PLX 9054 PCI KIT V3.0

www.edukit.co.kr

TEL : 0505-586-8086

목 차

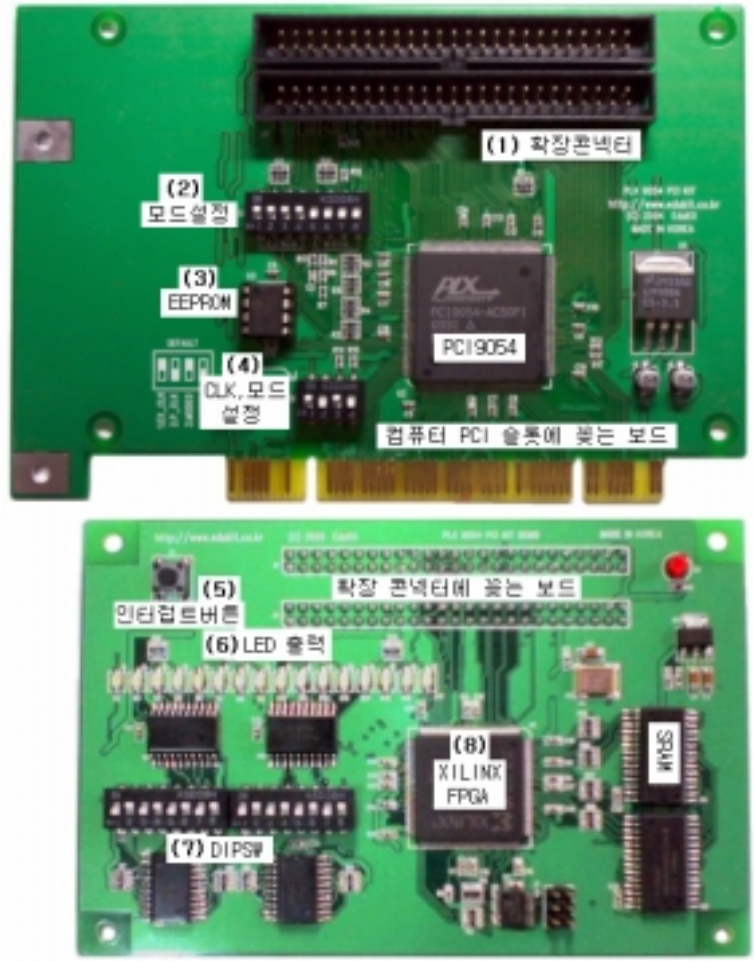
참고 1 : 추가로 업그레이드 되는 소프트웨어는 인터넷 http://www.edukit.co.kr 에서
다운 받으실 수 있습니다.

참고 2 : XP에서 CD가 읽어지지 않는 경우가 있는데, 그 CD를 넣고 리부트하여 보면 내용이
읽혀집니다. XP OS에서만 그러하며, 다른 컴퓨터나 다른 OS를 통하여 읽을 때에는 이상 없습니다.

1. PLX 9054 KIT V3.0 에 대하여

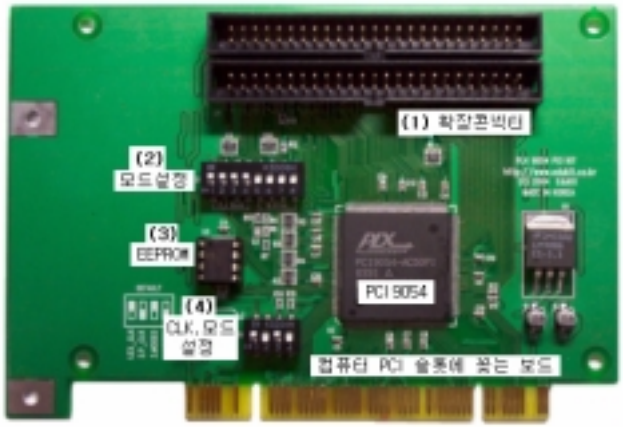
PLX 9054 KIT V3.0는 PLX Technology 사의 PCI Bus Master Interface Chip인 PCI 9054 소자를 사용한 PCI 실험키트이다. PCI 9030 등의 PCI Target Interface에 익숙한 하드웨어와 소프트웨어 기술자들이 PCI 버스 인터페이스를 이용한 DMA 전송을 적은 비용과 시간에 익힐 수 있도록 고안되었다. PLX 9054 KIT는 PLX 9054보드, Demo FPGA 보드, 데모 프로그램과 각종 윈도우 드라이버가 들어있는 CD, 설명서로 구성되어 있다.

PLX 9054 KIT V3.0 는 SRAM을 Read, Write ,16비트 입출력포트, 푸시버튼을 이용한 인터럽트 발생 및 제거 ,그리고 XILINX FPGA내부의 512 x 32 bit Block Ram 의 입출력 테스트를 해 볼 수 있다.



2. PLX 9054 Board 설명

2-1. PCI 9054 board



PCI버스에 삽입되는 보드이다. DIP SW를 통해서 9054보드의 동작 모드 및 Local clk 을 설정할 수 있다. 50핀의 2개의 확장 콘넥터를 통하여 PCI9054의 Local 신호가 외부로 입출력된다.

- (1)번 확장 콘넥터 J3은 보드의 위쪽, J4은 하단의 콘넥터로 PCI 9054 Chip의 신호선이 그대로 출력된다. J3에서, PC+12V은 PCI 슬롯의 +12를 말한다. 보드의 JP1점퍼가 Short되어 있어야 출력된다. EX_CLK은 9054 Local Clk를 넣어주어야 하는 입력단자이다. FPGA보드에서 CLK를 공급 받는다. J4에서, PC-12V은 PCI 슬롯의 -12를 말한다. 보드의 JP2점퍼가 Short되어 있어야 출력된다.나머지 단자들은 PCI 9054 Chip의 신호선이며 기타 전원 VCC(5V), 3.3V, GND등이다.

J3 CON50A-MALE

GND	1	2	PC+12V
/LBIGEND	3	4	/LCCS
/LINT	5	6	LDMPAF
LBREQo	7	8	LBREQi
/LADS	9	10	LHOLDA
EX_CLK	11	12	LDP3
LDP2	13	14	LD1
LD0	15	16	LD5
LD4	17	18	LD9
LD8	19	20	LD13
LD12	21	22	LD17
LD16	23	24	LD21
LD20	25	26	LD25
LD24	27	28	LD29
LD28	29	30	/LBE1
/LBE0	31	32	LA2
LW_/R	33	34	LA6
LA5	35	36	LA10
LA9	37	38	LA14
LA13	39	40	LA18
LA17	41	42	LA22
LA21	43	44	LA26
LA25	45	46	LA30
LA29	47	48	3.3V
VCC	49	50	GND

J4 CON50A-MALE

GND	1	2	PC-12V
LUSERi	3	4	LUSERo
/LRESETO	5	6	/LWAIT
/LBLAST	7	8	/LSERR
LCLK2	9	10	LHOLD
LDP0	11	12	LDP1
/LREADY	13	14	/LBTERM
LD2	15	16	LD3
LD6	17	18	LD7
LD10	19	20	LD11
LD14	21	22	LD15
LD18	23	24	LD19
LD22	25	26	LD23
LD26	27	28	LD27
LD30	29	30	LD31
/LBE2	31	32	/LBE3
LA3	33	34	LA4
LA7	35	36	LA8
LA11	37	38	LA12
LA15	39	40	LA16
LA19	41	42	LA20
LA23	43	44	LA24
LA27	45	46	LA28
LA31	47	48	3.3V
VCC	49	50	GND

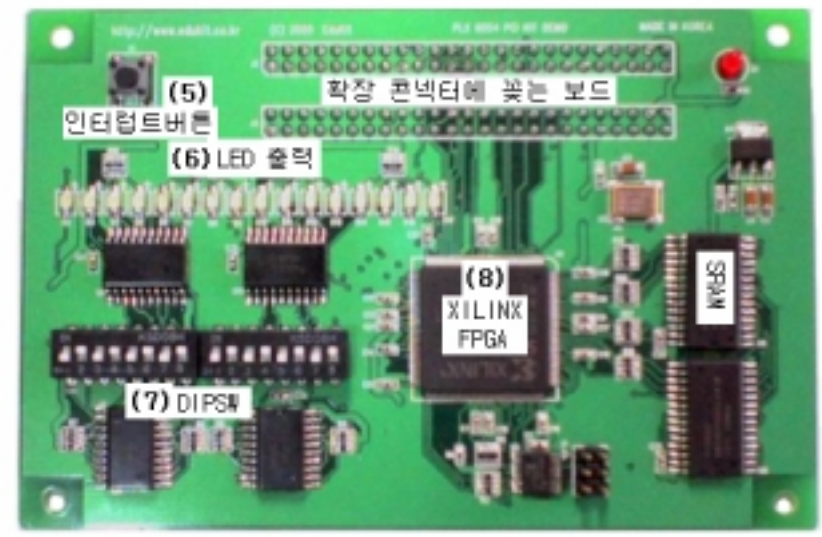
- (2)번 16핀 DIP SW은 C,J 모드를 사용할 때는 1234은 ON, 5678은 OFF시킨다. (DEFAULT)
  M 모드를 사용할 때는 1234은 OFF, 5678은 ON시킨다.

- (4)번 8핀 DIP SW 1,2번은 9054에서 사용하는 LCLK을 선택하는 기능으로

1번 ON, 2번 OFF이면
외부에서 공급되는 Clock을 사용한다. (DEFAULT)
1번 OFF,2번 ON이면
PCI슬롯에서 공급되는 33MHz의 CLOCK를 사용한다.
3번,4번은 PCI 9054 Chip의 MODE0,MODE1 단자로,
3번 ON,4번 ON이면 M 모드,
3번 ON, 4번 OFF이면 J모드 (DEFAULT)
3번 OFF,4번 OFF이면 C모드로 PCI 9054가 동작한다.

- 좌 하단의 JP1,JP2은 PCI슬롯에서 나오는 -12V,+12V를 외부 확장콘넥터 J3,J4로 나오게 할것인가를 설정하는 점퍼이다. OPEN되어 있으면 차단된다.

2-2. Demo FPGA Board



PCI 9054보드에 콘넥터를 통하여 위에 얹는 보드이다.
외부 16비트 SRAM, 인터럽트 테스트 버튼 , LED출력, DIP SW 입력 기능,
FPGA내부의 XILINX BLOCK RAM 입출력기능을 갖는다.
좌하단의 JTAG Port를 통하여 사용자가 마음대로 FPGA 내용을 수정할 수 있다.

3. PLX 9054 드라이버의 설치

3-1 윈도우 98,2000,XP Driver의 설치

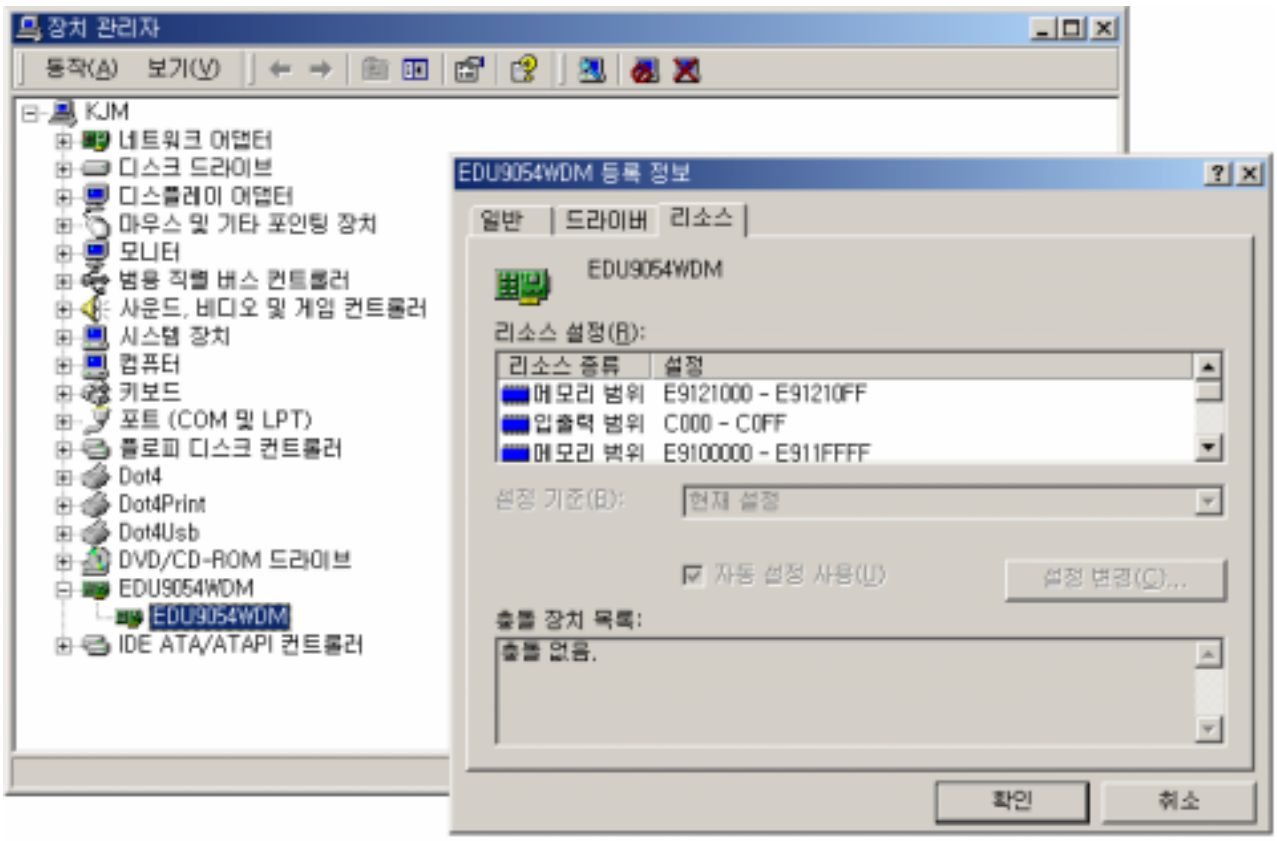
윈도우가 시작되면, 새로운 하드웨어 발견이라고 하면서 드라이버를 요구 한다. 제공되는 CD의 Driver 경로 지정하여 주면 edu9054.sys가 윈도우의 system32Wdrivers 디렉토리에 복사된다. 윈도우를 재 시작하도록 한다.

Windows98이상이라면 윈도우버젼에 상관없이 동일한 드라이버파일을 사용한다.

디바이스 드라이버가 올바로 설치되어 있다면, 제어판 - 시스템 - 하드웨어탭 - 장치관리자를 선택하여 보면

시스템장치에 PCI 9054 WDM 드라이버가 제대로 설치되어 있음을 확인 할 수 있다.

9054 장치 드라이버의 위에서 우측 마우스 버튼을 눌러 등록정보를 보면 같이 9054 보드가 점유한 메모리 및 인터럽트 번호가 보인다. 인터럽트 번호는 다른 하드웨어 장치와 공유 될 수 있다.



참고 : 드라이버 설치가 잘 안될 때

1. 9054 Board의 슬롯에 닿는 부분을 지우개로 깨끗하게 닦는다.
2. VGA card만 제외한 모든 카드를 컴퓨터의 슬롯에서 빼고, PLX Board를 꽂아 본다.
3. 9054 Board를 다른 PCI 슬롯에 꽂아 본다.
4. 장치관리자에서 "EduKit PCI9054 Driver"에 느낌표나 X가 표시되어 있으면 드라이버를 제거하고, "새로 고침" 버튼을 눌러 다시 설치한 후에, 윈도우를 재 시작한다.

4. PLX 9054 KIT의 어드레스 MAP

PCI 9054에 연결된 장치는 2개의 어드레스 영역으로 나누어져 있는데,
Space 0은 내부 XILINX의 BLOCK RAM과 Control
Space 1은 외부 SRAM이 할당되어 있다.

(Block는 램처럼 연속적으로 어드레스가 증가하면서 DMA동작이 이루어지는 장치를 말하고,
Point는 FIFO처럼 한 번지에서만 이루어지는 장치를 말한다.)

4-1 SPACE 0 (PCI 9054 Local address : 0x0)

0x0~0x7ff (DWord, 32bit, Block) : XILINX Internal BLOCK RAM, Read/Write
512 x 32bit 의 메모리이다.

0x800 (Dword, 32bit,Point) : Write : LED (D15-D0)
원하는 LED를 켜고 끌수 있다.
Read : LED (D15-D0), DIP SW (D31-D16)
D15-D0 에는 최근 출력한 LED값이
D31-D16 에는 외부의 DIP SW 상태가 읽힌다.

0x804 (Dword, 32bit,Point) : Write : IE_PBSW (D0) , ACK_PBSW(D1)
PBSW의 인터럽트가 걸리게 하려면 D0에 '1'
PBSW의 인터럽트플래그(FLAG_PBSW)을 0으로 만들려면
ACK_PBSW를 '1'로 Write한다.
Read : IE_PBSW (D0) , FLAG_PBSW(D1), EXT_PBSW(D2)
PBSW의 인터럽트 Enable/Disable상태는 D0
외부의 인터럽트푸시버튼이 눌렸으면 FLAG_PBSW가 '1'
외부 PBSW의 현재상태가 읽힌다. 평소에 '1'
눌리면 '0'이 읽힌다.

4-2 SPACE 1 (PCI 9054 Local address : 0x100000)

0x0 ~ 0xffff (DWORD, 32bit, Block) : External SRAM Read/Write

실제로 외부램은 16비트이지만 PC의 32비트 작업에 16비트 2번 작업이 이루어지게 하였다.

즉, 0x0번지에 0x12345678을 쓰면, 실제로 외부 SRAM에는 0x0 : 0x1234, 0x1 : 0x5678 의 2번의 Write 작업이 이루어지게 된다. 읽는 작업도 동일하다.

실제 Test프로그램에서 Define되어 있는 값은 다음과 같다.

```
// SPACE 0
#define LOC_PHY_INT_RAM_ADDR      0x0

#define LOC_INT_RAM               0x0

#define LOC_LEDDIPSW          0x800
      //  RD     D15 - D0  : output ,   D31 - D16 : input
      //  WR     D15 - D0  : output
#define LOC_INT_FLAG_ACK       0x804
     // RD
      #define EN_PBSW            0x1
      #define FLAG_PBSW          0x2
      #define EXT_PBSW           0x4
     //  WR
     // #define EN_PBSW          0x1
       #define ACK_PBSW          0x2

// SPACE 1
#define LOC_PHY_EXT_RAM_ADDR      0x100000

#define LOC_EXT_RAM               0x0
```

5. PLX 9054 Board의 EEPROM

PLX 9054 보드내의 EEPROM(93CS66B)의 내용에 따라 PCI Configuration의 레지스터 및 Local Register의 내용이 변경된다. IRQ 및 점유하는 어드레스 영역과 갯수, PCI 9054 소자 핀의 용도 설정들을 할 수 있다. 자세한 설정 방법은 PCI 9054의 데이터시트를 참조한다.

5-1 다음은 보드에 설정되어 있는 EEPROM의 내용이다.

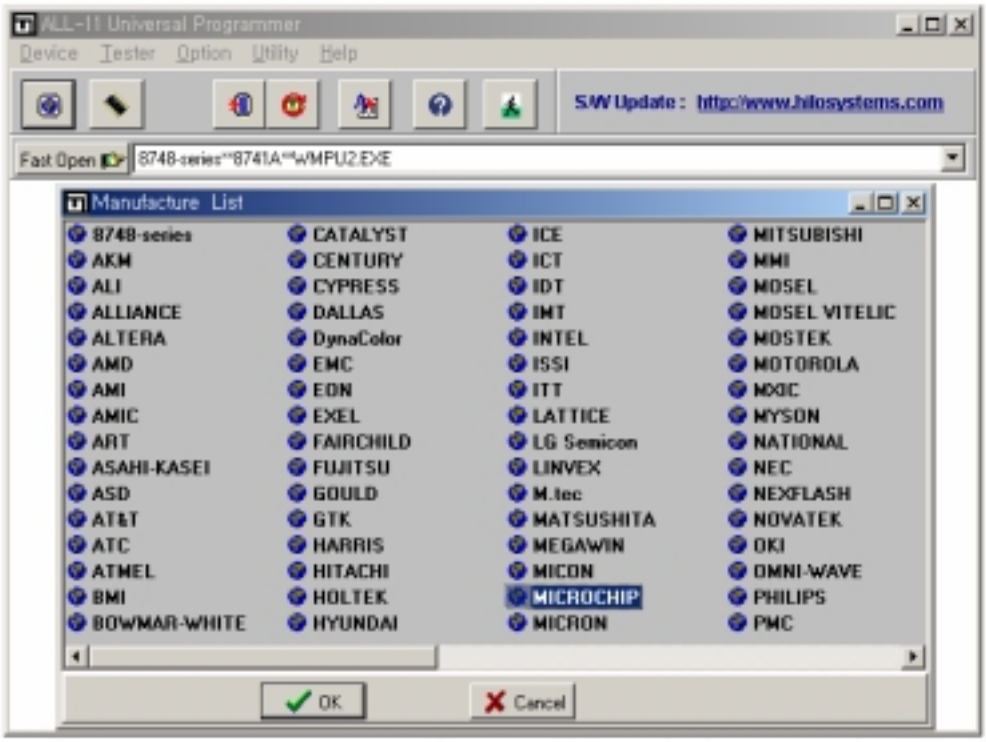
Offset	Register	Value	
0h	PCIIDR	905415cd	
4h	PCICCR,PCIREV	0680000b	
8h	PCIMLR,PCIMGR,	00000100	
	PCIIPR,PCIILR		
Ch	MBOX0	00000000	
10h	MBOX1	00000000	
14h	LASORR	FFF80000	512 Kbyte공간
18h	LASOBA	00000001	
1Ch	MARBR	0823FFFF	
20h	20: BIGEND : 0xff	00305500	
	21: LMISC : 0x01		
	22-3:PROT_AREA : 0x0400		
24h	EROMRR	00000000	
28h	EROMBA	00000010	
2ch	LBRD0	F2000243	32bit
30h	DMRR	00000000	
34h	DMLBAM	00000000	
38h	DMLBAI	00000000	
3ch	DMPBAM	00000000	
40h	DMCFGA	00000000	
44h	44-5 : Sub ID	905410b5	
	46-7 : Verdor ID		
48h	LAS1RR	FFF80000	512 Kbyte 공간
4ch	LAS1BA	00100001	
50h	LBRD1	00000243	32bit
54h	54-5 : Reserved	00004c06	
	56-7 : HS_NEXT/HS_CNTL		

인터럽트는 #INTA를 사용한다. PCI Configuration Register의 Base 어드레스를 메모리와 I/O 모두 잡히게 하였다. PCI 9054 소자의 Space0은 512K byte,32 bit,
Space 1은 512 Kbyte , 32bit 가 할당 되도록 하였다.

5-2 EEPROM의 내용 수정

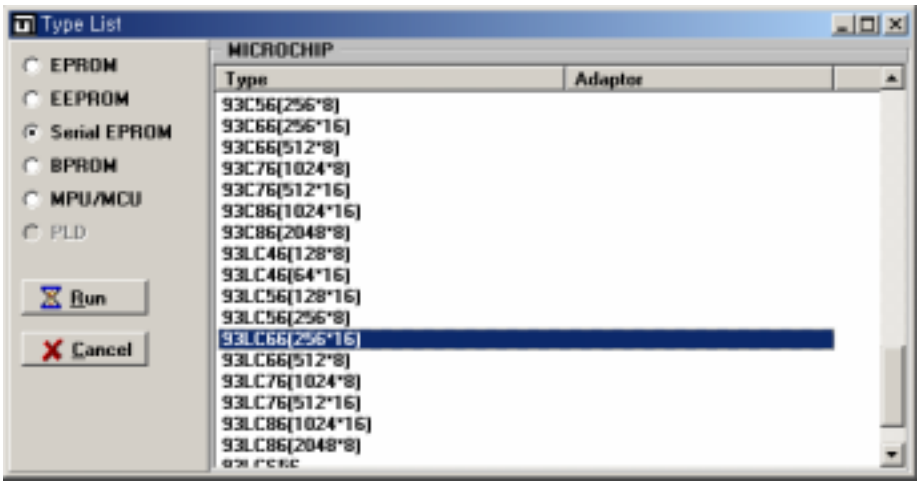
기존 보드에 꽂혀 있는 EEPROM을 사용하거나 EEPROM를 구해야 한다.
Sequential Read기능이 있는 93CS56B 또는 93CS66B 이 사용가능하나, 이것과 동일한 기능이 있는 MicroChip사의 93LC66B또는 93LC56B의 eeprom이 가능하다. 93LC66B, 93LC56B 처럼 끝에 B type인 것을 사용한다. B type은 16비트 데이터 폭을 가지며 8비트 데이터 폭을 갖는 A type은 사용할 수 없다.

All11 롬라이터의 경우, Manufacture List에서 MICROCHIP를 선택한다.



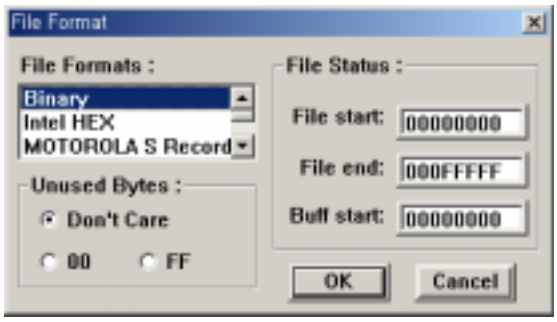
93LC56의 경우 128x16bit 모드의 것을 선택하고

93LC66의 경우 256x16bit 모드의 것을 선택한다.



그리고 File을 Open하여 CD의 HardwareWeepromW9054LC66B.bin 파일을

File Format를 Binary로 로드하여 Write하면 된다.



한편, 데이터를 롬라이터 프로그램등을 이용하여 버퍼보기로 하여 내용을 읽어보면 다음과 같은 순서를 갖는다.

```
000000  54 90 CD 15 80 06 0B 00  00 00 00 01 00 00 00 00
000010  00 00 00 00 F8 FF 00 00  00 00 01 00 23 08 FF FF
000020  30 00 00 55 00 00 00 00  00 00 10 00 00 F2 43 02
000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000040  00 00 00 00 54 90 B5 10  F8 FF 00 00 10 00 01 00
000050  00 00 43 02 00 00 06 4C  00 00 00 00 00 00 00 00
000060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
000090  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
0000a0  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
0000b0  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
0000c0  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
0000d0  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
0000e0  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
0000f0  FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF
```

순서가 틀려지므로 주의하도록 한다.

원래의 데이터가 00H : 905415cd 라면 (1234 순서)

00H : 54 90 CD 15이 된다. (2143 순서)

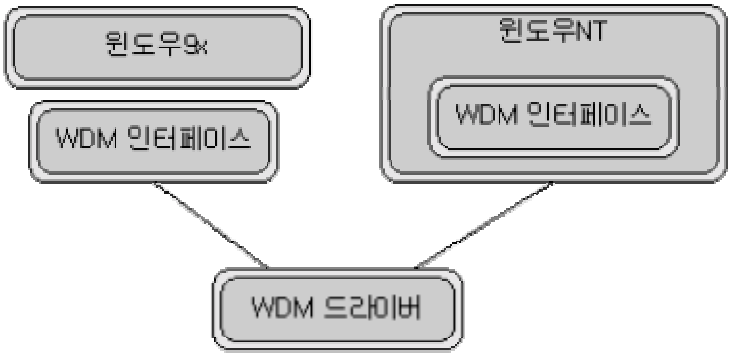
내용을 수정할 일이 있으면 메뉴의 Editor를 불러서 원하는 어드레스에 있는 값을 순서를 고려하여 수정한다.

6. WDM 디바이스 드라이버 설명

참고 : CD의 Driver 경로에 PLX 9054 보드의 윈도우 WDM 드라이버의
컴파일이 완료된 파일들이 있다. Win98이상의 윈도우에 동일하게 호환된다.

6-1 WDM 드라이버

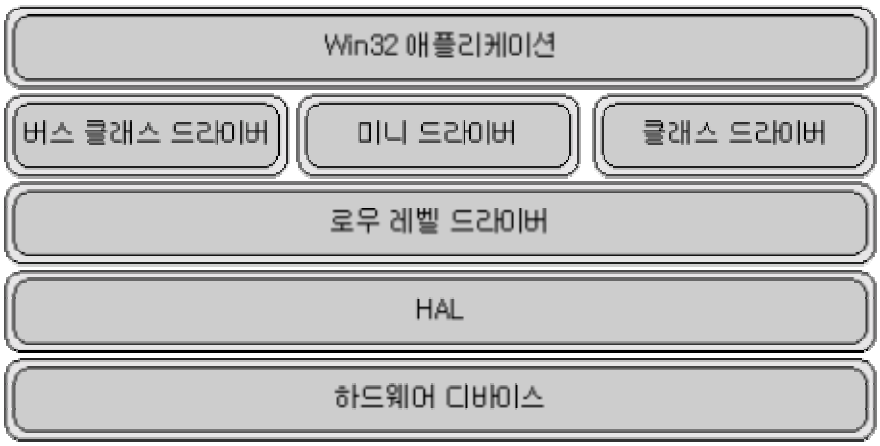
WDM은 윈도우 OS버전간의 차이에 고민할 필요 없이 각각의 OS에 사용할 수 있는 윈도우 2000, XP , 윈도우 98, 윈도우 Me 그리고 미래의 윈도우 OS 에 적용되는 편리한 디바이스 드라이버 모델이다. 플러그 앤 플레이(PNP)를 지원 하기 때문에 PC의 BIOS가 자동으로 활당한 자원을 넘겨받아 사용하므로 개발자가 일일이 하드웨어 자원을 찾아 할당하는 번거로운 과정이 필요 없다. WDM 드라이버는 기존 NT 드라이버를 기반으로 하는 실행 파일(sys 형태)로 작성되며 대부분의 소스 코드가 윈도우 NT와 비슷하므로 먼저 NT 드라이버 작성에 필요한 사항을 알고 있는 경우에는 유리하다.



<그림 > WDM 드라이버 모델

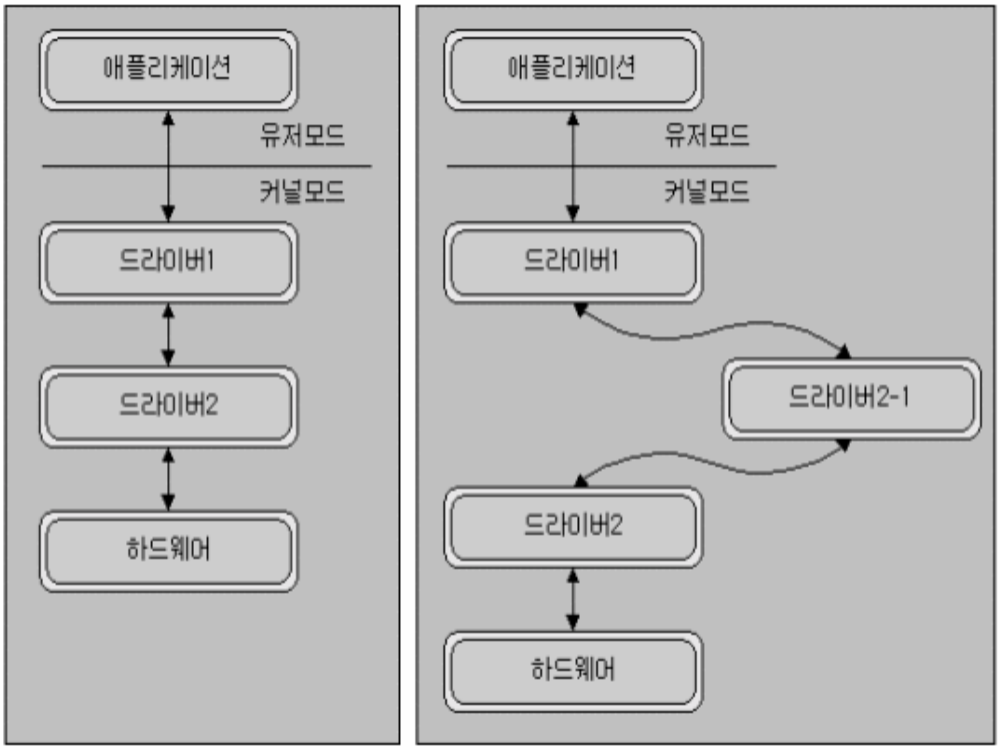
6-2 계층형 드라이버와 필터 드라이버

윈도우에서는 Win32 기반의 사용자 모드와 커널 모드로 구분되어 소프트웨어가 실행된다. 커널모드의 드라이버는 HAL 이라는 인터페이스를 통해 실제 디바이스와 통신한다.



<그림 > 드라이버 계층도

계층형 구조를 가진 윈도우는 특정 디바이스에 대한 드라이버를 하드웨어에 근접한 순으로 여러 층에 걸쳐 구성하고 상위 드라이버가 애플리케이션에서 입출력 요청을 받아 차례대로 하위 드라이버에 처리 요청을 전달한다. 필터 드라이버는 계층형 드라이버 중간에 드라이버를 삽입시켜 이 드라이버가 입출력 요구를 처리하도록 만든 구조이다. WDM 드라이버가 필터 드라이버에 해당한다.



6-3. WDM 드라이버의 컴파일

소스는 동일하나 각 OS환경에서 컴파일 하여야 한다. 방법은 다음과 같다.

1) Windows 98

Windows98에서는 컴파일되지 않으며 다른 OS에서 컴파일된 것을 그대로 사용한다.

2) Windows ME

VC++6.0, Win 2000 DDK를 설치한후에 2000 ddk의 Free Build

도스창으로 간후에 wdm의 경로에서 run.bat를 실행한다.

3) Windows 2000

VC++6.0, Win 2000 DDK를 설치한후에 2000 ddk의 Free Build

도스창으로 간후에 wdm 경로에서 run.bat를 실행한다.

4) Windows XP

VC++6.0, Win XP DDK를 설치한후에 XP ddk의 Free Build 도스창으로

간후에 wdm 경로에서 run.bat를 실행한다.

7. PLX 9054 보드의 응용프로그램 설명

제공되는 CD에 테스트 응용프로그램 (Edu9054test.exe)이 있다.
Edu9054testexe Dos의 Console 화면에서 작동하는 것으로, 이 프로그램의 소스는 CD의 Sotware 경로에 압축되어 있다.

7-1 MailBoxRead,Write Test : MailBox은 9054 Chip속의 내부 레지스터이다.

```
=================  PLX9054 PCI KIT Test V3.0 ========================
 0. Help           8. Target Write       J. 9054 Reg Dump
 1. Write Mailbox  9. Target Read        P. Push Switch
 2. Read Mailbox   A. EEPROM Write       Q. Int Ram DMA Read
 4. Wr 9054 Reg    B. EEPROM Read        R. Int Ram DMA Write
 5. Rd 9054 Reg    C. Set DmaTimeout     S. Ext Ram DMA Read
 6. DMA Write      D. LED Demo           T. Ext Ram Dma Write
 7. DMA Read       F. FPGA Reg           X. Exit Demo
 Enter Selection (0 :help) ==> 1

Choose an outgoing register for writing (0-7) :0

Data to write: 0x12345678

Mailbox data written successfully

 Enter Selection (0 :help) ==> 2

Choose a mailbox register for reading (0-7): 0

0x12345678 is in the mailbox
```

4

MailBox는 0-7사이가 존재하는데 0번에 데이터 0x12345678를 쓴 후에 다시 읽어보는 예제이다.

7-2 PCI 9054 Register Read,Write Test

```
=================  PLX9054 PCI KIT Test V3.0 ========================
 0. Help           8. Target Write       J. 9054 Reg Dump
 1. Write Mailbox  9. Target Read        P. Push Switch
 2. Read Mailbox   A. EEPROM Write       Q. Int Ram DMA Read
 4. Wr 9054 Reg    B. EEPROM Read        R. Int Ram DMA Write
 5. Rd 9054 Reg    C. Set DmaTimeout     S. Ext Ram DMA Read
 6. DMA Write      D. LED Demo           T. Ext Ram Dma Write
 7. DMA Read       F. FPGA Reg           X. Exit Demo
 Enter Selection (0 :help) ==> 5

Enter Reg9054 Offset (Byte Boundary) to Read: 0xb4

Reg[0xb4] = 0x0 read from Reg9054

 Enter Selection (0 :help) ==> 4

Enter Reg9054 Offset (Byte Boundary) to Write: 0xb4

Enter Data to Write to Reg9054: 0x12345678

Reg[0xb4] = 0x12345678  written to Reg9054

 Enter Selection (0 :help) ==> 5

Enter Reg9054 Offset (Byte Boundary) to Read: 0xb4

Reg[0xb4] = 0x12345678 read from Reg9054

 Enter Selection (0 :help) ==> 4

Enter Reg9054 Offset (Byte Boundary) to Write: 0xb4

Enter Data to Write to Reg9054: 0x0

Reg[0xb4] = 0x0  written to Reg9054
```

PCI 9054 Chip의 내부 레지스터 0xb4를 내용을 읽어보니, 0x0 값이 들어있고
0xb4 레지스터에 0x12345678을 쓴 후에 다시 읽어보니 0x12345678 이 제대로 써져 있는 것을 확인 할 수 있었다.
그리고 Test를 마쳤으므로, 다시 원래의 값인 0x0으로 되돌려 놓았다.

7-3. DMA Write Test

PCI 9054 Chip의 DMA 제어 장치를 이용하여 PC의 CPU 작동없이 PC의 메모리의 내용을 PCI 9054에 연결된 SRAM이나 FIFO에 고속으로 데이터를 쓴다.

Add-on/Local Address에는 PCI9054에 연결되어 있는 장치의 실제 PCI9054 Local Address를 입력하여 주어야 한다.

```
================  PLX9054 PCI KIT Test V3.0 ========================
 0. Help           8. Target Write       J. 9054 Reg Dump
 1. Write Mailbox  9. Target Read        P. Push Switch
 2. Read Mailbox   A. EEPROM Write       Q. Int Ram DMA Read
 4. Wr 9054 Reg    B. EEPROM Read        R. Int Ram DMA Write
 5. Rd 9054 Reg    C. Set DmaTimeout     S. Ext Ram DMA Read
 6. DMA Write      D. LED Demo           T. Ext Ram Dma Write
 7. DMA Read       F. FPGA Reg           X. Exit Demo
 Enter Selection (0 :help) ==> 6

Enter Add-on/Local Address (Byte Boundary): 0x0
Enter 1 for block, 2 for point: 1
Enter Data Width (DWord:0 , Word:1 ,Byte:2): 0

Enter the Count (1 - 128000): 128

Dma Write in Progress offset=0x0 datawidth=0x0 count=0x80
Successful Transfer, Data Written :
0x25974304 0x25974305 0x25974306 0x25974307 0x25974308
0x25974309 0x2597430a 0x2597430b 0x2597430c 0x2597430d
0x2597430e 0x2597430f 0x25974310 0x25974311 0x25974312
0x25974313 0x25974314 0x25974315 0x25974316 0x25974317
0x25974318 0x25974319 0x2597431a 0x2597431b 0x2597431c
0x2597431d 0x2597431e 0x2597431f 0x25974320 0x25974321
0x25974322 0x25974323 0x25974324 0x25974325 0x25974326
0x25974327 0x25974328 0x25974329 0x2597432a 0x2597432b
0x2597432c 0x2597432d 0x2597432e 0x2597432f 0x25974330
0x25974331 0x25974332 0x25974333 0x25974334 0x25974335
0x25974336 0x25974337 0x25974338 0x25974339 0x2597433a
0x2597433b 0x2597433c 0x2597433d 0x2597433e 0x2597433f
0x25974340 0x25974341 0x25974342 0x25974343 0x25974344
0x25974345 0x25974346 0x25974347 0x25974348 0x25974349
0x2597434a 0x2597434b 0x2597434c 0x2597434d 0x2597434e
0x2597434f 0x25974350 0x25974351 0x25974352 0x25974353
0x25974354 0x25974355 0x25974356 0x25974357 0x25974358
0x25974359 0x2597435a 0x2597435b 0x2597435c 0x2597435d
0x2597435e 0x2597435f 0x25974360 0x25974361 0x25974362
0x25974363 0x25974364 0x25974365 0x25974366 0x25974367
0x25974368 0x25974369 0x2597436a 0x2597436b 0x2597436c
0x2597436d 0x2597436e 0x2597436f 0x25974370 0x25974371
0x25974372 0x25974373 0x25974374 0x25974375 0x25974376
0x25974377 0x25974378 0x25974379 0x2597437a 0x2597437b
0x2597437c 0x2597437d 0x2597437e 0x2597437f 0x25974380
0x25974381 0x25974382 0x25974383
```

PC의 메모리에 있는 버퍼의 내용을 Block방식, Dword 데이터 폭으로 128개의 데이터를 XILINX BLOCK RAM의 0번지부터 128 DWord 개수의 DMA Write를 수행한다. 0x25974304~ 에서 시작되는 값은, DMA Write동작이 이루어지면 +1씩 계속 증가하는 임의의 값이다. 현재, RAM의 0x00~0x03 번지에는 0x25974305가 Write되어져 있을 것이다.

만일에 SPACE1 에 있는 외부 SRAM의 경우라면,
위의 예제화면에서 Enter Add-on/Local Addresss를 0x100000 를 입력하여야 한다.

7-4 DMA Read Test

```
=================  PLX9054 PCI KIT Test V3.0 ========================
 0. Help           8. Target Write       J. 9054 Reg Dump
 1. Write Mailbox  9. Target Read        P. Push Switch
 2. Read Mailbox   A. EEPROM Write       Q. Int Ram DMA Read
 4. Wr 9054 Reg    B. EEPROM Read        R. Int Ram DMA Write
 5. Rd 9054 Reg    C. Set DmaTimeout     S. Ext Ram DMA Read
 6. DMA Write      D. LED Demo           T. Ext Ram Dma Write
 7. DMA Read       F. FPGA Reg           X. Exit Demo
 Enter Selection (0 :help) ==> 7

Enter Add-on/Local Address (Byte Boundary): 0x0
Enter 1 for block, 2 for point: 1
Enter Data Width (DWord:0 , Word:1 ,Byte:2): 0

Enter the Count (1 - 128000): 128

Dma Read in Progress offset=0 datawidth=0 count=80
Successful Transfer, Data Read:
0x25974304 0x25974305 0x25974306 0x25974307 0x25974308
0x25974309 0x2597430a 0x2597430b 0x2597430c 0x2597430d
0x2597430e 0x2597430f 0x25974310 0x25974311 0x25974312
0x25974313 0x25974314 0x25974315 0x25974316 0x25974317
0x25974318 0x25974319 0x2597431a 0x2597431b 0x2597431c
0x2597431d 0x2597431e 0x2597431f 0x25974320 0x25974321
0x25974322 0x25974323 0x25974324 0x25974325 0x25974326
0x25974327 0x25974328 0x25974329 0x2597432a 0x2597432b
0x2597432c 0x2597432d 0x2597432e 0x2597432f 0x25974330
0x25974331 0x25974332 0x25974333 0x25974334 0x25974335
0x25974336 0x25974337 0x25974338 0x25974339 0x2597433a
0x2597433b 0x2597433c 0x2597433d 0x2597433e 0x2597433f
0x25974340 0x25974341 0x25974342 0x25974343 0x25974344
0x25974345 0x25974346 0x25974347 0x25974348 0x25974349
0x2597434a 0x2597434b 0x2597434c 0x2597434d 0x2597434e
0x2597434f 0x25974350 0x25974351 0x25974352 0x25974353
0x25974354 0x25974355 0x25974356 0x25974357 0x25974358
0x25974359 0x2597435a 0x2597435b 0x2597435c 0x2597435d
0x2597435e 0x2597435f 0x25974360 0x25974361 0x25974362
0x25974363 0x25974364 0x25974365 0x25974366 0x25974367
0x25974368 0x25974369 0x2597436a 0x2597436b 0x2597436c
0x2597436d 0x2597436e 0x2597436f 0x25974370 0x25974371
0x25974372 0x25974373 0x25974374 0x25974375 0x25974376
0x25974377 0x25974378 0x25974379 0x2597437a 0x2597437b
0x2597437c 0x2597437d 0x2597437e 0x2597437f 0x25974380
0x25974381 0x25974382 0x25974383
```

XILINX BLOCK RAM의 0x0 번지부터의 내용을 PC의 메모리에 DMA동작으로 읽어 왔다.

아까 DMA Write한 값이 제대로 써져 있다는 것을 확인할 수 있다.

7-5 Target Read/Write Test

Target은 PC의 CPU가 직접 데이터를 PCI 9054의 장치를 읽고 쓰는 것으로 CPU자원을 많이 차지하므로 소량의 데이터를 읽을 때만 사용한다.

```
=================  PLX9054 PCI KIT Test V3.0 ========================
 0. Help           8. Target Write       J. 9054 Reg Dump
 1. Write Mailbox  9. Target Read        P. Push Switch
 2. Read Mailbox   A. EEPROM Write       Q. Int Ram DMA Read
 4. Wr 9054 Reg    B. EEPROM Read        R. Int Ram DMA Write
 5. Rd 9054 Reg    C. Set DmaTimeout     S. Ext Ram DMA Read
 6. DMA Write      D. LED Demo           T. Ext Ram Dma Write
 7. DMA Read       F. FPGA Reg           X. Exit Demo
 Enter Selection (0 :help) ==> 8

Enter write region (Space 0 or 1): 0

Enter Add-on/Local Address (Byte Boundary): 0x0
Enter 1 for block, 2 for point: 1

Enter Data Width (DWord:0 , Word:1 ,Byte:2): 0

Enter Count (1 - 128000): 2

Enter value 0 for writing: 0x12345678

Enter value 1 for writing: 0x56789abc

 Enter Selection (0 :help) ==> 9

Enter read region (Space 0 or 1): 0

Enter Add-on/Local Address (Byte Boundary): 0x0
Enter 1 for block, 2 for point: 1

Enter Data Width (DWord:0 , Word:1 ,Byte:2): 0

Enter Count (1 - 128000): 2

ReadTarget in Progress offset=0 datawidth=0 count=2
0x12345678 0x56789abc
```

Space 0의 XLINX RAM의 0번지부터 2 DWord의 데이터를 써보고, 다시 읽어본다.

7-6 EEPROM Test

EEPROM의 9054 보드에 장착되어 있는 93LC66B을 말하는 것으로, PCI 9054 Chip에 전원이 들어올 때 PCI Configuration, Local Address등을 초기화 시킨다. 사용자가 PCI 9054의 여러가지 Option기능들을 이 EEPROM을 통하여 설정할 수가 있다. 대개 EEPROM은 ROM Writer로 수정하지만
이 테스트 프로그램을 통하여 바꿀 수 있다. 그런데 값의 정확한 의미를 알고 쓰지 않으면 컴퓨터가 다시 재 부팅 되었을 경우 잘못된 값 때문에 원하는 동작이 일어나지 않는다.

```
=================  PLX9054 PCI KIT Test V3.0 ========================
 0. Help           8. Target Write       J. 9054 Reg Dump
 1. Write Mailbox  9. Target Read        P. Push Switch
 2. Read Mailbox   A. EEPROM Write       Q. Int Ram DMA Read
 4. Wr 9054 Reg    B. EEPROM Read        R. Int Ram DMA Write
 5. Rd 9054 Reg    C. Set DmaTimeout     S. Ext Ram DMA Read
 6. DMA Write      D. LED Demo           T. Ext Ram Dma Write
 7. DMA Read       F. FPGA Reg           X. Exit Demo
 Enter Selection (0 :help) ==> b

Choose Read offset (Byte Boundary): 0x0

0x905415CD is read from EEPROM 0
 Enter Selection (0 :help) ==> b

Choose Read offset (Byte Boundary): 0x30

0x0 is read from EEPROM 30
 Enter Selection (0 :help) ==> a

Choose write offset (Byte Boundary): 0x30

Data to write: 0x1234

EEPROM data written successfully

 Enter Selection (0 :help) ==> b

Choose Read offset (Byte Boundary): 0x30

0x1234 is read from EEPROM 30
 Enter Selection (0 :help) ==> a

Choose write offset (Byte Boundary): 0x30

Data to write: 0x0

EEPROM data written successfully
```

0번지의 값을 읽어서 PCI ID값 0x905415cd을 읽었다. 0x30의 값을 읽어보니 0x0 이었고 여기게 0x1234를 쓰고 읽어보았더니 제대로 써있었다. 다시 원래대로의 값으로 되돌려 놓았다. (반드시 원래의 값으로 되돌려 놓아야 한다.)

7-7 DMA Timeout, LED Demo, FPGA Reg

DMA Timeout값은 DMA/Read Write 동작 시에 원하는 작업이 제대로 이루어지지 않았을 경우, 그 작업을 취소시키는 시간을 정하는 것으로 5000ms을 써넣으면 5초 이내에 명령 내린 DMA동작이 이루어 지지 않으면 명령을 취소시킨다.
Default값이 5000ms 이다. LED Demo은 LED가 좌에서 우로 스크롤된다. FPGA Register 읽기는 PCI 9054의 Space 0의 Control Register 값을 읽는 기능이다.

```
=================  PLX9054 PCI KIT Test V3.0 ========================
 0. Help           8. Target Write       J. 9054 Reg Dump
 1. Write Mailbox  9. Target Read        P. Push Switch
 2. Read Mailbox   A. EEPROM Write       Q. Int Ram DMA Read
 4. Wr 9054 Reg    B. EEPROM Read        R. Int Ram DMA Write
 5. Rd 9054 Reg    C. Set DmaTimeout     S. Ext Ram DMA Read
 6. DMA Write      D. LED Demo           T. Ext Ram Dma Write
 7. DMA Read       F. FPGA Reg           X. Exit Demo
 Enter Selection (0 :help) ==> c

Enter new timeout (ms...0 = No timeout): 5000

 Enter Selection (0 :help) ==> d

 Enter Selection (0 :help) ==> f
[I/O]  LED=0xffff  DIPSW=0xaaaa
[INT]  EN_PBSW = 0  FLAG_PBSW = 0 PBSW = 1
```

현재 PBSW의 인터럽트가 켜있지 않고 (EN_PBSW = 0)
푸시버튼이 눌린 적이 없으며(FLGA_PBSW = 0)
외부 푸시버튼도 눌려있는 상태가 아니다. (PBSW = 1)

7-8 PCI 9054 Register Dump

```
=================  PLX9054 PCI KIT Test V3.0 ========================
 0. Help           8. Target Write       J. 9054 Reg Dump
 1. Write Mailbox  9. Target Read        P. Push Switch
 2. Read Mailbox   A. EEPROM Write       Q. Int Ram DMA Read
 4. Wr 9054 Reg    B. EEPROM Read        R. Int Ram DMA Write
 5. Rd 9054 Reg    C. Set DmaTimeout     S. Ext Ram DMA Read
 6. DMA Write      D. LED Demo           T. Ext Ram Dma Write
 7. DMA Read       F. FPGA Reg           X. Exit Demo
 Enter Selection (0 :help) ==> j

 < 9054 register dump >
[0x0000] = 0xFFF80000  [0x0004] = 0x00000001  [0x0008] = 0x0823FFFF
[0x000C] = 0x00305500  [0x0010] = 0x00000000  [0x0014] = 0x00000010
[0x0018] = 0xF2000243  [0x001C] = 0x00000000  [0x0020] = 0x00000000
[0x0024] = 0x00000000  [0x0028] = 0x00000000  [0x002C] = 0x00000000
[0x0030] = 0x00000000  [0x0034] = 0x00000008  [0x0038] = 0x00000000
[0x003C] = 0x00000000  [0x0040] = 0x12345678  [0x0044] = 0x00000000
[0x0048] = 0x00000000  [0x004C] = 0x00000000  [0x0050] = 0x00000000
[0x0054] = 0x00000000  [0x0058] = 0x00000000  [0x005C] = 0x00000000
[0x0060] = 0x00000000  [0x0064] = 0x00000000  [0x0068] = 0x0F0C0180
[0x006C] = 0x100F767E  [0x0070] = 0x905410B5  [0x0074] = 0x0000000C
[0x0078] = 0x12345678  [0x007C] = 0x00000000  [0x0080] = 0x00020643
[0x0084] = 0x5B271000  [0x0088] = 0x000001F4  [0x008C] = 0x0000000C
[0x0090] = 0x0000000B  [0x0094] = 0x00020643  [0x0098] = 0x5B271000
[0x009C] = 0x000001F4  [0x00A0] = 0x0000000C  [0x00A4] = 0x00000003
[0x00A8] = 0x00001010  [0x00AC] = 0x0823FFFF  [0x00B0] = 0x00000000
[0x00B4] = 0x00000000  [0x00B8] = 0x00000000  [0x00BC] = 0x00000000
[0x00C0] = 0x00000002  [0x00C4] = 0x00000000  [0x00C8] = 0x00000000
[0x00CC] = 0x00000000  [0x00D0] = 0x00000000  [0x00D4] = 0x00000000
[0x00D8] = 0x00000000  [0x00DC] = 0x00000000  [0x00E0] = 0x00000000
[0x00E4] = 0x00000000  [0x00E8] = 0x00000050  [0x00EC] = 0x00000000
[0x00F0] = 0xFFF80000  [0x00F4] = 0x00100001  [0x00F8] = 0x00000243
[0x00FC] = 0x00000000
```

CI 9054 내부의 모든 어드레스 값을 읽어서 표시한다.

PCI 9054의 내부 어드레스는 0x0~0xff 사이의 값을 갖는다.

자세한 사용방법은 PCI 9054 DataSheet를 참조한다.

7-9 Push Button 인터럽트 테스트

P' 메뉴를 누르면 보드에 있는 인터럽트 버튼을 누를때마다 PC인터럽트가 발생하여 test프로그램에 알려준다. 다시 'P'를 누르면 이 테스트 모드가 해제된다. 버튼을 누를 때 많은 횟수가 카운트 되는 것은 Push Button의 채터링 현상이다.

```
=================  PLX9054 PCI KIT Test V3.0 ========================
 0. Help           8. Target Write       J. 9054 Reg Dump
 1. Write Mailbox  9. Target Read        P. Push Switch
 2. Read Mailbox   A. EEPROM Write       Q. Int Ram DMA Read
 4. Wr 9054 Reg    B. EEPROM Read        R. Int Ram DMA Write
 5. Rd 9054 Reg    C. Set DmaTimeout     S. Ext Ram DMA Read
 6. DMA Write      D. LED Demo           T. Ext Ram Dma Write
 7. DMA Read       F. FPGA Reg           X. Exit Demo
 Enter Selection (0 :help) ==> p
Interrupt Start Press push switch

 Enter Selection (0 :help) ==>
 Enter Selection (0 :help) ==> [Sw=  01 ]
 Enter Selection (0 :help) ==> [Sw=  02 ]
 Enter Selection (0 :help) ==> [Sw=  03 ]
              ...생략...
 Enter Selection (0 :help) ==> [Sw=1552 ]
 Enter Selection (0 :help) ==> [Sw=1553 ]
 Enter Selection (0 :help) ==> [Sw=1554 ]
 Enter Selection (0 :help) ==> [Sw=1555 ]
 Enter Selection (0 :help) ==> p
PushSwitch Interrupt End
```

버튼을 한 번을 누르자, 디바이스 드라이버내의 인터럽트 서비스 루틴이
1555번 인식을 하여 수행되었다.
컴퓨터의 성능이 좋을수록 이 값은 더 커진다.

7-10 XILINX Block Ram의 DMA Read/Write Test

간편하게 XILINX RAM의 DMA Read/Write 동작을 검증하여 볼 수 있다.

'r' 를 눌러서 위의 값들을 Write하고 나서 'q' 를 누르면 동일 한 값이 읽힌다.

```
=================  PLX9054 PCI KIT Test V3.0 ========================
 0. Help           8. Target Write       J. 9054 Reg Dump
 1. Write Mailbox  9. Target Read        P. Push Switch
 2. Read Mailbox   A. EEPROM Write       Q. Int Ram DMA Read
 4. Wr 9054 Reg    B. EEPROM Read        R. Int Ram DMA Write
 5. Rd 9054 Reg    C. Set DmaTimeout     S. Ext Ram DMA Read
 6. DMA Write      D. LED Demo           T. Ext Ram Dma Write
 7. DMA Read       F. FPGA Reg           X. Exit Demo
 Enter Selection (0 :help) ==> r

Internal Ram Dma Write in Progress offset= 0x0 count= 0x80
Successful Transfer, Data Written:
0x25974384 0x25974385 0x25974386 0x25974387 0x25974388 0x25974389 0x2597438a
0x2597438b 0x2597438c 0x2597438d 0x2597438e 0x2597438f 0x25974390 0x25974391
0x25974392 0x25974393 0x25974394 0x25974395 0x25974396 0x25974397 0x25974398
0x25974399 0x2597439a 0x2597439b 0x2597439c 0x2597439d 0x2597439e 0x2597439f
0x259743a0 0x259743a1 0x259743a2 0x259743a3 0x259743a4 0x259743a5 0x259743a6
0x259743a7 0x259743a8 0x259743a9 0x259743aa 0x259743ab 0x259743ac 0x259743ad
0x259743ae 0x259743af 0x259743b0 0x259743b1 0x259743b2 0x259743b3 0x259743b4
0x259743b5 0x259743b6 0x259743b7 0x259743b8 0x259743b9 0x259743ba 0x259743bb
0x259743bc 0x259743bd 0x259743be 0x259743bf 0x259743c0 0x259743c1 0x259743c2
0x259743c3 0x259743c4 0x259743c5 0x259743c6 0x259743c7 0x259743c8 0x259743c9
0x259743ca 0x259743cb 0x259743cc 0x259743cd 0x259743ce 0x259743cf 0x259743d0
0x259743d1 0x259743d2 0x259743d3 0x259743d4 0x259743d5 0x259743d6 0x259743d7
0x259743d8 0x259743d9 0x259743da 0x259743db 0x259743dc 0x259743dd 0x259743de
0x259743df 0x259743e0 0x259743e1 0x259743e2 0x259743e3 0x259743e4 0x259743e5
0x259743e6 0x259743e7 0x259743e8 0x259743e9 0x259743ea 0x259743eb 0x259743ec
0x259743ed 0x259743ee 0x259743ef 0x259743f0 0x259743f1 0x259743f2 0x259743f3
0x259743f4 0x259743f5 0x259743f6 0x259743f7 0x259743f8 0x259743f9 0x259743fa
0x259743fb 0x259743fc 0x259743fd 0x259743fe 0x259743ff 0x25974400 0x25974401
0x25974402 0x25974403
```

7-11 SRAM의 DMA Read/Write Test

간편하게 SRAM의 DMA를 통한 Read/Write 동작을 검증하여 볼 수 있다.

't' 를 눌러서 위의 값들을 Write하고 나서 's' 를 누르면 동일 한 값이 읽힌다.

```
=================  PLX9054 PCI KIT Test V3.0 ========================
 0. Help           8. Target Write       J. 9054 Reg Dump
 1. Write Mailbox  9. Target Read        P. Push Switch
 2. Read Mailbox   A. EEPROM Write       Q. Int Ram DMA Read
 4. Wr 9054 Reg    B. EEPROM Read        R. Int Ram DMA Write
 5. Rd 9054 Reg    C. Set DmaTimeout     S. Ext Ram DMA Read
 6. DMA Write      D. LED Demo           T. Ext Ram Dma Write
 7. DMA Read       F. FPGA Reg           X. Exit Demo
 Enter Selection (0 :help) ==> t

SRam Dma Write in Progress offset= 0x100000 count= 0x80
Successful Transfer, Data Written:
0x25974404 0x25974405 0x25974406 0x25974407 0x25974408 0x25974409 0x2597440a
0x2597440b 0x2597440c 0x2597440d 0x2597440e 0x2597440f 0x25974410 0x25974411
0x25974412 0x25974413 0x25974414 0x25974415 0x25974416 0x25974417 0x25974418
0x25974419 0x2597441a 0x2597441b 0x2597441c 0x2597441d 0x2597441e 0x2597441f
0x25974420 0x25974421 0x25974422 0x25974423 0x25974424 0x25974425 0x25974426
0x25974427 0x25974428 0x25974429 0x2597442a 0x2597442b 0x2597442c 0x2597442d
0x2597442e 0x2597442f 0x25974430 0x25974431 0x25974432 0x25974433 0x25974434
0x25974435 0x25974436 0x25974437 0x25974438 0x25974439 0x2597443a 0x2597443b
0x2597443c 0x2597443d 0x2597443e 0x2597443f 0x25974440 0x25974441 0x25974442
0x25974443 0x25974444 0x25974445 0x25974446 0x25974447 0x25974448 0x25974449
0x2597444a 0x2597444b 0x2597444c 0x2597444d 0x2597444e 0x2597444f 0x25974450
0x25974451 0x25974452 0x25974453 0x25974454 0x25974455 0x25974456 0x25974457
0x25974458 0x25974459 0x2597445a 0x2597445b 0x2597445c 0x2597445d 0x2597445e
0x2597445f 0x25974460 0x25974461 0x25974462 0x25974463 0x25974464 0x25974465
0x25974466 0x25974467 0x25974468 0x25974469 0x2597446a 0x2597446b 0x2597446c
0x2597446d 0x2597446e 0x2597446f 0x25974470 0x25974471 0x25974472 0x25974473
0x25974474 0x25974475 0x25974476 0x25974477 0x25974478 0x25974479 0x2597447a
0x2597447b 0x2597447c 0x2597447d 0x2597447e 0x2597447f 0x25974480 0x25974481
0x25974482 0x25974483
```

8. PLX 9054 Board 회로도