

# JTAG Port를 이용한 On-Board Flash Fusing①

## JTAG의 이해와 구성

근래의 Embedded System의 개발에 있어서 중요한 부분을 차지하고 있는 것 중의 하나가 바로 JTAG이라 해도 이젠 과언이 아니게 되었다. 그만큼 사용 빈도 또한 높아졌음을 의미한다. 본 연재에서는 여러 활용분야 중에 JTAG Port를 직접 액세스하여 보드 상에 실장되어 있는 Flash에 데이터를 기록하는 방법을 설명하고자 한다. 물론, 훌륭한 ICE 장비들을 이용하면 쉽게 해결될 일일 지도 모른다. 하지만, 이 글을 쓰는 목적은 JTAG의 이해를 돕는데 있다. 그 첫 번째로서 JTAG의 등장 배경 및 구성에 대해 먼저 설명하고자 한다.

글: 김재성/(주)메리테크 개발팀장  
jskim@meritech.co.kr

### [ 연재순서 ]

- ➔ 1 JTAG 이해와 구성
- 2 JTAG 동작원리
- 3 Flash Memory 구조 및 동작원리
- 4 BSDL File 구조
- 5 JTAG Port 제어 S/W 설계

### JTAG 배경

처음 JTAG이 등장하게 된 배경은 board-level 테스트(보드에 실장되어 있는 칩과 칩 사이의 interconnection을 테스트 하는 것)를 하기 위한 방법의 구현에서 부터 시작했다고 볼 수 있다.

1980년대 중반 이전에는 “bed of nails”이라 불리는 탐침을 이용하여 직접 PCB 의 Test Point를 contact하는 방식으로 보드를 검증 및 테스트 했었다. 이후, 패키지의 소형화로 인하여 핀 간격이 좁아지고, PCB layer가 다층화되면서 이러한 방법의 테스트가 어려워졌다. 1980년대 중반, JTAG(Joint Test Action Group)이라는 세계 각국의 200개 이상의 주요 전자 및 반도체 제조회사들로 구성된 단체가 결성되었다. JTAG은 보드 테스트의 문제 해결책으로서 ‘A Serial Shift Register Around the Boundary of the Device’ 라는 개념의 boundary scan을 고안했는데, 당시에는 칩 내부에 shift/update register를 할당하여 board-level 테스트를 쉽게 하는 것이 주된 목적이었다. 이들에 의해 확립된 Test Access Port 와 boundary scan은 1989년에 IEEE Committee에 선정되었으며, 1990년 IEEE는 JTAG에서 만든 1149.1이라는 표준안을 채택함으로써 그 성과를 거두게 된다. 1993년에 1차 correction and improvement된 버전이 발표되었고, 이와 동시에 BSDL(Boundary Scan Description Language)이 발표되어 현재에 이르고 있다. 흔히 JTAG과 boundary scan을 혼용해서 사용하는 경우가 많은데, 좀 더 정확하게 표현하자면 JTAG 보다는 boundary scan 또는 IEEE 1149.1로 부르

는 것이 맞는 표현이라 하겠다.

boundary scan이 표준안으로 자리잡은 초기에는 세계적인 ASIC 벤더들이 앞다퉀 이를 수용하였으며 당시 최대의 회사였던 텍사스 인스트루먼트(TI)와 IBM 등은 이 새로운 기술을 수용하는 데 적극적이었고, 또한 큰 성공을 거두었다. 초기에는 board-level 테스트를 목적으로 탄생하였고 본래의 목적에 충실한 활약을 하였지만, boundary scan은 여러 가지 다른 애플리케이션에도 광범위하게 사용되고 있다. 현재 가장 쉽게 접할 수 있는 것이 ARM이나 DSP Core 등과 같은 범용 RISC 코어들의 디버깅에 사용되는 것이다. 이외에도 보드 상에 실장되어 있는 여러 디바이스들을 제어하는 것, 예를 들면 I/O Access, Memory Operation 등을 할 수가 있다.

### JTAG의 Testability Benefits

전통적인 board-level 또는 device-level 테스트는 많은 시간이 소요되며, 각 보드 또는 디바이스의 종류마다의 특정한 하드웨어와 복잡한 ATE(Automated Test Equipment)를 필요로 한다. 그 결과 비용 및 개발시간이 증가하게 된다. 이것은 종래의 경우보다 타임 투 마켓의 개념이 더욱 더 중요시 되는 고도의 기술 시장에서 짧은 개발기간에 우수한 품질의 제품을 제공해야 하는 회사들에겐 문제가 아닐 수 없다. 이러한 문제를 해결하기 위한 혁신적인 방법 중의 하나는 “수행되면서 테스트 할 수 있는” design-for-test 기술을 통합시키는 것이다. 이것은 테스트 프로그램을 개발하는 시간을 줄일 수 있고 낮은 비용의 ATE 솔루션을 구성할 수 있게 한다. JTAG, 즉 boundary scan이 바로 이러한 기술의 구현이라 하겠다.

좀더 실질적인 예를 들면 하드웨어 개발적인 측면에서 볼 때 테스트를 하다보면 문제의 원인을 알아내기 위해서 특정 단자의 상태를 임의로 설정해야 될 필요가 있게 된다. 이미 디바이스는 PCB 상에 트레이스(trace)를 통하여 연결이 되어 있어 강제로 인가하려면 핀의 연결을 끊어야 할지도 모른다. 그런데 여러 조건을 테스트 하려면 디바이스에 연결되어진 수많은 배선들을 끊고 있고 하는 것이 쉽지 않은 작업일 것이고, 동작중인 상태에서 디바이스의 테스트를 해야 한다면 이런 방법도 불가능할 수 있다. 또한 동작중에 신호의 변화를 알아내기 위해 별도의 회로를 추가해야 될지도 모른다. 이런 회로를 일일이 구현하는 것도 문제가 되며, 그 추가된 회로가 문제를 일으킬 소지도 있다. 그래서 절충안으로 제시된 것이 CPLD와 같은 대규모

LSI에 이런 기능을 하는 로직을 함께 집적시키는 방법이다. 그러면 물론 CPLD에 넣을 수 있는 용량의 일부분을 낭비하는 결과를 초래할 수 있지만, 그에 비해 하드웨어 검사 및 테스트를 하기 위해 소모되는 시간 등을 고려한다면 전체적인 하드웨어 개발에 있어서는 결과적으로 이익이 될 수 있을 것이다.

## JTAG 구성

JTAG을 구성하는 요소는 크게 4가지로 요약할 수 있다. JTAG이 어떤 디바이스에 내장되었을 때 외부(해당 디바이스를 테스트하기 위해 접근하는 호스트 장치)와의 인터페이스를 제공하는 TAP(Test Access Port), boundary scan을 구성하며 디바이스 각각의 주요 I/O핀마다 대응되는 BSC(Boundary Scan Cell), 4개의 각기 다른 목적으로 사용되는 내부 Register, 그리고 전체의 제어 역할을 담당하는 TAPC(Test Access Port Controller)로 구성된다. 그림 1은 디바이스 내에 JTAG이 내장 설계된 칩을 보여준다. 통상 JTAG을 내장한 디바이스를 IEEE 1149.1 Compliant Device라고 한다.

JTAG Compliant Device는 그림 1에서 보는 바와 같이 칩 내부에 본래의 기능에 해당하는 ASIC 블록과 테스트 및 기타 목적을 위한 JTAG 관련 블록이 있다. JTAG 블록은 4개의 Register와 이들을 적절하게 TDI 또는 TDO에 연결시켜 데이터를 주고받을 수 있는 구조로 TAP Controller가 그 중추적인 역할을 담당하고 있다. 입출력의 상태를 읽어들이거나 인가할 때는 BSR(Boundary Scan Register)를 통한 serial shift 방법으로 BSC에 접근 및 그에 대응되는 입출력 핀을 최종적으로 액세스하게 된다.

### TAP(Test Access Port)

JTAG은 IEEE에서 승인된 규격이기 때문에 이를 내장한 디바이스라면 모두 다 TAP에 의한 인터페이스를 제공한다. TAP은 단지 5개의 신호로 구성되는 상당히 간단한 구조를 갖는다.

TAP은 필수적으로 사용되는 TDI, TDO, TMS, TCK 신호와 선택적으로 사용되는 TRST 신호로 이루어져 있으며 그 기능은 표 1과 같다.

5개의 신호 중 TDI, TDO는 Multiplexing 되어 각각의 Register와 직접 연결될 수 있으며, 어느 Register에 연결되느냐에 따라 이송되는 데이터는 I/O핀, 명령, ID, 내부 Register(User

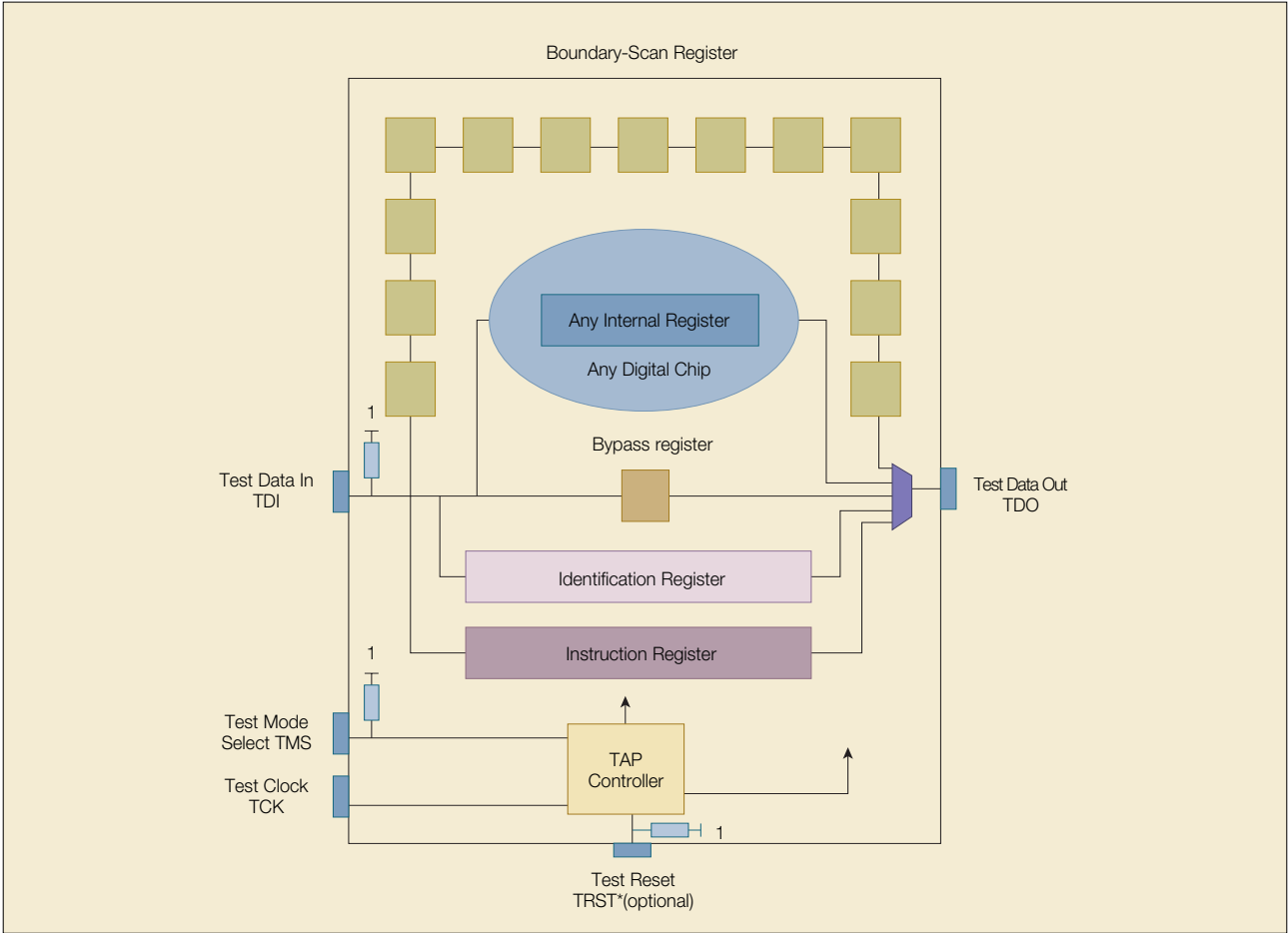


그림 1. IEEE 1149.1 Device Architecture

Signal Name	Type	Description
TDI(Test Data In)	Input	직렬 테스트 데이터 입력 신호(Default - High 상태) TCK 신호와 동기되어 TCK가 rising edge 일 때 shift 입력된다.
TDO(Test Data Out)	Output	직렬 테스트 데이터 출력 신호(Default - High-Z 상태) TCK 신호와 동기되어 TCK가 falling edge 일 때 shift 출력된다
TCK(Test Clock Input)	Input	테스트 동기 신호 각 신호의 동기를 맞추기 위한 Main Clock
TMS(Test Mode Select)	Input	테스트 모드 선택 제어 신호(Default - High 상태) JTAG이 제공하는 test mode를 선택 TCK 신호와 동기되어 TCK가 rising edge 일 때 TMS의 신호에 따라 JTAG state machine의 state가 천이되어 모드가 변경된다.
TRST(Test Reset)	Input	TAP Controller 비동기 리셋 신호(Default - High 상태) Low 상태가 입력될 때 TAP Controller를 초기화한다. Test Logic Reset State와 같은 기능

표 1.TAP 신호의기능

Register) 데이터가 된다. TMS, TCK는 TAP Controller의 state machine을 제어하여 Register를 select 및 활성화시키는데 사용되며, TRST는 이러한 TAP Controller를 리셋시켜 초기화 할 수 있다. 추가적으로 Select, Enable, Reset 등의 신호들이 있지만 필자는 일반적인 예를 기준으로 설명하는 관계로 생략한다.

**BSC(Boundary Scan Cell)**

테스트를 하고자 하는 주요 입출력 핀에 대해 테스트 패턴에 해당되는 신호를 인가 또는 현재의 상태를 검증하기 위해 내부적으로 cell이 구성되는데, 이것이 바로 boundary scan cell이다. 그림 2에서 알 수 있듯이 BSC는 입출력 핀과 내부 코어 사이에 존재하면서 상호 테스트를 가능하게 한다.

BSC는 경계 영역 즉, 디바이스의 핀에 대해서만 구성을 하는데, 그 개수는 디바이스마다 다르다. 이들 중 전원, TAP, Clock 등은 BSC를 구성하지 않는다. 따라서 BSC는 주로 I/O 및 기타 특수 목적의 핀에 구성되게 된다. 그렇다면 전원 및 기타 제외되는 핀들이 10개인 100핀의 디바이스가 있다고 가정할 때, 이 디바이스의 BSC는 90개 일까?

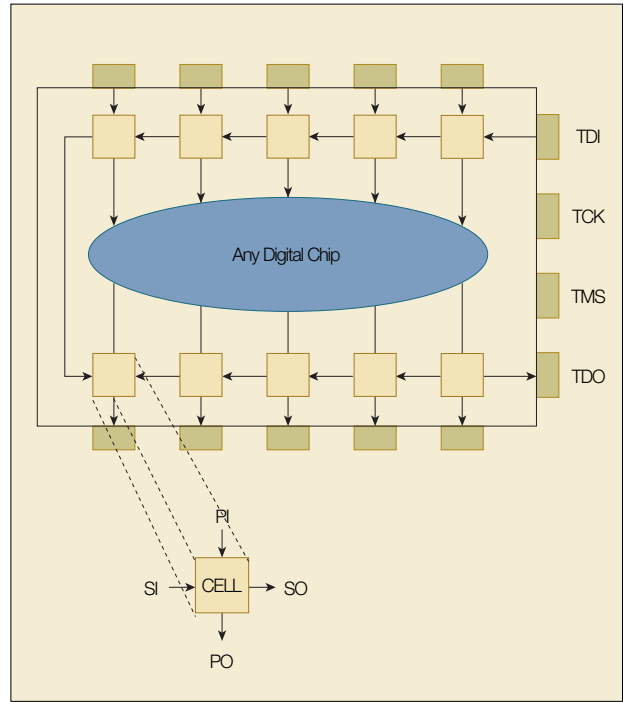


그림 2 Boundary Scan Cell

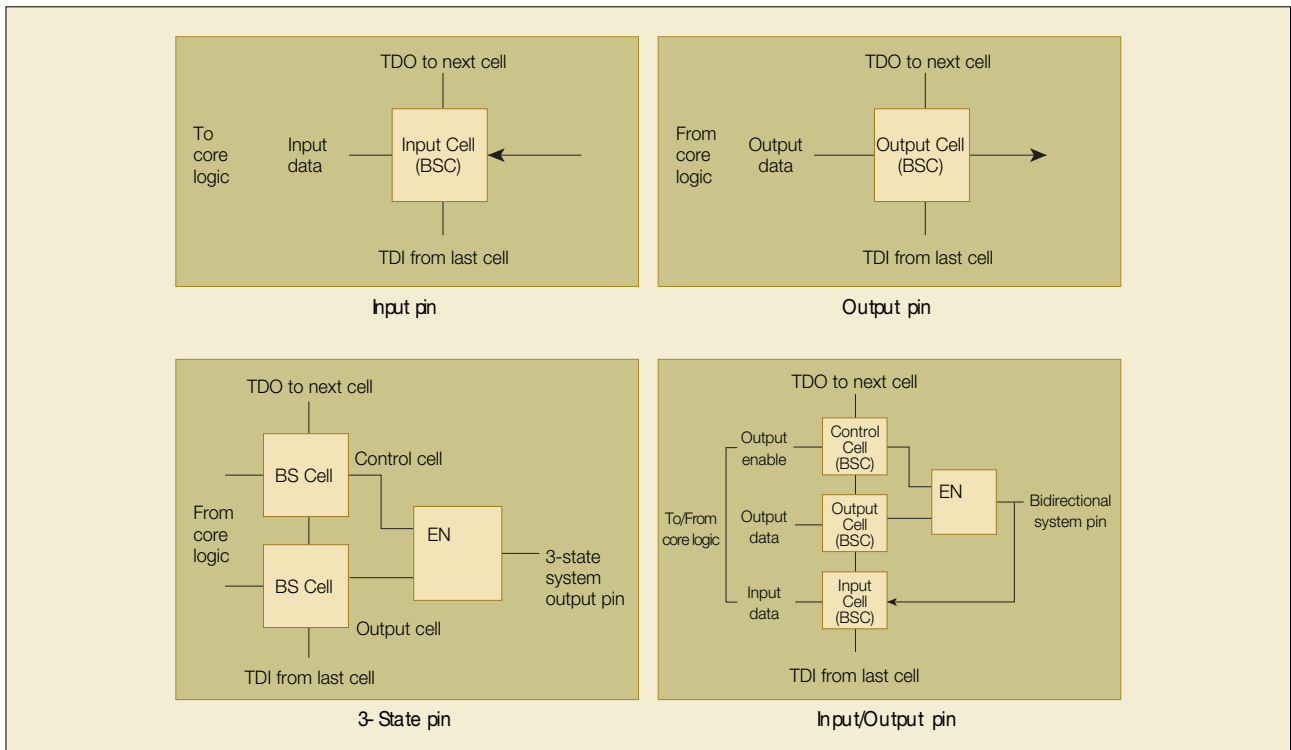


그림 3 IO Pin의 Cell 구성

Signal	Description
PI (Parallel Input)	외부 신호 입력 또는 내부 Core의 출력 (외부 Device 및 내부 Core에 연결)
PO (Parallel Output)	외부 신호 출력 또는 내부 Core의 입력 (외부 Device 및 내부 Core에 연결)
SI (Serial Input)	직렬 입력 (Boundary Scan Register에 연결)
SO (Serial Output)	직렬 출력 (Boundary Scan Register에 연결)

표 2. BSC의 기본 구조

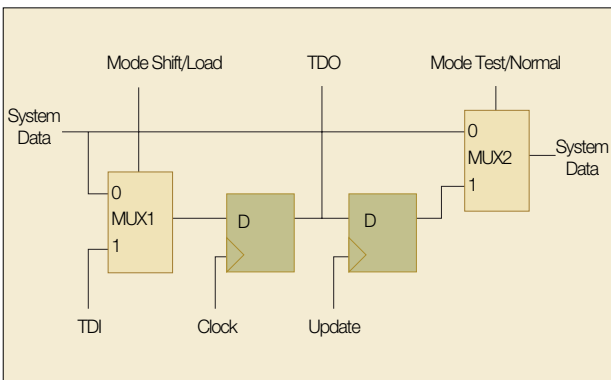


그림 4. BSC의 세부 구성

Register	Description
Instruction	디바이스에 따라 구성되는 bit width가 다르며 통상 CPLD, FPGA는 5, 6bit이며 MCU는 4, 5bit로 구성된다. TDI와 TDO를 어느 Register에 연결할 지를 결정한다.
Identification	32bit로 구성된다. 디바이스의 정보를 얻을 수 있다.
Bypass	TDI와 TDO를 바로 연결시킨다. 여러 디바이스가 Chain으로 연결되어 있을 때 유용하게 사용된다.
Boundary Scan	BSC를 통한 I/O핀 검사

표 3. 내부 Register의 종류와 기능

그렇지 않다. 그 이유는 I/O에 종류에 따라 입력, 출력, 입출력, 3-State 등의 신호형태를 갖기 때문에 BSC도 이러한 형태로 구현되어야 된다. 예를 들자면 입출력 핀의 경우 그림 3처럼 입력 cell, 출력 cell, direction cell이 필요하게 된다. 다시 말해 1핀에 3개의 cell이 존재하는 셈이 되며, 그로 인해 핀의 개수와 cell의 개수는 항상 일치하지는 않는다.

**BSC의 기본 구조 및 기능**

Cell은 PI, PO(parallel 입출력), SI, SO(serial 입출력) 및 Flip-Flop으로 구성된다.

BSC의 기능은 다음과 같다.

- ① PI에 들어오는 신호를 메모리에 기억(캡처 기능)
- ② 메모리에 있는 내용을 PO에 인가하는 기능
- ③ PI의 신호를 PO에 바로 넘기는 기능. 이때는 PI의 신호가 바로 PO에 전달된다
- ④ SI의 입력을 메모리에 기억시키고 메모리의 내용을 SO로

옮기는 기능. 이 기능을 이용하여 각 CELL의 내용을 외부로부터 읽어 올 수 있고 특정 상태를 내부에 인가할 수 있는 것이다.

위의 이런 기능은 CELL 외부에 있는 레지스터에 의해 제어를 받게 된다.

**BSC의 동작원리**

BSC를 좀더 세분화하여 도식화 하면 그림 4와 같은 구조를 갖는다.

구성을 보면 PI, PO 데이터의 경로를 설정할 수 있는 2개의 Multiplexer와 SI, SO의 데이터를 Latch시켜 기억할 수 있는 2개의 D Flip-Flop으로 구성되어 있다.

각종의 신호들을 먼저 살펴보면 다음과 같다.

- ① System Data: PI, PO에 인가되는 데이터
- ② Clock: SI, PI 데이터를 Latch하여 기억하기 위한 Pulse 신호

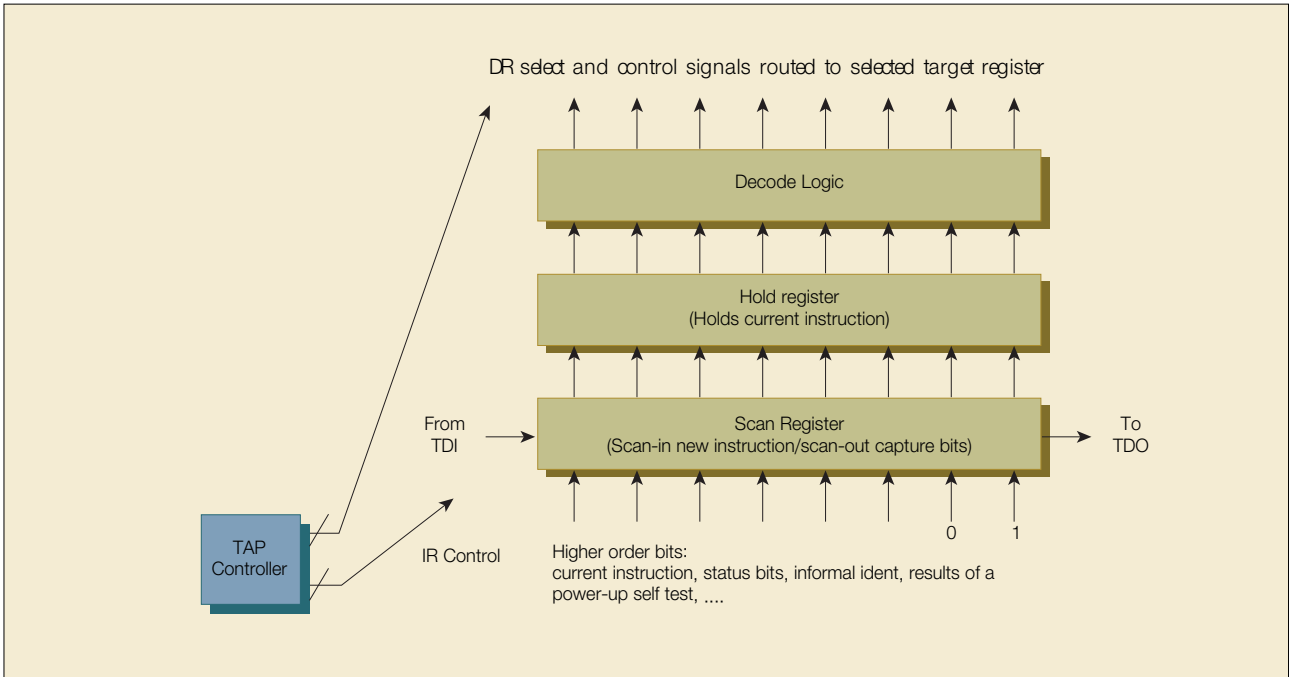


그림 5. Instruction Register 구조

- ③ Update: SI, PI 데이터를 PO에 인가하기 위한 Pulse 신호
- ④ Mode Shift/Load: PI 데이터의 경로를 SI 또는 PO로 설정하기 위한 선택 신호
- ⑤ Mode Test/Normal: SO 또는 PI 데이터를 PO로 설정하기 위한 선택 신호

이제 BSC의 동작에 대해서 알아보자. 각각의 Cell들은 Normal mode, Update mode, Capture mode, Shift mode의 4가지 동작 모드를 가지고 있으며 메모리 구성요소는 데이터의 back-end, front-end multiplexing과 함께 D-flip flop을 사용한다.

- Normal mode: PI에 인가된 데이터는 직접 PO로 pass 된다.
- Update mode: SI로 입력된 데이터를 PO에 인가한다.
- Capture mode: PI는 SI로 router되고 그 값은 next ClockDR에 의해 캡처 된다.
- Shift mode: SI로 입력된 데이터가 SO를 통하여 다음 SI로 입력된다.

### 내부 Register

JTAG은 내부에 4개의 Instruction Register, Identification

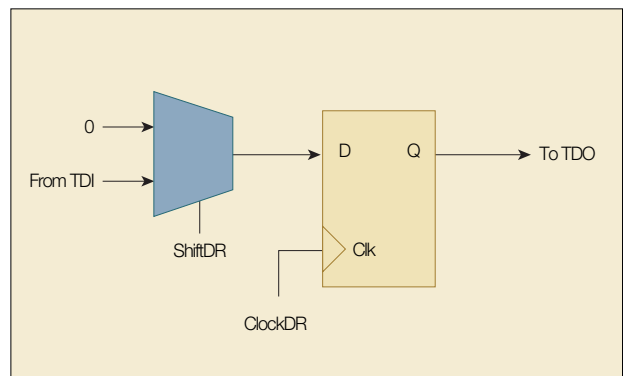


그림 6. Bypass Register 구조

Register, Bypass Register 및 Boundary Scan Register가 존재한다. 이들은 각각 서로 다른 형태의 기능을 제공한다.

### Instruction Register

Instruction Register는 Scan Register, Hold Register, Decoder Logic으로 구성되어 있다. 그림 5는 Instruction Register의 내부 구조를 나타낸다. TDI로부터 명령에 해당되는 데이터가 직렬 shift되어 Scan Register에 인가되고, 이는 다시 Hold Register에 Latch 된다. 최종적으로 Latch된 명령

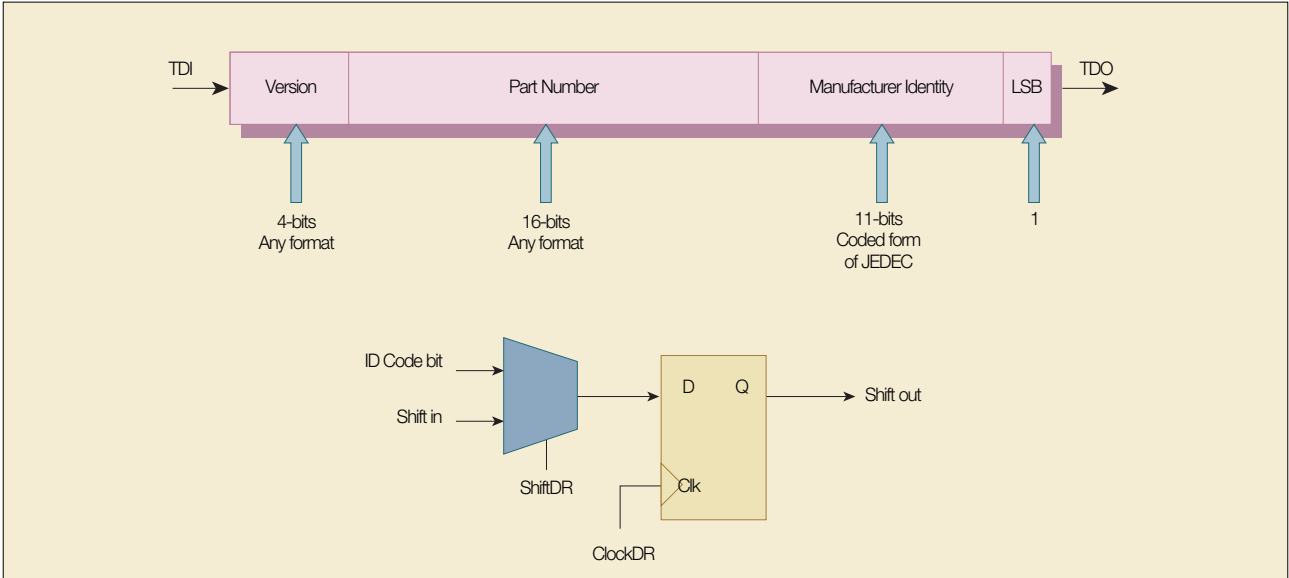


그림 7. Identification Register 구조

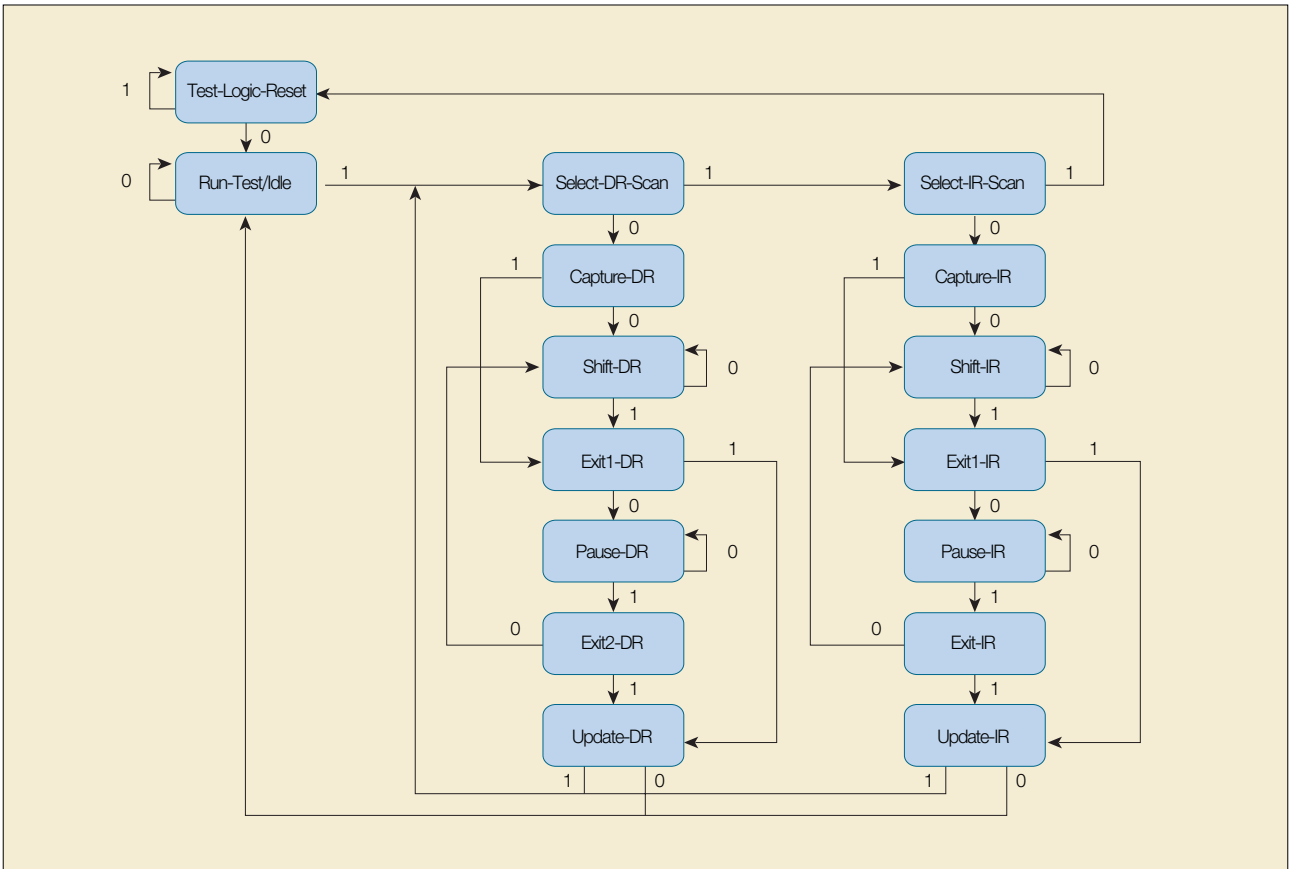


그림 8. State Machine 상태도

TAPC status	Description
Test-Logic-Reset	TAPC 및 JTAG과 관련된 모든 내용을 초기화하며 상태의 시작점이 된다.
Run-Test/Idle	JTAG을 동작상태로 진입시키고 TAP를 통하여 적용한 내용이 각 장비에 영향을 준다.
Select DR-Scan	명령 레지스터에 의해 선택된 boundary scan cell에 제어를 할 것인가 아닌가에 대한 중간 상태 값이고 특별히 하는 기능은 없다.
Capture DR	명령 레지스터에 의해 선택된 boundary scan cell의 PI 단자 값을 내부 시프트 레지스터 쪽으로 적용되게 한다. 즉 현재 상태를 시프트 할 수 있게 준비하는 기능이다.
Shift DR	명령 레지스터에 의해 선택된 boundary scan cell의 내부 값을 SO에 출력시키고 SI값을 내부에 적용할 수 있게 한다. 이 상태로 있을 때 TCLK값을 한 클럭 줄 때마다 TDI값이 명령 레지스터에 의해 선택된 boundary scan cell의 SI에 연결되고 SO의 값이 TDO에 연결된다.
Exit1 DR	상태 변이용 중간 상태이고 특별히 하는 기능은 없다.
Pause DR	상태 변이용 중간 상태이고 특별히 하는 기능은 없다.
Exit2 DR	상태 변이용 중간 상태이고 특별히 하는 기능은 없다.
Update DR	명령 레지스터에 의해 선택된 boundary scan cell의 PO 단자 값에 내부 시프트 레지스터 쪽의 내용을 적용시킨다
Select IR-Scan	명령 레지스터에 제어를 할 것인지 않을 것인지에 대한 중간 상태 값이다. 상태 변이용 중간 상태이고 특별히 하는 기능은 없다.
Capture IR	명령 레지스터의 boundary scan cell의 PI 단자 값을 내부 시프트 레지스터 쪽으로 적용되게 한다. 즉, 현재 상태를 시프트 할 수 있게 준비하는 기능이다.
Shift IR	명령 레지스터의 boundary scan cell의 내부 값을 SO에 출력시키고 SI값을 내부에 적용할 수 있게 한다. 이 상태로 있을 때 TCLK값을 한 클럭 줄 때마다 TDI값이 명령 레지스터 boundary scan cell의 SI에 연결되고 SO의 값이 TDO에 연결된다.
Exit1 IR	상태 변이용 중간 상태이고 특별히 하는 기능은 없다.
Pause IR	상태 변이용 중간 상태이고 특별히 하는 기능은 없다.
Exit2 IR	상태 변이용 중간 상태이고 특별히 하는 기능은 없다.
Update IR	명령 레지스터에 의해 선택된 boundary scan cell의 PO 단자 값에 내부 시프트 레지스터 쪽의 내용을 적용시킨다.

표 4. TAP State Machine

데이터는 Decoder Logic에서 해석되어 해당 명령을 수행하게 된다.

**Bypass Register**

Bypass Register는 그림 6과 같이 Shift 기능이 제공되며, Bypass 명령에 의해서 선택되는 1Bit Register이다. Parallel Output이 없어 Update 데이터 레지스터 Control이 레지스터에 영향을 주지 않지만, 그림 6에서처럼 0의 Capture 데이터 레지스터 Control이 hard-wired value를 capture하는 레지스터에 영향을 주게 된다.

**Identification Register**

Identification Register(그림 7)는 Optional한 Register로서 반드시 존재해야 되는 것은 아니며, 그에 따라 EXTEST와 같은 필수적인 명령과는 달리 IDCODE 명령 역시 Optional 하게 구성된다.

이 Register를 통하여 각종 디바이스의 정보를 얻게 되는데, 그 내용은 다음과 같다.

- Bit 0은 항상 1
- Bit 1~11까지는 JEDEC 식별 코드의 Form을 사용하여 디바이스의 제조사를 식별한다.
- Bit 12~27까지는 16 Bit Free Format Part Number이다.
- Bit 28~31은 같은 디바이스의 Revision 번호를 나타내며, 4 Bit로서 16개의 버전을 지정하는 Free Format Field이다.

**TAPC(Test Access Port Controller)**

TAPC는 TMS, TCK에 입력되는 신호에 의해 내부에 존재하는 state machine의 상태천이가 발생되며 이에 따라 적절한 제어를 하게 된다. state machine은 16가지의 상태를 가지며 그림 8은 state machine의 상태도를 나타낸다. <sup>Ref</sup> <sub>100</sub>