

1. EZFlash

EZFlash란

EZFlash란 JTAG를 사용하여 SA1110의 Boundary CELL을 통하여 하드웨어의 연결상태와 동작을 TEST 할 수 있으며, Flash의 Read, Write기능을 사용하여 부트프로그램을 써 넣기 위하여 만들어진 프로그램이다

EZ-X5 보드를 위한 실행파일은 ezflash.X5 이다.
윈도우용은 EZFlashw 이란 이름을 가진다.

2. EZ-X5 보드의 초기상태 및 EZFlash 동작 환경

보드 동작 확인

EZ-X5에 전원을 공급한 후 JTAG 케이블(FA-365C)를 연결하고, 전원 스위치 켜었을 때 EZ-X5의 LED 4개가 교차로 켜진 후 꺼지면 정상적으로 동작한 상태이다.

JTAG 케이블을 1장의 연결도를 참고하여 리눅스 PC와 연결한다.

3. EZFlash 프로그램 위치

소스 위치

EZFlash의 소스와 실행 프로그램은 배포한 CDROM의 /sw/ezflash.x5 에 위치한다.3

EZFlash의 소스와 실행 프로그램

linux상에서 cdrom을 읽기 위하여 다음과 같이 마운트를 수행하여야 한다.

```
[root@jdt /]# mount /dev/cdrom /mnt/cdrom
mount: block device /dev/cdrom is write-protected, mounting read-only
[root@jdt /]# █
```

EZFlash의 소스와 실행 프로그램은 배포한 CDROM의 /sw/ezflash.x5 에 위치한다.

```
[root@jdt cd_rom]# ls
cross_compiler doc hw sw util
[root@jdt cd_rom]# ls -al
total 28
drwxrwxrwx  7 nobody  nobody    4096 Jul  1 20:22 .
drwxrwxrwx 18 nobody  nobody    4096 Jul  1 20:20 ..
drwxrwxrwx  2 nobody  nobody    4096 Jul  1 20:21 cross_compiler
drwxrwxrwx  2 nobody  nobody    4096 Jul  1 20:22 doc
drwxrwxrwx  2 nobody  nobody    4096 Jul  1 20:22 hw
drwxrwxrwx  9 nobody  nobody    4096 Jul  1 20:26 sw
drwxrwxrwx  2 nobody  nobody    4096 Jul  1 20:22 util
[root@jdt cd_rom]# cd sw/ezflash.x5/
[root@jdt ezflash.x5]# ls -al
total 88
drwxrwxrwx  2 nobody  nobody    4096 Jul  1 20:24 .
drwxrwxrwx  9 nobody  nobody    4096 Jul  1 20:26 ..
-rwxrw-rw-  1 nobody  nobody    4587 Jul  1 20:16 Makefile
-rwxrw-rw-  1 nobody  nobody   12651 Dec 12  2001 jtag.c
-rwxrw-rw-  1 nobody  nobody    2227 Apr 23 23:20 jtag.h
-rwxrw-rw-  1 nobody  nobody   13439 Jun 27 19:04 main.c
-rwxrw-rw-  1 nobody  nobody   15969 Jul  1 20:18 mx291vx.c
-rwxrw-rw-  1 nobody  nobody    1744 Jun 27 18:55 mx291vx.h
-rwxrw-rw-  1 nobody  nobody   11793 Jun  6 00:40 pxa255.c
-rwxrw-rw-  1 nobody  nobody    2248 May  2 13:38 pxa255.h
[root@jdt ezflash.x5]#
```

EZFlash의 실행 프로그램은 배포한 CDROM의 /sw/image/ezflash.x5 에 위치한다.
이 실행파일을 복사한다.

```
[root@jdt image]# cp ezflash.x5 /tmp
```

4. EZFlash 사용 설명

4.1. 초기화면

```
[root@jdt ezflash.x5]# ./ezflash.x5

ezflash pax255 v0.1 ( 2003-06-27)
make by J.D&T Co.,Ltd ( You Young-chang, fooji )

Options
-p      : JTAG port test      [ !!! single option  ]
-c      : PXA255 CELL test    [ !!! single option  ]
-f      : FLASH test          [ !!! single option  ]
-r      : Flash Read          [                    ]
-w      : Flash Write         [ default option     ]
-v      : Flash Verify        [                    ]
-b base  : Flash base Address [ default : 0x00000000 ]
-o offset : Flash write Address [ default : 0x00000000 ]
-s size  : Flash r/w size    [                    ]

> Init JTAG Interface...
> 1 [PWD=/tmp/ezflash.x5]

> Flash Write From File....
> Error!!! -> No File Name

program end
[root@jdt ezflash.x5]#
```

Ezflash.x5 명령 단독으로 수행하면 옵션 설명이 나온다.

각 옵션의 의미는 다음과 같다.

- -p : JTAG 시험모드
- -c : PXA255 CELL 시험 모드
- -f : BOOT FLASH 시험 모드
- -r : FLASH를 읽어 파일에 저장
- -w : FLASH 에 파일 자료 쓰기
- -v : FLASH 내용과 파일 내용 비교
- -b base : FLASH 시작 주소 설정
- -o offset : FLASH 에 쓸 주소 설정
- -s size : FLASH Read/Write 크기 설정

4.2. Options별 기능 설명

4.2.1. -p 옵션

JTAG의 TDI, TMS, TCK, TDO 핀을 제어하여 TAP Controller의 상태 변화를 TEST 하기 위한 옵션이다.

```
[root@jdt ezflash.x5]# ./ezflash.x5 -p

ezflash pax255 v0.1 ( 2003-06-27)
make by J.D&T Co.,Ltd ( You Young-chang, fooji )

Options
-p      : JTAG port test      [ !!! single option  ]
-c      : PXA255 CELL test   [ !!! single option  ]
-f      : FLASH test         [ !!! single option  ]
-r      : Flash Read          [                    ]
-w      : Flash Write         [ default option    ]
-v      : Flash Verify        [                    ]
-b base  : Flash base Address [ default : 0x00000000 ]
-o offset : Flash write Address [ default : 0x00000000 ]
-s size  : Flash r/w size    [                    ]

> Init JTAG Interface...
> 2 [(null)]

# JTAG DEBUG MODE
# 1) TDI -> HIGH  2) TDI -> LOW
# 3) TMS -> HIGH  4) TMS -> LOW
# 5) TCK -> HIGH  6) TCK -> LOW
# r) Change Run Test Idle Mode
# i) Set IR Value
# d) Set DR Value
# h) Command List
# q) Quit

# TDI : [L] / TMS : [L] / TCK : [L] / TCK : [L] / TDO : [H]
Command ?
```

한 Version에서는 JTAG의 상태변화를 콘솔로 확인을 할 수 있으며, 육안으로 확인을 하시고자 하시는 분은 오실로스코프를 이용하여 직접 해당 핀을 측정해보면 된다.

다음 화면은 JTAG Port를 Test 하기 위한 것으로 Command ?에 1~6, r, i, d, h, q를 입력하면 된다..

■ 명령 : 1~6

TDI, TMS, TCK 핀을 테스트하기 위한 명령이며, 이의 확인은 TDO를 통하여 읽은 값을 콘솔로 보여준다.

```
Command ? 1
# TDI : [H] / TMS : [L] / TCK : [L] / TCK : [L] / TDO : [H]
Command ? 2
# TDI : [L] / TMS : [L] / TCK : [L] / TCK : [L] / TDO : [H]
Command ? 3
# TDI : [L] / TMS : [H] / TCK : [L] / TCK : [L] / TDO : [H]
Command ? 4
# TDI : [L] / TMS : [L] / TCK : [L] / TCK : [L] / TDO : [L]
Command ? 5
# TDI : [L] / TMS : [L] / TCK : [H] / TCK : [H] / TDO : [H]
Command ? 6
# TDI : [L] / TMS : [L] / TCK : [L] / TCK : [L] / TDO : [H]
Command ?
```

■ 명령 : r

TAP Controller State Transitions를 Run-Test/Idle Mode 상태로 바꾸는 명령이다.

3의 Test-Logic Reset 상태에서 JTAG 제어가 가능한 1.4의 Run-Test/Idle 상태로 이동하는 것을 일괄 처리한 명령이다.

[APPENDIX 1.8. Test Access Port(TAP) Controller State Transitions 참조]

```
# JTAG DEBUG MODE
# 1) TDI -> HIGH    2) TDI -> LOW
# 3) TMS -> HIGH    4) TMS -> LOW
# 5) TCK -> HIGH    6) TCK -> LOW
# r) Change Run Test Idle Mode
# i) Set IR Value
# d) Set DR Value
# h) Command List
# q) Quit

# TDI : [L] / TMS : [L] / TCK : [L] / TCK : [L] / TDO : [H]
Command ? r
```

■ 명령 : i

Instruction Register[IR] 값을 입력한다.

Instruction Register[IR]의 길이는 5bit이며, No Parity bit이다.

[APPENDIX 1.9. Access IR 참조]

다음과 같은 일반적인 Instruction을 수행한다.

Instruction	Binary Code
EXTEST	00000
SAMPLE/PRELOAD	00001
CLAMP	00100
HIGHZ	00101
IDCODE	00110
BYPASS	11111
Private	00010, 00011, 00111, 01000-01111, 10000-11110

위의 Instruction Register값 중 하나를 입력하면 다음과 같은 값을 읽을 수 있다. [입력 값이 다를 수도 있음]

```
# JTAG DEBUG MODE
# 1) TDI -> HIGH    2) TDI -> LOW
# 3) TMS -> HIGH    4) TMS -> LOW
# 5) TCK -> HIGH    6) TCK -> LOW
# r) Change Run Test Idle Mode
# i) Set IR Value
# d) Set DR Value
# h) Command List
# q) Quit

# TDI : [L] / TMS : [L] / TCK : [L] / TCK : [L] / TDO : [H]
Command ? i
? IR Pattern ?[ ex : 00010 ] : 00010
Read IR 00001
Command ?
```

■ 명령 : d

Test Data Register[DR] 값을 입력한다. [APPENDIX 1.10. Access DR 참조]

```
# JTAG DEBUG MODE
# 1) TDI -> HIGH    2) TDI -> LOW
# 3) TMS -> HIGH    4) TMS -> LOW
# 5) TCK -> HIGH    6) TCK -> LOW
# r) Change Run Test Idle Mode
# i) Set IR Value
# d) Set DR Value
# h) Command List
# q) Quit

# TDI : [L] / TMS : [L] / TCK : [L] / TCK : [L] / TDO : [H]
Command ? r
Command ? i
? IR Pattern ?[ ex : 00010 ] : 11111
Read IR 00001
Command ? d
? DR Pattern ?[ ex : 10010 ] : 10010
Read DR 00100
Command ? d
? DR Pattern ?[ ex : 10010 ] : 01110
Read DR 11100
Command ? d
? DR Pattern ?[ ex : 10010 ] : 00101
Read DR 01010
Command ?
```

위 화면은 Test Data Register[DR]를 TEST하기 위한 것이다.

1. 명령 [r]은 JTAG 제어가 가능한 Run-Test/Idel 상태로 이동한 것이다
2. 명령 [i]는 Run-Test/Idel >> Select-DR-Scan >> Select-IR-Scan >> Capture-IR >> Shift-IR 상태로 이동한 것이다.
3. ? IR Pattern ?[ex : 00010] : 에서 BYPASS명령인 [11111]을 입력한다.
4. 명령 [d]는 Run-Test/Idel >> Select-DR-Scan >> Capture-DR >> Shift-DR 상태로 이동한 것이다.
5. ? DR Pattern ?[ex : 10010] : 에서 [10010]을 입력한다.
6. 읽은 값이 왼쪽으로 1만큼 쉬프트 한 값 [00100]을 읽었다면 정상이다.
7. ? DR Pattern ?[ex : 10010] : 에서 [01110]을 입력한다.
8. 읽은 값이 왼쪽으로 1만큼 쉬프트한 값 [11100]을 읽었다면 정상이다.
9. 이제 사용자 원하시는 값을 넣고, 읽은 값이 왼쪽으로 1만큼 쉬프트한 값인지를 직접 확인하면 된다.

[? DR Pattern ?에 5bit 보다 큰 bit를 써넣어도 됩니다. Ex] 011000100]

■ 명령 : h

명령 목록을 다시 보여 준다. [이후 모든 옵션에서 적용이 됨]

```
Command ? h
# JTAG DEBUG MODE
# 1) TDI -> HIGH   2) TDI -> LOW
# 3) TMS -> HIGH   4) TMS -> LOW
# 5) TCK -> HIGH   6) TCK -> LOW
# r) Change Run Test Idle Mode
# i) Set IR Value
# d) Set DR Value
# h) Command List
# q) Quit
```

```
Command ? █
```

■ 명령 : q

프로그램을 종료한다. [이후 모든 옵션에서 적용이 됨]

```
# JTAG DEBUG MODE
# 1) TDI -> HIGH   2) TDI -> LOW
# 3) TMS -> HIGH   4) TMS -> LOW
# 5) TCK -> HIGH   6) TCK -> LOW
# r) Change Run Test Idle Mode
# i) Set IR Value
# d) Set DR Value
# h) Command List
# q) Quit
```

```
Command ? q
```

```
program end
[root@jdt ezflash.x5]# █
```


4.2.2. -c 옵션

PXA255의 CELL을 TEST하기 위한 옵션이다.

```
[root@jdt ezflash.x5]# ./ezflash.x5 -c

ezflash pax255 v1.0 ( 2003-06-27)
make by J.D&T Co.,Ltd ( You Young-chang, fooji )

Options
-p      : JTAG port test      [ !!! single option  ]
-c      : PXA255 CELL test    [ !!! single option  ]
-f      : FLASH test          [ !!! single option  ]
-r      : Flash Read          [                    ]
-w      : Flash Write         [ default option     ]
-v      : Flash Verify        [                    ]
-b base : Flash base Address  [ default : 0x00000000 ]
-o offset : Flash write Address [ default : 0x00000000 ]
-s size  : Flash r/w size     [                    ]

> Init JTAG Interface...
> 2 [(null)]

# JTAG Change Run Test Idle Mode
# PXA255 DEBUG MODE
# 1) Set EXTEST  2) Set SAMPLE/PRE LOAD
# 3) Set CLAMP   4) Set HIGHZ
# 5) Set IDCODE  6) Set BYPASS
# 7) Get PXA255 ID
# 8) Bypass Test
# h) Command List
# q) Quit

Command ?
```

- 1) Set EXTEST : 칩외부 회로 시험을 하기 위한 명령이다.
- 2) Set SAMPLE/PRE LOAD: 핀/프리로드 데이터를 샘플링 하기 위한 명령이다.
- 3) Set CLAMP
- 4) Set HIGHZ : 하이임피던스/ONCE 모드 시험을 하기 위한 명령이다.
- 5) Set IDCODE : IDCODE를 시험하기 위한 명령이다.
- 6) Set BYPASS : 칩의 시스템 논리 회로를 우회하기 위한 명령이다.
- 7) Get PXA255 ID : PXA255의 ID를 읽기위한 명령이다.
- 8) Bypass Test : BYPASS를 시험하기 위한 명령이다.

현재 TAP는 Run-Test/Idle상태에 있다.

■ 명령 : 1

EXTEST(00000) Instruction에 의해 Boundary-Scan(BS) Register는 Test Mode로 전환이 된다.

이 명령은 TDI와 TDO사이에 BS Register로 연결된다.

```
# JTAG Change Run Test Idle Mode
# PXA255 DEBUG MODE
# 1) Set EXTEST  2) Set SAMPLE/PRE LOAD
# 3) Set CLAMP   4) Set HIGHZ
# 5) Set IDCODE  6) Set BYPASS
# 7) Get PXA255 ID
# 8) Bypass Test
# h) Command List
# q) Quit
```

```
Command ? 1
# IR Setting..Value [00000]
Command ?
```

Instruction Registers가 EXTEST 명령어일 경우에 BS Cell은 동작 중에 Test Mode로 전환이 된다.

MODE별 동작 상태

CAPTURE-DR 상태일 경우 : 시스템 핀으로부터의 입력과 시스템 핀에 대한 Cell에 로드된다.

SHIFT-DR 상태일 경우 : 이전에 적재된 테스트 데이터는 BS Register로 부터 TDO핀으로 쉬프트 되고, 새로운 데이터는 TDI를 통해서 BS Register Parallel Input Latch로 쉬프트 된다.

Update-DR 상태일 경우 : 새로운 테스트 데이터는 BS Register Parallel Output Latch로 전송된다.

주의한 점은 이 같은 데이터는 즉시 System Logic 과 System pins에 적용된다.

■ 명령 : 2

SAMPLE/PRE LOAD(00001) Instruction에 의해 BS Register는 정상모드로 전환된다.

이 명령은 TDI와 TDO사이의 BS Register로 연결된다.

```
# JTAG Change Run Test Idle Mode
# PXA255 DEBUG MODE
# 1) Set EXTEST  2) Set SAMPLE/PRE LOAD
# 3) Set CLAMP   4) Set HIGHZ
# 5) Set IDCODE  6) Set BYPASS
# 7) Get PXA255 ID
# 8) Bypass Test
# h) Command List
# q) Quit
```

```
Command ? 2
# IR Setting..Value [00001]
Command ? █
```

Instruction Registers가 이 명령을 가지고 Load 된다면 모든 BS Cell은 일반 동작 모드 상태로 전환이 된다.

MODE별 동작 상태

CAPTURE-DR 상태일 경우 : BS Cell이 신호를 받아 들일 때는 TCK가 rising edge 상태일 경우이고, 정상적인 시스템 동작에는 영향을 받지 않는다.

SHIFT-DR 상태일 경우 : 샘플된 데이터는 BS Register에서 TDO를 통해서 BS Register Parallel Input Latch로 쉬프트 된다.

Update-DR 상태일 경우 : 미리 로드된 데이터는 BS Register Parallel Output Latch로 전달된다.

주의할 점은 이 데이터는 SAMPLE/PRELOAD 명령이 활성화 된 상태라면 System Logic과 Sytem Pins에 적용되지 않는다.

■ 명령 : 3

CLAMP(00100) Instruction은 1bit shift register(the BYPASS register)를 TDI와 TDO사이에 연결을 시킨다.

```
# JTAG Change Run Test Idle Mode
# PXA255 DEBUG MODE
# 1) Set EXTEST  2) Set SAMPLE/PRE LOAD
# 3) Set CLAMP   4) Set HIGHZ
# 5) Set IDCODE  6) Set BYPASS
# 7) Get PXA255 ID
# 8) Bypass Test
# h) Command List
# q) Quit

Command ? 3
# IR Setting..Value [00100]
Command ?
```

이 명령이 Instruction Register에 로드 되었을 때 모든 출력 신호의 상태는 BS Register에 로드 된 값에 의해 미리 정의된다..

MODE별 동작 상태

CAPTURE-DR 상태일 경우 : Logic 0가 Bypass Register에 의해 강제로 로드된다.

SHIFT-DR 상태일 경우 : 테스트 데이터는 TDI를 통해서 Bypass Register로 쉬프트되고, 다시 하나의 TCK Cycle 후에 TDO를 통해서 출력된다.

주의할 점은 쉬프트된 첫 번째 bit는 0일 것이다.

Update-DR 상태일 경우 : Bypass Register는 Update-DR상태에서는 영향을 받지 않는다.

■ 명령 : 4

HIGHZ(00101) Instruction은 TDI와 TDO사이에 Device Identification Register로 연결 한다.

```
# JTAG Change Run Test Idle Mode
# PXA255 DEBUG MODE
# 1) Set EXTEST  2) Set SAMPLE/PRE LOAD
# 3) Set CLAMP   4) Set HIGHZ
# 5) Set IDCODE  6) Set BYPASS
# 7) Get PXA255 ID
# 8) Bypass Test
# h) Command List
# q) Quit
```

```
Command ? 4
# IR Setting..Value [01000]
Command ?
```

이 명령이 Instruction Register에 의해 로드 될 경우 모든 출력이 비활성 구동 상태에 놓인다.

MODE별 동작 상태

CAPTURE-DR 상태일 경우 : Logic 0가 Bypass Register에 의해 강제로 로드된다.

SHIFT-DR 상태일 경우 : 테스트 데이터는 TDI를 통해서 Bypass Register로 쉬프트되고, 다시 하나의 TCK Cycle 후에 TDO를 통해서 출력된다.

주의할 점은 쉬프트된 첫 번째 bit는 0일 것이다.

Update-DR 상태일 경우 : Bypass Register는 Update-DR상태에서는 영향을 받지 않는다.

■ 명령 : 5

IDCODE(00110) Instruction은 TDI와 TDO사이에 연결된다.

Identification(ID) Register는 manufacturer, part number, version값을 저장하고 있는 32비트 register이다.

```
# JTAG Change Run Test Idle Mode
# PXA255 DEBUG MODE
# 1) Set EXTEST  2) Set SAMPLE/PRE LOAD
# 3) Set CLAMP   4) Set HIGHZ
# 5) Set IDCODE  6) Set BYPASS
# 7) Get PXA255 ID
# 8) Bypass Test
# h) Command List
# q) Quit
```

```
Command ? 5
# IR Setting..Value [11110]
Command ?
```

이 명령을 가지고 Instruction Register가 로드 되면, 모든 BS-Cell은 동작 중에 정상모드로 전환된다.

MODE별 동작 상태

CAPTURE-DR 상태일 경우 : Device Identification Code는 ID Register로 강제 로드 된다.

SHIFT-DR 상태일 경우 : 이전에 로드된 Device Identification Code가 ID Register로 강제 로드되면 데이터는 TDI 핀을 통해서 ID Register로 쉬프트 된다

Update-DR 상태일 경우 : ID Register는 영향을 받지 않는다.

■ 명령 : 6

BYPASS(11111) Instruction은 1bit 레지스터를 TDI와 TDO사이에 연결된다.

```
# JTAG Change Run Test Idle Mode
# PXA255 DEBUG MODE
# 1) Set EXTEST  2) Set SAMPLE/PRE LOAD
# 3) Set CLAMP   4) Set HIGHZ
# 5) Set IDCODE  6) Set BYPASS
# 7) Get PXA255 ID
# 8) Bypass Test
# h) Command List
# q) Quit
```

```
Command ? 6
# IR Setting..Value [11111]
Command ?
```

이 명령이 Instruction Register에 로드 되었을 때, 모든 BS-Cell은 동작 중에 정상 모드로 전환된다.

이 명령은 System Pins에 영향을 주지 않는다.

MODE별 동작 상태

CAPTURE-DR 상태일 경우 : Logic 0가 Bypass Register에 의해 강제로 로드된다.

SHIFT-DR 상태일 경우 : 테스트 데이터는 TDI를 통해서 Bypass Register로 쉬프트되고, 다시 하나의 TCK Cycle 후에 TDO를 통해서 출력된다.

주의할 점은 쉬프트된 첫 번째 bit는 0일 것이다.

Update-DR 상태일 경우 : Bypass Register는 Update-DR상태에서는 영향을 받지 않는다.

■ 명령 : 7

PXA255 Device Identification(ID) Code Register의 32비트를 읽기 위한 명령이다.
명령을 실행 시키면 다음과 같은 화면이 나타난다.

```
# JTAG Change Run Test Idle Mode
# PXA255 DEBUG MODE
# 1) Set EXTEST  2) Set SAMPLE/PRE LOAD
# 3) Set CLAMP   4) Set HIGHZ
# 5) Set IDCODE  6) Set BYPASS
# 7) Get PXA255 ID
# 8) Bypass Test
# h) Command List
# q) Quit

Command ? 7
Get ID Register Value [01101001001001100100000000010011]
Stepping Value      [0110] :
Part Number [1001001001100100] : PXA255
Manufacturer ID [00000001001] : Intel Corporation
Command ?
```

■ 명령 : 8

Bypass Test를 하기 위한 명령이다.
명령을 실행 시키면 다음과 같은 화면이 나타난다.

```
# JTAG Change Run Test Idle Mode
# PXA255 DEBUG MODE
# 1) Set EXTEST  2) Set SAMPLE/PRE LOAD
# 3) Set CLAMP   4) Set HIGHZ
# 5) Set IDCODE  6) Set BYPASS
# 7) Get PXA255 ID
# 8) Bypass Test
# h) Command List
# q) Quit

Command ? 8
Put Bypass Value [ 01001000101010110111011011001011 ]
Get Bypass Value [ 10010001010101101110110110010110 ]
Bypass Passing Ok.
Command ?
```

이 TEST의 시퀀스는 다음과 같다.

1. TAPC를 Run-Test/Idle 상태로 만든다.
2. Access IR를 이용하여 Instruction Register(IR)에 BYPASS [11111] 명령을 넣는다.
3. Access DR를 이용하여 임의의 패턴을 밀어 넣고, 나온 패턴이 왼쪽으로 1bit 쉬프트한 값이 나오는가?
위의 화면과 같은 값이 나온다면 JTAG 인터페이스가 존재한다.

4.2.3. -f 옵션

INTEL FLASH의 읽기, 쓰기, 검지를 TEST하기 위한 옵션이다.

```
[root@jdt ezflash.x5]# ./ezflash.x5 -f

ezflash pax255 v1.0 ( 2003-06-27)
make by J.D&T Co.,Ltd ( You Young-chang, fooji )

Options
-p      : JTAG port test      [ !!! single option  ]
-c      : PXA255 CELL test    [ !!! single option  ]
-f      : FLASH test          [ !!! single option  ]
-r      : Flash Read          [                    ]
-w      : Flash Write         [ default option     ]
-v      : Flash Verify        [                    ]
-b base : Flash base Address  [ default : 0x00000000 ]
-o offset : Flash write Address [ default : 0x00000000 ]
-s size  : Flash r/w size     [                    ]

> Init JTAG Interface...
> 2 [(null)]

# JTAG Change Run Test Idle Mode
# IR Setting..Value [00000]
# Flash DEBUG MODE
# r) Flash Read
# w) Flash Write
# d) Flash Detect Test
# h) Command List
# q) Quit

Command ?
```

- r) Flash Read : FLASH 읽기
- w) Flash Write : FLASH 쓰기
- d) Intel Flash Detect Test : FLASH 검지
- h) Command List : 명령 목록을 다시 보여 준다.
- 5) Quit : 프로그램을 종료한다.

다음 화면은 INTEL FLASH Test를 하기 위한 것으로 Command ?에 r, w, d, h, q 를 입력한다.

현재 TAP는 Run-Test/Idle Mode 상태이며, Instruction Register는 EXTEST(00000)이다.

■ 명령 : r

Flash에 기록된 데이터를 읽어오는 명령이다.

명령을 실행 시키면 다음과 같은 화면이 나타난다.

```
# JTAG Change Run Test Idle Mode
# IR Setting..Value [00000]
# Flash DEBUG MODE
# r) Flash Read
# w) Flash Write
# d) Flash Detect Test
# h) Command List
# q) Quit

Command ? r
? Read Start Address ?[ ex) 000 ] : 0
? Read End Address ?[ ex) 800 ] : 100
0000 0000 :06 00 00 EA 33 00 00 EA 32 00 00 EA 31 00 00 EA ....3...2...1...
0000 0010 :2D 00 00 EA 2F 00 00 EA 2E 00 00 EA 2D 00 00 EA -.../.....-...
0000 0020 :00 00 0F E1 1F 00 C0 E3 D3 00 80 E3 00 F0 29 E1 .....).
0000 0030 :AC 00 9F E5 AC 10 9F E5 00 10 80 E5 02 00 A0 E3 .....
0000 0040 :10 0E 06 EE 0E 1D 01 E2 01 0C 51 E3 01 00 00 0A .....Q....
0000 0050 :01 00 A0 E3 10 0E 06 EE 15 1F 07 EE 10 0F 12 EE .....
0000 0060 :00 00 A0 E1 04 F0 4F E2 10 0F 11 EE 01 0A 80 E3 .....0.....
0000 0070 :10 0F 01 EE 10 0F 12 EE 00 00 A0 E1 04 F0 4F E2 .....0.
0000 0080 :01 50 A0 E3 34 00 00 EB 19 00 00 EB 56 00 00 EB .P..4.....V...
0000 0090 :02 50 A0 E3 30 00 00 EB 94 00 00 EB 03 50 A0 E3 .P..0.....P..
0000 00A0 :2D 00 00 EB AC 00 00 EB B2 00 00 EB 38 00 9F E5 -.....8...
0000 00B0 :04 D0 40 E2 05 50 A0 E3 27 00 00 EB 2C 00 9F E5 ..@..P..'.....
0000 00C0 :02 0B 80 E2 00 F0 A0 E1 35 00 00 EA 06 50 A0 E3 .....5....P..
0000 00D0 :21 00 00 EB 32 00 00 EA 07 50 A0 E3 1E 00 00 EB !...2....P.....
0000 00E0 :2F 00 00 EA 00 00 30 41 61 01 00 00 00 00 F2 A0 /.....0Aa.....
0000 00F0 :00 00 F0 A0 E4 00 9F E5 02 11 A0 E3 54 10 80 E5 .....T...
0000 0100 :00 10 A0 E3 58 10 80 E5 08 10 A0 E3 5C 10 80 E5 ....X.....\...
Command ?
```

Read Start Address 와 Read End Address에 사용자 임의로 주소를 입력하시면 원하는 부분을 읽을 수 있다.

[Read Start Address < Read End Address]

■ 명령 : w

Intel Flash에 데이터를 기록하는 명령이다.

```
# JTAG Change Run Test Idle Mode
# IR Setting..Value [00000]
# Flash DEBUG MODE
# r) Flash Read
# w) Flash Write
# d) Flash Detect Test
# h) Command List
# q) Quit

Command ? w
? Write Base   Address ?[ ex) 000 ] : 0
? Write Offset Address ?[ ex) 800 ] : 40000
? Write Size           ?[ ex) 256 ] : 256
Detect MX29LV400T Flash : 22B9
SIZE 4M-BIT [512Kbyte]
Erase : OK
Write : OK
Command ?
```

명령 w를 입력하면 아래의 3가지 질문을 요구한다.

```
? Write Base   Address ?[ ex) 000 ] : 0
? Write Offset Address ?[ ex) 800 ] : 40000
? Write Size           ?[ ex) 256 ] : 256
```

첫번째, Base Address란 Flash에 데이터를 기록할 시작 주소를 의미한다.

두번째, Offset Address란 Flash에 데이터를 기록할 시작 주소에서부터 Flash에 기록하고자하는 실질적인 주소의 위치를 의미한다.

세번째, Size란 Base Address + Offset Address 주소에서부터 시작을 하여 Flash에 데이터를 기록할 크기를 의미한다.

즉, 이 의미는 시작주소 0x00000000 번지 + Offset 주소 0x00000000 번지 = 0x00000000에서부터 256 크기 만큼 Flash에 데이터를 써 넣으라는 의미이다.

자 그럼 이 결과를 읽어보도록 하자.

```
Command ? r
? Read Start Address ?[ ex) 000 ] : 40000
? Read End   Address ?[ ex) 800 ] : 40100
0004 0000 :00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F .....
0004 0010 :10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F .....
0004 0020 :20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F  !"#$%&'()*+,-./
0004 0030 :30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F  0123456789:;<=>?
0004 0040 :40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F  @ABCDEFGH IJKLMNO
0004 0050 :50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F  PQRSTU VWXYZ[\]^_
0004 0060 :60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F  'abcdefg hijklmno
0004 0070 :70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F  pqrstuvwxyz{|}~.
```

주소 0x000000번지에서 000000FF[SIZE : 256]까지 Flash에 데이터를 써 넣은 것을 확인 할 수 있다.

■ 명령 : d

Intel Flash의 Detect 명령이다.

명령을 실행 시키면 다음과 같은 화면이 나타난다.

```
# JTAC Change Run Test Idle Mode
# IR Setting..Value [00000]
# Flash DEBUG MODE
# r) Flash Read
# w) Flash Write
# d) Flash Detect Test
# h) Command List
# q) Quit

Command ? d
? Base Address ?[ ex) 000 ] : 0
Detect MX29LV400T Flash : 22B9
SIZE 4M-BIT [512Kbyte]
Command ?
```

■ 명령 : h

명령 목록을 다시 보여 준다.

■ 명령 : q

프로그램을 종료 한다.

4.2.4. -w 옵션

FLASH에 파일의 DATA를 기록하기 위한 옵션이다.

먼저 FLASH에 쓸 이미지 파일을 현재의 디렉토리에 복사한다.

```
[root@jdt ezflash.x5]# cp /mnt/cdrom/sw/image/ezboot.x5 ./
[root@jdt ezflash.x5]# ./ezflash.x5 ezboot.x5 -w
```

```
> Flash Write From File....
> Setting Data
> Base Address : 0
> Offset Address : 0
> Size : 58692
> File Name : ezboot.x5
Put Bypass Value [ 0100100010101011011011011001011 ]
Get Bypass Value [ 1001000101010110110110110100110 ]
Bypass Passing Ok.

Get ID Register Value [011010010010011001000000000100110]
Stepping Value [0110] :
Part Number [1001001001100100] : PXA255
Manufacturer ID [00000001001] : Intel Corporation

# IR Setting..Value [00000]
Detect MX29LV400T Flash : 22B9
SIZE 4M-BIT [512Kbyte]

Erase : OK
Write : OK
> Flash Write End.....

program end
[root@ezboard /root]$
```

FLASH에 Write하는 명령은

[./ezflash.x5 써넣을 파일명 { -w -b 시작주소 -o Offset 주소 -s DATA 크기 }]

참고. { } 의 내용은 생략 가능하다.

위의 화면은

1. Base Address, Offset Address, Size를 지정하지 않으면 디폴트값으로 Address는 0x0번지에서부터 시작을 하며, Size는 파일명의 크기가 자동으로 지정된다.
2. FLASH에 Write하기전에 BYPASS를 시험하여 JTAG의 인터페이스를 확인한다.
3. BYPASS TEST가 끝나면 IDCODE를 읽어 PXA255의 존재 유무를 확인한다.
4. IDCODE TEST가 끝나면 FLASH의 존재 유무를 확인하는 시험을 진행한다.
5. 모든 진행이 정상적이면, FLASH에 기록되어 있던 DATA를 삭제한다.
6. 삭제가 끝나면, FLASH에 DATA를 써 넣기 시작한다.
7. 모든 작업이 끝나면, > Flash Write End.....메시지와 함께 프로그램 종료된다.

4.2.5. -r 옵션

FLASH에 기록된 내용을 읽어 파일에 저장하는 옵션이다.

위의 4.2.4에서 정상적으로 파일의 DATA가 FLASH에 쓰여졌는지를 알아보기 위하여 FLASH에 기록된 DATA를 파일로 저장을 하기 위한 것이다.

```
[root@jdt ezflash.x5]# ./ezflash.x5 -r ezboot.x5.read -b 0 -s 58692
```

```
> Init JTAG Interface...
> 7 [ezboot.x5.read]

> Flash Read and Save File....
> Setting Data
> Base Address : 0
> Offset Address : 0
> Size : 58692
> File Name : ezboot.x5.read
Put Bypass Value [ 0100100010101011011011011001011 ]
Get Bypass Value [ 1001000101010110110110110010110 à @Xô @
Bypass Passing Ok.

Get ID Register Value [01101001001001100100000000010011Xô @
Stepping Value [0110] :
Part Number [1001001001100100] : PXA255
Manufacturer ID [00000001001] : Intel Corporation

# IR Setting..Value [00000]
Detect MX29LV400T Flash : 22B9
SIZE 4M-BIT [512Kbyte]

> Flash Read End.....

program end
[root@ezboard /root]$
```

FLASH에서 Read하는 명령은

[./ezflash.x5 -r 저장할 파일명 { -b 시작주소 -o Offset 주소} -s 파일의 크기]

참고. {} 는 생략 가능하다.

위의 화면은

1. FLASH에 Read하기전에 BYPASS를 시험하여 JTAG의 인터페이스를 확인한다.
2. BYPASS TEST가 끝나면 IDCODE를 읽어 PXA255의 존재 유무를 확인한다.
3. IDCODE TEST가 끝나면 FLASH의 존재 유무를 확인하는 시험을 진행한다.
4. 모든 진행이 정상적이면, FLASH에 기록된 DATA를 읽어 지정한 파일로 저장
5. 모든 작업이 끝나면, > Flash Read End.....메시지와 함께 프로그램 종료된다.

4.2.6. -v 옵션

FLASH에 기록된 내용과 파일로 저장된 내용을 비교하기 위한 옵션이다.

위의 [4.2.4] FLASH의 쓰기 와 [4.2.5] FLASH의 읽기 작업이 제대로 이루어졌는지를 검증하기 위한 것이다.

```
[root@jdt ezflash.x5]# ./ezflash.x5 -v ezboot.x5
```

[FLASH에 기록된 내용과 FLASH를 읽어 파일로 저장한 DATA가 일치할 경우]

```
> Flash Verify From File....
> Setting Data
> Base Address : 0
> Offset Address : 0
> Size : 58692
> File Name : ezboot.x5
Put Bypass Value [ 01001000101010110111011011001011 ]
Get Bypass Value [ 10010001010101101110110110010110 ]
Bypass Passing Ok.

Get ID Register Value [01101001001001100100000000010011]
Stepping Value [0110] :
Part Number [1001001001100100] : PXA255
Manufacturer ID [00000001001] : Intel Corporation

# IR Setting..Value [00000]
Detect MX29LV400T Flash : 22B9
SIZE 4M-BIT [512Kbyte]

> Flash Verify End.....

program end
[root@ezboard /root]$
```

FLASH를 Verify하는 명령은

[./ezflash.x5 -v 저장한 파일명 { -b 시작주소 -o Offset 주소 -s DATA 크기 }]
참고. {} 는 생략 가능하다.

위의 화면은

1. FLASH에 Read하기전에 BYPASS를 시험하여 JTAG의 인터페이스를 확인한다.
2. BYPASS TEST가 끝나면 IDCODE를 읽어 PXA255의 존재 유무를 확인한다.
3. IDCODE TEST가 끝나면 FLASH의 존재 유무를 확인하는 시험을 진행한다.
4. 모든 진행이 정상적이면, FLASH에 기록된 DATA와 저장된 파일을 비교한다.
5. 모든 작업이 끝나면, > Verify End.....메시지와 함께 프로그램 종료된다.

4.2.7. -b base 옵션

FLASH에 기록하거나 읽기 위한 시작 주소를 설정하기 위한 옵션이다.
단독으로는 사용할 수 없으며, FLASH의 -r, -w, -v 옵션과 함께 사용된다.

4.2.8. -o offset 옵션

FLASH에 실질적으로 기록하거나 읽기 위한 실제주소를 설정하기 위한 옵션이다.
즉, [시작 주소 + Offset 주소 = 실제 주소]가 된다.
단독으로는 사용할 수 없으며, FLASH의 -r, -w, -v 옵션과 함께 사용된다.

4.2.9. -s size 옵션

FLASH의 Read/Write 크기를 설정하는 옵션이다.
FLASH는 실제주소 + Size 만큼 Read/Write 하게 된다.
단독으로는 사용할 수 없으며, FLASH의 -r, -w, -v 옵션과 함께 사용된다.

다음 예제는 [4.2.6, 4.2.7, 4.2.8]의 옵션들을 시험하기 위한 것이다.

1. FLASH에 Base Address : 0번지, Offset Address : 0번지, Size : 2048의 DATA를 쓴다.

```
[root@jdt ezflash.x5]# ./ezflash.x5 ezboot.x5 -w -b 0 -o 0 -s 2048
```

2. FLASH에서 Base Address : 0번지, Offset Address : 0번지, Size : 1024에 쓴 DATA를 읽어어 ezboot_1이라는 파일명으로 저장한다.

```
[root@jdt ezflash.x5]# ./ezflash.x5 -r ezboot.x5 -b 0 -o 0 -s 2048
```

3. 쓰고, 읽은 DATA를 비교한다.

```
[root@jdt ezflash.x5]# ./ezflash.x5 -v ezboot.x5 -b 0 -o 0 -s 2048
```