쓸 수가 없다. 또한 BIST 기법을 적용하려고 하여도 부가의 하드웨어인 LFSR과 MISR을 추가시키면 너무 많은 면적의 증가가 있기 때문에 불가능하다.

Data ROM의 테스트를 위한 하드웨어의 증가도 줄이고 테스트 시간도 줄일 수 있는 방법으로 기존에 존재하는 하드웨어를 약간 변형하여 하드웨어의 커다란 증가 없이 테스트 시간을 대폭 줄이는 방법을 제시한다.

Data ROM의 BIST 기법에 의한 테스트를 하기 위해서는 LFSR과 MISR이 필요하다. 이들을 새로이 추가시키면 면적의 증가가 많기 때문에 이미 존재하는 마크로 프로그램 카운터와 accumulator를 약간 변형하여 Data ROM의 테스트에 이용하면 면적의 증가 없이 테스트 시간을 줄일 수 있다. 즉 마크로 프로그램 카운터는 주소의 랜덤 자동 생성을 위해 LFSR로 쓰이고 accumulator는 출력 데이터의 압축을 위해 MISR로 사용하는 것이다. 그러면 Data ROM과 마크로 프로그램 카운터, accumulator를 연결하는 연결선과 LFSR, MISR로 변형하는데 필요한 약간의 하드웨어의 증가만으로 Data ROM의 테스트에 걸리는 시간을 대폭 줄일 수 있다.

마크로 프로그램 카운터를 Data ROM의 테스트에 이용하는 LFSR로 만드는 데는 아무런 하드웨어의 변형 없이 단지 연결선만이 필요로 하다. 마크로 프로그램 카운터는 PCU의 Microcode ROM을 테스트하는데 이미 LFSR로 사용하기 위해 설계되어있기 때문에 그대로 사용하면 되는 것이다. 더군다나 Microcode

ROM의 테스트를 위한 주소가 7비트로 Data ROM의 주소와 같은 수이기 때문에 회로의 변경이 전혀 필요 없고 단지 연결선이 증가되기 때문에 면적의 증가는 전혀 없다.

Accumulator를 Data ROM의 테스트를 위한 MISR로 변형시키는데는 아주 약간의 하드웨어만이 필요하다. Data ROM과 accumulator와의 연결은 이미 데이터 버스로 연결되어 있기 때문에 더이상의 재원의 증가가 필요 없고 MISR로 변형시키는데도 이미 accumulator가 scan 레지스터로 쓰이고 있기 때문에 시프트 동작을 할 수 있다. 따라서 특성 방정식 ( $X^5 + X^3 + X^2 + 1$ ) ()을 만들어 주는 XOR 게이트 4개와 MUX 4개만이 새로운 MISR을 만드는데 필요한 하드웨어의 전부이다. 그림 6에 Data ROM의 새로운 BIST 구현을 위한 회로도도를 나타내었다.

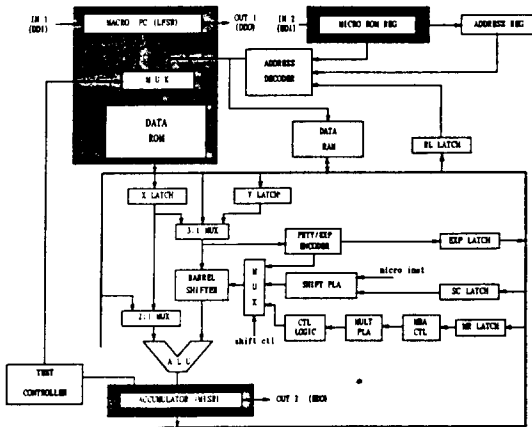


그림 6. Data ROM의 BIST 회로도  
Fig. 6. BIST structure of data ROM.

새로운 설계에 의한 테스트 방법은 우선 ACU의 scan path를 통해 마이크로 명령어 레지스터에 'ACC <- data ROM'라는 명령어를 입력시키고 latch시킨다. 명령어가 입력되어 latch되면 매크로 프로그램 카운터는 LFSR로 accumulator는 MISR로 동작시킨다. 이때 LFSR과 MISR의 최상위 비트는 핀을 통해 외부에서 관측한다.

새로운 설계에 의한 이득 및 손실을 살펴보면 칩 면적에 있어서 4개의 XOR 게이트와 4개의 MUX를 만드는데 필요한 40개의 트랜지스터에 해당하는 면적의 증가가 있었다. Scan path를 사용하여 마이크로 명령어에 의해 Data ROM을 제어하면서 테스트하면 테스트 벡터를 시프트 시키는데 scan path의 길이만큼 시간이 걸린다. Data ROM의 일정한 주소에 해당하는 저

장 데이터를 확인하는데 scan path의 길이인 36 클럭 사이클이 소비된다. 그러나 새로이 BIST 기법을 사용하면 일정한 주소에 해당하는 저장 데이터를 확인하는데 1 클럭 사이클만이 소비 되기 때문에 테스트 시간의 이득은 1 / (scan path의 길이 = 36)이 된다.

2. PCU의 PLA 테스트를 위한 새로운 Scan path

그림 3에 PCU의 테스트를 위한 scan path의 구조가 있다. Scan path의 구조를 보면 한블록의 테스트를 위해 그 블록의 입력이 되는 모든 레지스터를 scan path로 연결하여 테스트 벡터를 인가하고 있다. 분할 scan 기법을 사용함에도 불구하고 테스트 벡터의 비트 수가 테스트되어지는 블록의 입력수 보다 많아 시프트하는데 불필요한 시간의 낭비가 있는 블록이 있다. 이러한 테스트 시간의 낭비가 있는 블록의 scan path를 재조정함으로써 시프트 시간을 줄일 수 있다. 이에 해당되는 블록은 Macro jump/interrupt control logic과 Macro instruction decoder로 이들을 위한 새로운 scan path는 그림 7에 나타내었다.

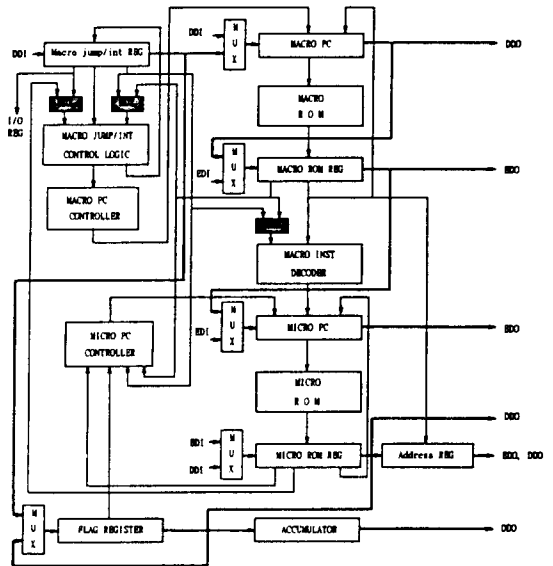


그림 7. PCU PLA의 테스트를 위한 Scan path  
Fig. 7. Scan path for testing PCU PLA.

그림 7에서 보듯이 Macro jump/interrupt control logic의 테스트를 위한 scan path는 DDI 핀 -> Macro jump/interrupt register -> Macro PC -> DDO 핀으로 연결되어 총 scan path의 길이가 종전의 27 비트에서 18비트로 1/3만큼 줄어들었다. 또

한 Macro instruction decoder의 경우는 종전의 DDI 핀 → Macro PC → Macro ROM register → Micro PC → EDO 핀으로 연결되는 긴(26 비트) scan path를 가지고 있었는데 반해 재 설계된 scan path는 EDI 핀 → Macro ROM register를 scan-input 경로로 사용하고 Micro PC를 scan-output 경로로 사용하여 scan path의 길이를 10 비트로 짧게하여 약 2/3만큼의 테스트 시간의 이득을 얻을 수 있다.

Micro PC controller의 경우는 이 블록의 입력이 신호가 되는 간단한 레지스터의 출력이 아니고 여러 기능 블록들이 차례차례 연결되어 만들어지는 복잡한 입력 데이터를 갖고 있기 때문에 종전의 scan path를 그대로 사용하여 오류 모델의 검사가 아닌 부분적인 기능 테스트를 한다.

Address register의 scan path는 종전에 사용된 경로가 최단거리를 갖는 scan path이므로 재설계 없이 그대로 사용한다.

PCU에 있는 PLA들의 테스트를 위해 약간 변경된 scan path를 사용하였는데, 이를 통해 PCU의 모든 PLA를 테스트하는데 약 20%의 테스트 시간 이득을 얻을 수 있었다. 반면에 이러한 scan path의 재설계를 위해 필요한 하드웨어로 MUX 5개에 해당되는 칩면적의 증가가 있었다.

3. ACU의 레지스터 및 PLA의 새로운 테스트 기법

ACU의 Register와 PLA들은 전체적인 scan path와 연결되어 있지 않기 때문에 scan path를 통한 직접적인 제어 및 관측이 불가능하다. 모든 블록들이 마이크로 명령어 레지스터의 제어를 받기 때문에 제어는 명령어를 이용하면 되지만 그들의 상태를 관측하는 것은 불가능하다. 따라서 기존의 testable design은 임시적으로 각 블록의 상태를 저장하는 테스트 레지스터를 설계하여 연속적인 명령어의 실행으로 테스트 입력도 가하고 출력도 scan path를 통해 관측할 수 있게 설계하였다. 이것은 테스트 설계의 중요한 요인 중의 하나인 관측성을 높인 것이다. 하지만 제어성을 높이지 않은 결과 각 블록의 테스트를 위해서는 3-4개의 연속적인 명령어 집합이 필요하다. 따라서 모든 블록의 크기가 작음에도 불구하고 테스트 시간이 매우 많이 걸린다.

ACU의 테스트 시간을 줄여 보고자 BIST 기법을 이용하여 각 블록에 대한 제어성 및 관측성을 향상시키고 테스트 방법을 다르게 하였다. 종전에 BIST 기법에 의해 설계되었던 테스트 레지스터를 변경하였고 이를 독립적인 scan path로 사용하였다. 각 블록에는

MUX를 사용하여 테스트 레지스터를 이용하여 직접적으로 테스트 벡터를 인가할 수 있게 설계하였다. 그림 8에 새롭게 테스트 레지스터를 이용한 ACU 블록의 테스트를 위한 구조를 구현하였다.

그림 8에서 보듯이 각 블록의 테스트 벡터가 테스트 레지스터를 통해 직접 인가되고 테스트 응답이 테스트 레지스터를 통해 직접 외부로 출력되기 때문에 테스트 벡터와 응답의 입출력에 필요한 여러 개의 명령어 집합을 테스트 벡터로 이용하는 종전의 테스트 방법보다 테스트 시간의 이득을 얻을 수 있다. 또한 한번 scan-in되는 테스트 벡터의 길이도 종전보다 약 반정도로 줄어들었기 때문에 약 100%의 시간 이득을 얻었다.

새로운 테스트 구조에 의한 ACU의 PLA와 레지스터의 테스트는 Source/Destination Decoder의 테스트, PRY/EXP Encoder 및 Exp latch의 테스트, Shift PLA 및 SC latch의 테스트, MULT PLA 및 MR latch의 테스트의 4가지 모드로 나누어 테스트 한다.

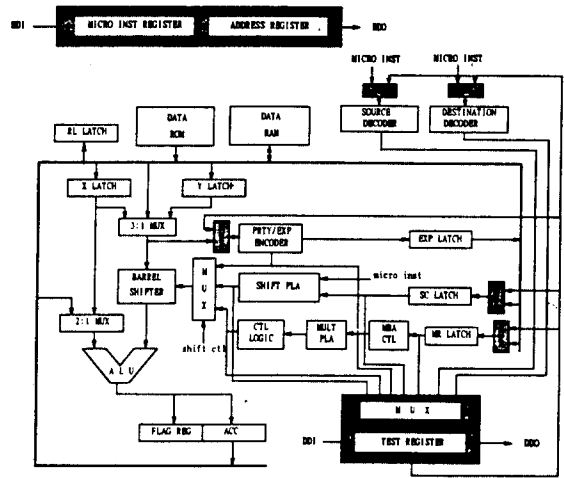


그림 8. ACU의 Testable design  
Fig. 8. Testable design for ACU.

각각의 테스트 모드시 각 블록은 테스트 제어장치의 제어를 받아 정상 동작시와는 전혀 다른 블록에서 입력 데이터를 받게 된다. 테스트 레지스터는 테스트되어 질 블록의 입력 테스트 벡터와 응답 데이터를 모두 저장하고 시프트 시키는데 사용된다. 테스트 순서를 보면 scan 모드에서 각 블록의 테스트 벡터가 테스트 레지스터로 입력된다. 입력이 완료되면 실행 모드로 바꾸어 실행을 시키고 그 응답은 다시 테스트 레지스터에 저

장된다. 실행이 끝나면 다시 scan 모드로 전환하여 외부로 출력하고 동시에 다음 테스트 벡터가 입력된다. 이를 각 블록에 대해 오류의 발생 여부를 보장할 수 있을 때까지 반복한다.

새로운 설계에 의한 면적의 증가는 테스트 레지스터를 다입력 시프트 레지스터로 바꾸어 주는데 필요한 5개의 입력을 갖는 16개의 MUX가 필요하고, 각 블록의 입력단에 정상동작시와 테스트 모드시 선택하여 입력을 받아들이는데 필요한 2개의 입력을 갖는 MUX 48개가 필요하다. 테스트 레지스터 scan-input의 제어에 필요한 회로등 총 180여개의 트랜지스터의 증가가 요구된다. 이는 종전의 테스트 레지스터를 만들어 주는데 필요한 270여개의 트랜지스터에 비하여 약 67%의 면적의 증가를 가져왔다.

시간적인 이득을 보면 Source/Destination Decoder에 필요한 시간은 한개의 테스트 벡터와 응답을 시프트 시키는데 16 클럭 사이클이 필요하다. 종전에는 두 Decoder를 따로따로 마이크로 명령어의 제어에 의해 테스트 하였기 때문에 한번의 테스트 벡터를 가하기 위해 2 X 36개의 클럭 사이클이 필요하였다. 그러므로 시간적인 이득은 약 4.5배정도 빨라진 것이다. PRTY/EXP Encoder를 종전의 방법으로 테스트 할 경우 1개의 테스트 벡터는 4개의 명령어 집합으로 이루어져 있다. 그러므로 1개의 테스트 벡터를 인가하고 실행하는데 4 X (36 + 1)개의 클럭 사이클이 사용되었다. 그런데 새로운 설계에 의하면 단지 16 + 1개의 클럭 사이클만이 필요하므로 PRTY/EXP Encoder의 경우는 9배 정도 테스트 시간이 짧아졌다. 마찬가지로 Shift PLA와 MULT PLA의 경우에도 종전의 테스트 방법에 의하면 같은 scan path에 의해 각각 3개와 4개의 명령어 집합이 한개의 테스트 벡터로 사용되었기 때문에 새로운 설계에 의한 테스트 시간은 각각 7배와 9배정도씩 빨라졌다. 이것은 ACU 블록에서 Data ROM과 Data RAM을 뺀 나머지 블록의 테스트는 약 7배정도 빠른시간안에 테스트할 수 있게 된 것이다. 새로운 테스트 구조에 의한 테스트를 할 경우 종전의 방법보다 테스트에만 쓰이는 칩면적의 증가는 약 67%인 것에 비하여 테스트 시간은 약 700% 향상되었다. 칩면적의 경우는 전체 칩면적에 비하여 매우 적은 증가이고 테스트 시간의 경우 전체 칩 테스트에 사용되는 시간에 많은 감소를 가져오므로 양호한 결과를 얻을 수 있었다.


#### IV. 성능 비교 분석

이전의 테스트를 위한 설계와 새롭게 설계한 것과의

성능 비교는 칩면적의 증가와 테스트 시간의 감소에 의해 이루어질 수 있다. 이들 요인은 서로 상반 관계를 갖는 것으로 테스트 시간의 감소를 얻으려면 부가적인 하드웨어의 증가로 인한 칩면적의 증가가 필연적이다. 따라서 최대한의 성능 향상을 얻으려면 아주 약간의 면적의 증가로 많은 테스트 시간의 감소를 얻는 것이다.

테스트 시간의 감소를 살펴보면 기존의 방법에 의한 칩 전체를 테스트하는데 10 MHz의 속도로 15 msec정도 걸리던 것이 약 40%의 시간의 감소를 얻어 9 msec 정도의 시간만으로 테스트가 가능하다. Data RAM을 테스트하는데 소비되는 시간이 전체 칩을 테스트하는데 걸리는 시간의 약 50%를 차지한다는 것을 감안할 때 각 부분 블록에 대해서는 매우 많은 시간의 감소가 이루어졌음을 알수있다. 특히 ACU의 블록의 경우 마이크로 명령어를 이용하여 테스트하던 것을 Data ROM에는 BIST에 의해 자동 테스트 되게 하였고, 레지스터와 PLA들은 새로운 scan path에 의해 테스트되어지게 한 결과 Data RAM의 테스트를 제외하면 기존의 방법을 사용하여 테스트 할 때의 시간의 약 20%정도의 시간만으로도 테스트가 가능하게 되었다.

표 2. 테스트 시간의 비교  
Table 2. Compare of test time.

\*  : 새롭게 설계된 블록

테스트 명칭	기존의 방법 (A)	새로운 방법 (B)	감소 비율 (A-B)/A
Macrocode ROM LFSR/MISR	146	146	0
Macrocode ROM INC/MISR	146	146	0
Dummy logic	224	224	0
Macro Interrupt/Jump control logic	3640	2470	0.32
Microcode ROM LFSR/MISR	1078	1078	0
Microcode ROM INC/MISR	1078	1078	0
Macro instruction decoder	1539	627	0.59
Micro PC controller	1368	1368	0
Address register	1008	1008	0
Source/Destination decoder	3240	765	0.76
Data ROMs	27936	146	0.99
Shift,Prty/Exp,Mult PLA	26352	5040	0.81
Data RAM - Functional	13824	13824	0
Data RAM - Checkerboard	55296	55296	0
I/O registers	7978	7978	0
총 합 계	144853	91194	0.37

표 2에 기존의 방법과 새로운 방법에 의한 테스트



시간을 비교 하여 새롭게 설계된 블록들의 테스트 시간이 감소됨을 나타내었다.

면적의 증가 측면에서 비교하면 기존에 사용하던 방법에 의해 설계된 테스트를 위한 하드웨어가 차지하는 비율이 전체 칩면적의 약 6%를 차지했었다. 약간의 하드웨어의 증가에 의한 새로운 방법에 의하면 테스트를 위한 하드웨어가 차지하는 비율은 약 6.5%로 전체 칩면적의 0.5%정도의 면적의 증가가 있었으므로 거의 무시할수있는 정도이다.

## V. 결 론

본 논문에서는 디지털 신호처리 프로세서의 Testable Design 기법에 대하여 자세히 알아보고 시간의 이득을 얻을 수 있도록 부분 블록들에 대한 재설계를 하였고 이들의 성능에 대하여 비교하였다.

모델로 사용된 프로세서에 쓰인 테스트를 위한 설계 기법들은 모두 최근에 발표된 기법들로 부분 분할 scan path 기법, Multiple scan path 기법, LFSR을 이용한 테스트 벡터의 자동 생성, MISR을 이용한 응답 데이터의 압축 기법, 진단 테스트 기법 등의 다양한 종류의 설계 기법들이 사용되었다. 이 칩은 일반적인 프로세서가 갖는 거의 모든 기능 블록들을 갖고 있기 때문에 이 칩에 쓰인 테스트를 위한 설계 기법들을 다른 프로세서에도 똑같이 적용할 수 있다.

테스트를 위한 이론적 기법들의 변환 없이 부분 블록들에 대해 적용하는 방법만을 약간 다르게 재설계 하였다. 같은 이론을 가지고 구현의 방법에 따라 많은 비용의 이득을 얻을 수 있다는 것을 알 수 있었다. 본 논문에서 테스트에 필요한 설계 방식의 적용에 있어서 사용한 방법들은 이미 존재하는 하드웨어의 공동 사용으로 매우 적은 칩면적의 증가로 매우 많은 시간의 이득을 얻는 기법(Data ROM의 BIST), 부분 분할 scan path 기법을 적용할때 테스트 벡터의 길이를 최대한 줄이기 위해 테스트되어지는 블록의 입력단에 MUX를 이용하여 scan path의 경로를 최대한 줄이는 기법(PCU 블록의 PLA의 테스트), 이미 존재하는 기억장치에 의해 제어 및 관측이 불가능한 블록에 대해 약간의 하드웨어를 이용하여 scan path를 형성하여 가제어성과 가관측성을 높이고 이를 직접 외부와 연결하여 직접적인 제어에 의해 테스트 벡터의 길이를 아주 짧게 하는 기법(ACU 블록의 PLA와 레지스터의 테스트) 이다. 이러한 방법들은 그 동안 발표되어 온 scan path 및 BIST 기법을 실제 칩에 구현할때 최대한의 이득을 얻기 위해 칩의 특성을 고려하여 설계하는 방법들이다. 어느 시스템이나 위의 개념을 적용하면

매우 적은 하드웨어의 증가에도 불구하고 많은 시간의 이득을 얻을 수 있다.

테스트에 소비되는 시간은 곧 제조 공장에서 생산되는 칩의 양과 직접적으로 연관되기 때문에 앞으로 VLSI 칩의 제조 기술이 발달될수록 테스트 시간은 매우 중요한 요인이 될 것이다. 따라서 본 논문에서는 약간의 하드웨어가 증가함에도 테스트 시간을 줄이는데 역점을 두었다. 그 결과 하드웨어의 증가는 거의 무시할 수 있을 정도로 즉 전체 칩 면적의 0.5%정도의 증가만이 있었고 그에 비하여 테스트 시간은 약 40%의 감소를 얻을 수 있었다.

앞으로 더 연구해야 할 방향은 특정한 모델에 관한 기법이 아닌 일반적인 회로에 대하여 쓰일 수 있는 방법들의 정립을 이루는 것이다.

## 참 고 문 헌

- [1] Luls A. Bonet, "Testability Features of a 32 kbps ADPCM Transcoder", Proc. ITC, pp.161-171, 1988.
- [2] CITT Recommendation G.721 Melbourne, 1988, "32 kbits/s Adaptive Differential Pulse Code Modulation (ADPCM)", CCITT study group XVIII.
- [3] W. C. Bruce, C. C. Hunter and L. A. Basto, "Testing Barrel Shifters in Microprocessors", Proc. ITC, pp.145-153, 1986.
- [4] Zuxi Sun, L. T. Wang, "Self-Testing of Embedded RAMs", Proc. ITC, pp.148-156, 1984.
- [5] L. T. Wang, E. J. McCluskey, "Built-in Self-Test for Sequential Machines", Proc. ITC, pp. 334-341, 1987.
- [6] XI-AN ZHU, MELVIN A. BREUER, "Analysis of Testable PLA Designs", IEEE Design & Test of Computers, pp.14-28, 1988.
- [7] S. Narayanan, C. A. Njinda, M. A. Breuer, "Optimal Sequencing of Scan Registers", Proc. ITC, pp.293-302, 1992.
- [8] R. Gupta and M. A. Breuer, "Ordering storage elements in a single scan chain", Proc. Int'l Conf. on CAD, pp.408-411, November, 1991.

- [9] M. S. Abadir and M. A. Breuer, "Scan path with Look Ahead SHifting (SPLASH)", Proc. ITC, pp. 696-704, 1986.
- [10] R. Gupta and M. A. Breuer, "BALLAST: A Methodology for Scan Design", Proc. 19th Intn'l. Symp. on Fault-Tolerant Computing, pp. 118-125, June, 1989.
- [11] H. Fujiwara and A. Yamamoto, "Parity-Scan Design to Reduce the Cost of Test Application", Proc. ITC, pp. 283-292, 1992.
- [12] M. Abramovici, M. A. Breuer, and A.S. D. Friedman, "Digital Systems Testing and Testable Design", W. H. Freeman & Co., New York, 1990.

---

 저 자 소 개
 

---



金東奭(正會員)

1967年 3月 30日生. 1992年 2月 충남대학교 공과대학 전자공학과 공학사. 1994年 2月 충남대학교 전자공학과 공학석사. 1994年 3月 ~ 現在 국방과학연구소 연구원. 주관심분야 :

VLSI 설계 및 프로세서의 테스터블 디자인

李起煥(正會員) 제 29권 A편 제 3호 참조

現在 충남대학교 전자공학과 부교수

金備鎰(正會員) 제 29권 A편 제 3호 참조

現在 충남대학교 전자공학과 부교수



崔海旭(正會員)

1950年 9月 25日生. 1973年 2月 서울대학교 공과대학 전자공학과 공학사. 1980年 9月 불란서 그레노블공과대학 전자공학과 (전자공학전공) 공학석사. 1984年 10月 불란서 그레노블공과대학 컴퓨터

공학과 (전자공학전공) 공학박사. 1982年 2月 ~ 1984年 11月 불란서 톰슨반도체(주)집적회로 설계연구원. 1990年 9月 ~ 1994年 8月 충남대학교 공과대학 전자공학과 부교수(겸임). 1984年 12月 ~ 現在 한국전자통신연구소 반도체연구단 책임연구원. 관심 분야 : DSP설계, 무선 LAN 프로토콜 프로세서 및 모델설계, 프로세서의 테스터블 디자인