

Lab 5

MPEG-2 System Integration and Testing Evaluation

Professor Ching-Lung Su
E-mail: kevinsu@yuntech.edu.tw
Http://soc.eecs.yuntech.edu.tw
NYUST/EL

Outline

- 5.1 MPEG-2 Decoder HW & SW Integration Flow**
- 5.2 Download the Linux Kernel**
- 5.3 IDCT Driver Programming**
- 5.4 Compilation the MPEG-2 Source Code**
- 5.5 Programming IDCT SIP**
- 5.6 System Integrations**
- 5.7 Software Verification and LA Measurement**

5.1 MPEG-2 Decoder HW & SW Integration Flow

5.1 MPEG-2 Decoder HW & SW Integration Flow

5.2 Download the Linux Kernel

5.3 IDCT Driver Programming

5.4 Compilation the MPEG-2 Source Code

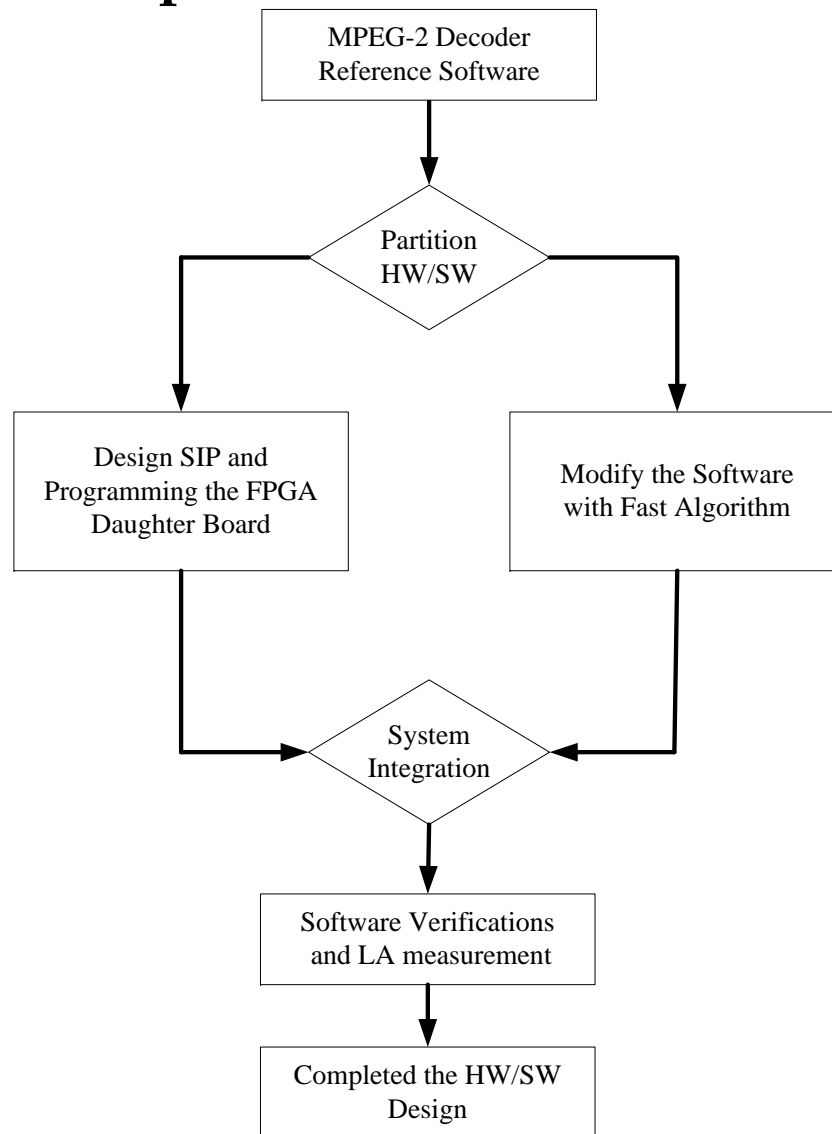
5.5 Programming IDCT SIP

5.6 System Integrations

5.7 Software Verification and LA Measurement

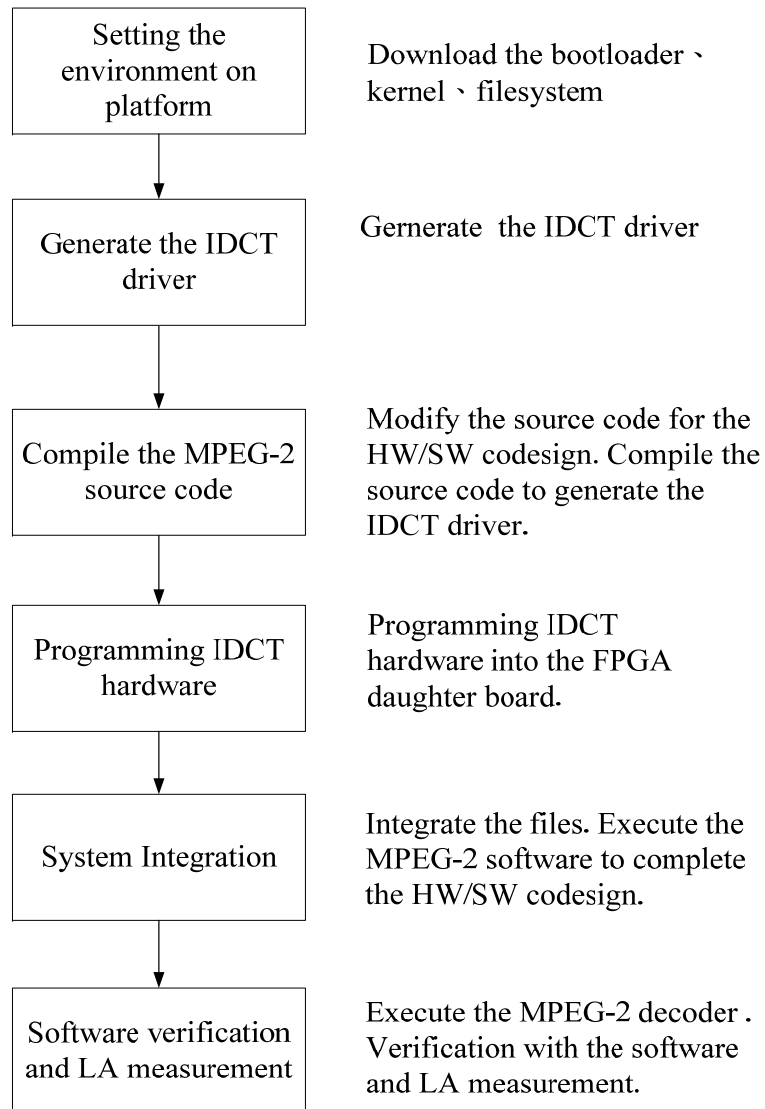
5.1 MPEG-2 Decoder HW & SW Integration Flow

◆ Integration Flow Graph:



5.1 MPEG-2 Decoder HW & SW Integration Flow

◆ Design Flows of Lab5



5.1 MPEG-2 Decoder HW & SW Integration Flow

◆ Prepared Design Files

■ Software Design Files

- MPEG-2 Decoder Execution File : mpeg2decode
- IDCT Driver : idct_driver.o

■ Hardware Design Files

- IDCT Hardware : idct.pof

5.2 Download the Linux Kernel

5.1 MPEG-2 Decoder HW & SW Integration Flow

5.2 Download the Linux Kernel

5.3 IDCT Driver Programming

5.4 Compilation the MPEG-2 Source Code

5.5 Programming IDCT SIP

5.6 System Integrations

5.7 Software Verification and LA Measurement

5.2 Download the Linux Kernel

◆ Modify Kernel Source Code File:

2.4.18-rmk7-pxa1-XSBase\arch\arm\mach-pxa\xhyper255.c

■ CPU->FPGA : bus width 16bits -> 32bits

```
/* setup memory timing for CS4 */
```

```
MSC2 = MSC_CS(4, MSC_RBUFF(MSC_RBUFF_SLOW) |
```

```
    MSC_RRR (2) |
```

```
    MSC_RDN (4) |
```

```
    MSC_RDF (4) |
```

```
    MSC_RBW (1) |
```

```
    MSC_RT  (0))
```

Rom Bus Width 16 Bits

Modify Rom Bus (line 176)

```
MSC2 = MSC_CS(4, MSC_RBUFF(MSC_RBUFF_SLOW) |
```

```
    MSC_RRR (7) |
```

```
    MSC_RDN (15) |
```

```
    MSC_RDF (15) |
```

```
    MSC_RBW (0) |
```

```
    MSC_RT  (1))
```

Rom Bus Width 32 Bits

5.2 Download the Linux Kernel

- ◆ [root\$super root]# vi /home/255/kernel/2.4.18-rmk7-pxal-XSBase255B/arch/arm/mach-pxa/xhyper255.c

```
root@localhost:~  
檔案(F) 編輯(E) 顯示(V) 終端機(T) 移至(G) 求助(H)  
MSC_RBW(1) |  
MSC_RT(0));  
printk(KERN_INFO "MCS1 = 0x%08x\n", MSC1);  
  
/* setup memory timing for CS4/5 */  
MSC2 = MSC_CS(4, MSC_RBUFF(MSC_RBUFF_SLOW) |  
MSC_RRR(7) |  
MSC_RDN(15) |  
MSC_RDF(15) |  
MSC_RBW(0) |  
MSC_RT(1)) |  
MSC_CS(5, MSC_RBUFF(MSC_RBUFF_SLOW) |  
MSC_RRR(2) |  
MSC_RDN(4) |  
MSC_RDF(4) |  
MSC_RBW(1) |  
MSC_RT(0));  
printk(KERN_INFO "MCS2 = 0x%08x\n", MSC2);  
"/home/255/kernel/2.4.18-rmk7-pxal-XSBase255B/arch/arm/mach-pxa/xhyper255.c" [唯讀] 220L, 7174C 176, 16 82%
```

5.2 Download the Linux Kernel

◆ Install Toolchain

■ [root\$super root]# vi ~/.bash_profile

● bash_profile : Setup the Environment



```
root@localhost:~  
檔案(F) 編輯(E) 顯示(V) 終端機(T) 移至(G) 求助(H)  
# .bash_profile  
  
# Get the aliases and functions  
if [ -f ~/.bashrc ]; then  
    . ~/.bashrc  
fi  
  
# User specific environment and startup programs  
  
PATH=$PATH:$HOME/bin  
PATH=$PATH:/usr/local/hybus-arm-linux-R1.1/bin/  
BASH_ENV=$HOME/.bashrc  
USERNAME="root"  
  
export USERNAME BASH_ENV PATH
```

■ [root\$super root]# source ~/.bash_profile

■ Logout and Login Linux System

5.2 Download the Linux Kernel

◆ Compile Kernel

- [root\$super root]# cd /home/255/kernel/2.4.18-rmk7-pxal-XSBase255B/
- [root\$super 2.4.18-rmk7-pxal-XSBase255B]#
make clean
- [root\$super 2.4.18-rmk7-pxal-XSBase255B]#
make dep
- [root\$super 2.4.18-rmk7-pxal-XSBase255B]#
make zImage

◆ Generation kernel image file “zImage” in “arch/arm/boot” directory

5.2 Download the Linux Kernel

◆ Setup TFTP and BOOTP Server Environment:

■ [root\$super root]# vi /etc/xinetd.d/bootp

```

root@localhost: /home/255/kernel/2.4.18-rmk7-pxa1-XSBase255B
檔案(F) 編輯(E) 顯示(V) 終端機(T) 移至(G) 求助(H)
service bootps
{
    disable     = no
    socket_type = dgram
    protocol    = udp
    wait        = yes
    user        = root
    server      = /usr/sbin/bootpd
}
/etc/xinetd.d/bootp 9L, 144C 9,1 全部

```

■ [root\$super root]# vi /etc/bootptab

```

root@localhost: /home/255/kernel/2.4.18-rmk7-pxa1-XSBase255B
檔案(F) 編輯(E) 顯示(V) 終端機(T) 移至(G) 求助(H)
test:\
ht=1:\
ha=0x151436188A11:\
ip=192.168.1.50:\
sm=255.255.255.0
"/etc/bootptab" 5L, 89C 5,1 全部

```

Give the board IP to setting bootptab file

5.2 Download the Linux Kernel

◆ Restart “Xined” Server

- [root\$super root]# /etc/rc.d/init.d/xinetd restart



A terminal window titled 'root@localhost:~' with a menu bar containing '檔案(F)', '編輯(E)', '顯示(V)', '終端機(T)', '移至(G)', and '求助(H)'. The terminal shows the command `/etc/rc.d/init.d/xinetd restart` being executed. The output is: `停止 xinetd:`, `啟動 xinetd:`, and a prompt `[root@localhost root]#`. A red box highlights the command line. A blue callout bubble points to the command with the text 'Restart “Xined” Server OK Message!'. To the right of the terminal, there are two green '確定' (Confirm) buttons in brackets.

```
root@localhost:~  
[root@localhost root]# /etc/rc.d/init.d/xinetd restart  
停止 xinetd:  
啟動 xinetd:  
[root@localhost root]#
```

◆ Copy Kernel Image to TFTP Server

- [root\$super root]# cp /home/255/kernel/2.4.18-rmk7-pxal-XSBase255B/arch/arm/boot/zImage /tftpboot

◆ Open Terminal and Power On XScale/PXA255

5.2 Download the Linux Kernel

- ◆ After booting. Terminal will show the message “Autoboot in progress”.
- ◆ Press any key to stop the process in three seconds. Enter command mode on Bootloader.

```
ddd - 超級終端機
檔案(F) 編輯(E) 檢視(V) 呼叫(C) 轉送(T) 說明(H)

XSBASE-R1
Copyright (C) 2004 Chhnet Co.,. ltd.
Support: http://www.hhnet.com.tw

Autoboot in progress, press any key to stop ..
Autoboot aborted
Type "help" to get a list of commands
XSBASE> _
```

Bootloader Command Line "XSBASE>"

5.2 Download the Linux Kernel

◆ Use TFTP Command Download Kernel

- [root\$super root]# tftp zImage kernel

ddd - 超級終端機

檔案(F) 編輯(E) 檢視(V) 呼叫(C) 轉送(T) 說明(H)

Enter "tftp zImage kernel" command

```

Autoboot aborted
Type "help" to get a list of commands
XSBASE> tftp zImage kernel
No IP. Bootp start...
Our Ethernet address is 1514 3618 8A11.
    Sending bootp packet...

Bootp Packet received.
    Host    (server) Ethernet : 000C 2954 0812
    Host    (server) IP       : 192.168.1.99
    Client  (target) Ethernet : 1514 3618 8A11
    Client  (target) IP       : 192.168.1.50

TFTP Start...
    Host    (server) IP       : 192.168.1.99
    Client  (target) IP       : 192.168.1.50
    Loading Filename          : zImage
    Save Address              : 0xA0008000

Loading start...
    0x0010D34C (1102668) bytes received.
    tftp done.

XSBASE> _
  
```

TFTP Download "kernel" OK Message

連線 00:01:24 自動偵測 115200 8-N-1 SCROLL CAPS NUM 擷 列印

5.2 Download the Linux Kernel

■ [root\$super root]# flash kernel

```
Host (server) Ethernet : 000C 2954 0812
Host (server) IP       : 192.168.1.99
Client (target) Ethernet : 1514 3618 8A11
Client (target) IP      : 192.168.1.50

TFTP Start...
Host (server) IP       : 192.168.1.99
Client (target) IP     : 192.168.1.50
Loading Filename       : zImage
Save Address

Loading start...
0x0010D34C (1102668) bytes saved.
tftp done.

XSBASE> flash kernel
Saving kernel to flash...
Erase flash blocks from 0x000C0000 to 0x002B
Erase Block at : 0x00280000.
Done.
Write to flash...
Writing at 0x002C0000...
Done
XSBASE> _
```

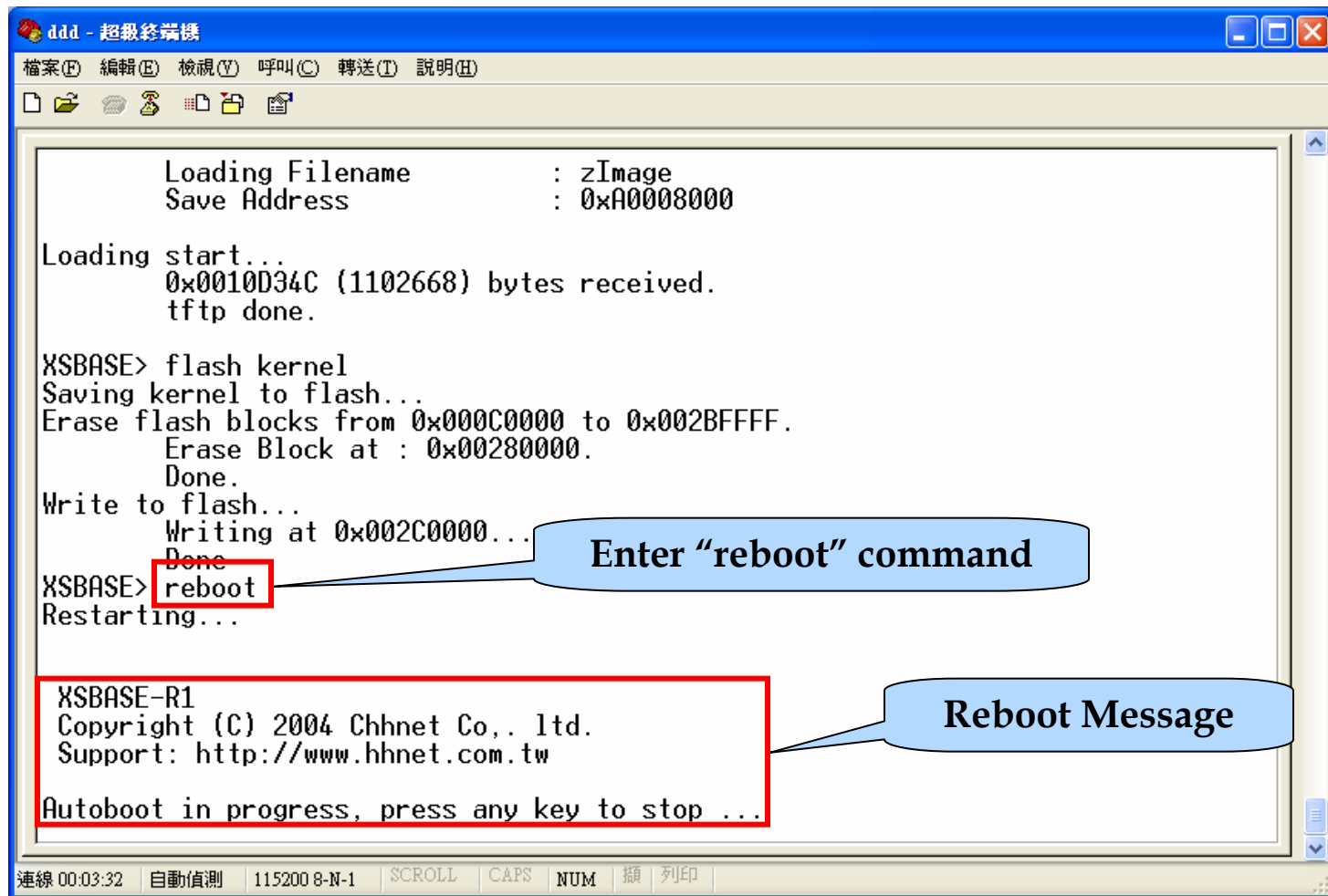
Enter "flash kernel" command

Write "kernel" to Flash Memory OK Message

5.2 Download the Linux Kernel

◆ Reboot with XScale PXA255

■ XSCALE> reboot



```
ddd - 超級終端機
檔案(F) 編輯(E) 檢視(V) 呼叫(C) 轉送(T) 說明(H)

Loading Filename      : zImage
Save Address         : 0xA0008000

Loading start...
0x0010D34C (1102668) bytes received.
tftp done.

XSBASE> flash kernel
Saving kernel to flash...
Erase flash blocks from 0x000C0000 to 0x002BFFFF.
Erase Block at : 0x00280000.
Done.
Write to flash...
Writing at 0x002C0000...
Done
XSBASE> reboot
Restarting...

XSBASE-R1
Copyright (C) 2004 Chhnet Co., Ltd.
Support: http://www.hhnet.com.tw

Autoboot in progress, press any key to stop ...
```

Enter "reboot" command

Reboot Message

5.3 IDCT Driver Programming

5.1 MPEG-2 Decoder HW & SW Integration Flow

5.2 Download the Linux Kernel

5.3 IDCT Driver Programming

5.4 Compilation the MPEG-2 Source Code

5.5 Programming IDCT SIP

5.6 System Integrations

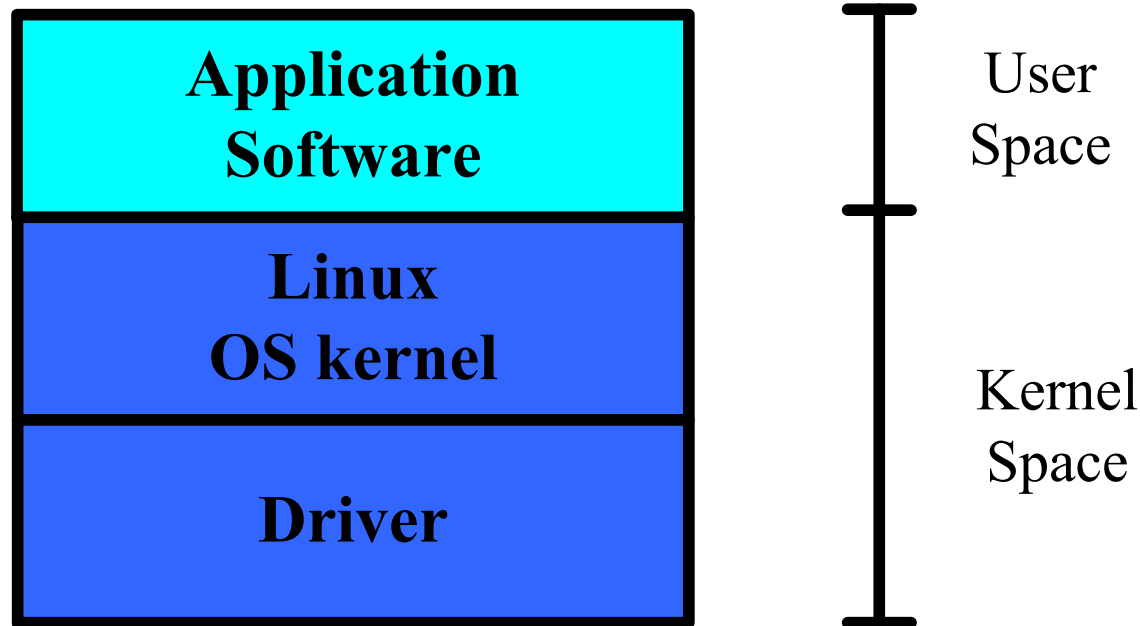
5.7 Software Verification and LA Measurement

5.3 IDCT Driver Programming

- ◆ Programming the IDCT Driver Functions
 - Open()
 - Read()
 - Write()
 - Close()
- ◆ Compile the IDCT Driver
 - Create the Driver Module

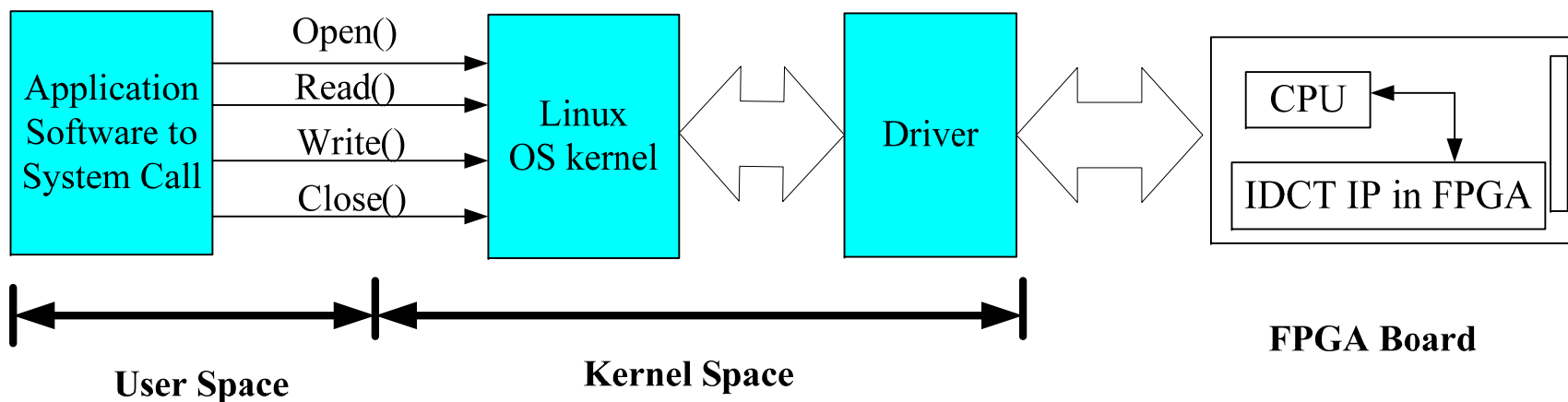
5.3 IDCT Driver Programming

- ◆ Kernel Space : Linux OS kernel and the IDCT Driver
- ◆ User Space : Application Software



5.3 IDCT Driver Programming

◆ Flow of the Software Functions to Hardware Operations



5.3 IDCT Driver Programming

■ Programming the IDCT Driver(1/6)

◆ File Name : idct_driver.c

```
-----  
#include "fpga_board_ioctl.h"  
#include "fpga_reg_set.h"  
  
#define FPGA_BOARD_MAJOR 72           //Major number  
#define DEVICE_NAME "FPGA_BOARD_6"  
#define INT_OWNER    "FPGA_BOARD_6"  
  
//open(),read(),write(),close()  
static struct file_operations fpga_board_fops={  
open:      fpga_board_open,  
release:   fpga_board_release,  
read:      fpga_board_read,  
write:     fpga_board_write,  
};  
-----
```

5.3 IDCT Driver Programming

■ Programming the IDCT Driver(2/6)

◆ File Name : idct_driver.c

```
-----  
int __init init_fpga_board_module_6(void) //When insmod the driver  
{  
    register_chrdev(FPGA_BOARD_MAJOR,  
                    DEVICE_NAME, &fpga_board_fops); //Register the driver  
    //printf("Init v1\n"); //Make sure the driver activity  
    //printf("frank\n");  
    return(0);  
}  
  
void __exit cleanup_fpga_board_module_6(void) //When remod the driver  
{  
    unregister_chrdev(FPGA_BOARD_MAJOR, DEVICE_NAME ); //Unregister the driver  
    //printf("cleanup FPGA Board\n"); //Make sure the driver activity  
}  
-----
```

5.3 IDCT Driver Programming

■ Programming the IDCT Driver(3/6)

◆ Open()

◆ File Name : idct_driver.c

```
-----  
//Physical memory address map virtual memory address  
static void phy_to_vir(FPGA_BOARD * f_b_p)  
{  
    request_mem_region(DARAM_PHY_ADDR, 1024, "test");  
    f_b_p->DARAM_VIR_ADDR = (unsigned long)ioremap(DARAM_PHY_ADDR,1024);  
    //Map function 1024 bytes  
}  
  
//When open the driver  
static int fpga_board_open(struct inode * inode_p, struct file * filp)  
{  
    static FPGA_BOARD f_b;  
    phy_to_vir(&f_b); //Call the "phy_to_vir" function  
    filp->private_data = &f_b;  
    return(0);  
}  
-----
```


5.3 IDCT Driver Programming

■ Programming the IDCT Driver(4/6)

◆ Write()

◆ File Name : idct_driver.c

```
-----
static ssize_t fpga_board_write(struct file * filp, const char * buff,
                                size_t count, loff_t *offp)
{
    // "count" value from application
    unsigned long value;
    int i,j;
    FPGA_BOARD * f_b_p;
    char *p = kmalloc(count, GFP_KERNEL); //require the memory "count" bytes
    f_b_p = filp->private_data;
    for(i=0; i<count; i+=4)
    {
        value = (p[i] | p[i+1]<<8 | p[i+2]<<16 | p[i+3]<<24);
        writel(value, f_b_p->DARAM_VIR_ADDR+i); //Write 32-bits data
    }
    //The virtual address
    return(count);
}
-----
```

5.3 IDCT Driver Programming

■ Programming the IDCT Driver(5/6)

◆ read()

◆ File Name : idct_driver.c

```
-----  
static ssize_t fpga_board_read(struct file * filp, char * buff,  
                               size_t count, loff_t *offp)  
{  
    int i;  
    unsigned long value; //require the memory "count" bytes  
    char *p = kmalloc(count, GFP_KERNEL);  
    FPGA_BOARD * f_b_p;  
    f_b_p = filp->private_data;  
    memcpy_fromio(p, f_b_p->DARAM_VIR_ADDR, count); //Read data from HW  
    copy_to_user((char *)buff, (char *)p, count); //copy data from kernel  
                                                    space to user space  
    return(count);  
}  
-----
```

5.3 IDCT Driver Programming

■ Programming the IDCT Driver(6/6)

◆ Close()

◆ File Name : idct_driver.c

```
//When the user application call close()
```

```
The close system in Linux kernel mean "release"
```

```
static int fpga_board_release(struct inode * inode_p, struct file * filp)
{
    FPGA_BOARD * f_b_p;
    f_b_p = filp->private_data;
    iounmap((void *)f_b_p->DARAM_VIR_ADDR );
    return(0) //Driver close, release the memory
}
```

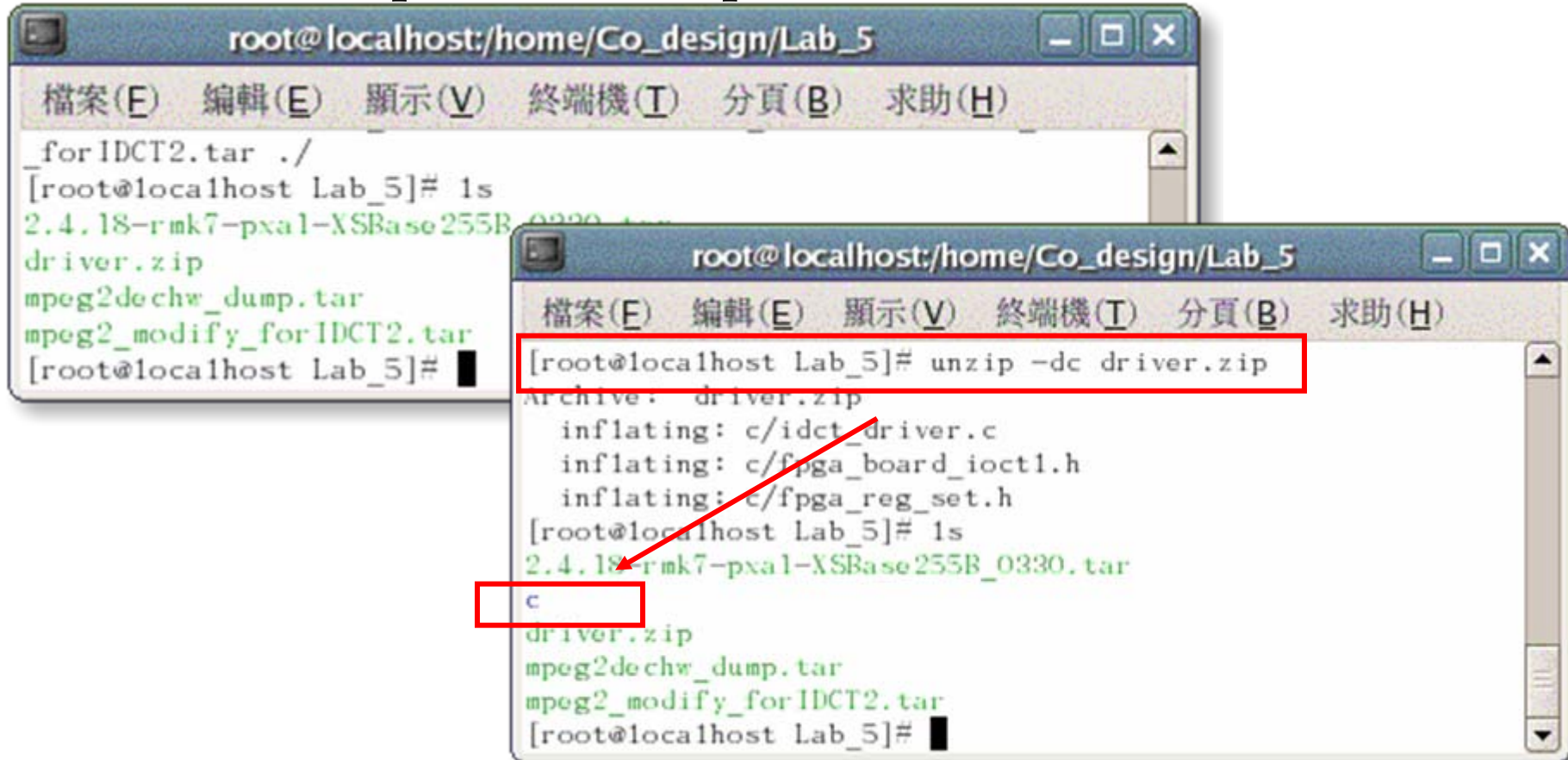
5.3 IDCT Driver Programming

■ Compilation the Driver

◆ Decompress “idct_driver.zip” to get Driver Source Code

■ Lab_5>ls

■ Lab_5>unzip -dc driver.zip



```
root@localhost:/home/Co_design/Lab_5
檔案(E) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)
_forIDCT2.tar ./
[root@localhost Lab_5]# ls
2.4.18-rmk7-pxa1-XSBase255B_0330.tar
driver.zip
mpeg2dechw_dump.tar
mpeg2_modify_forIDCT2.tar
[root@localhost Lab_5]#

root@localhost:/home/Co_design/Lab_5
檔案(E) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)
[root@localhost Lab_5]# unzip -dc driver.zip
Archive:  driver.zip
  inflating: c/idct_driver.c
  inflating: c/fpga_board_ioct1.h
  inflating: c/fpga_reg_set.h
[root@localhost Lab_5]# ls
2.4.18-rmk7-pxa1-XSBase255B_0330.tar
c
driver.zip
mpeg2dechw_dump.tar
mpeg2_modify_forIDCT2.tar
[root@localhost Lab_5]#
```

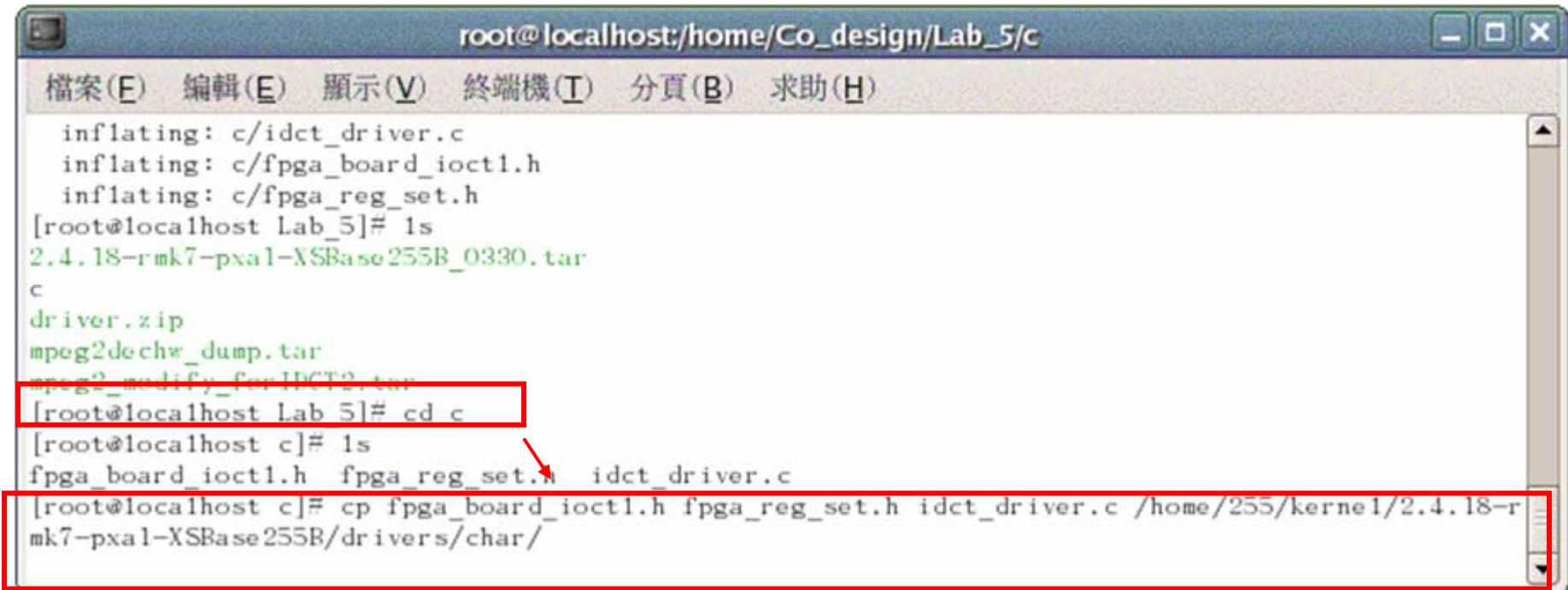
5.3 IDCT Driver Programming

■ Compilation the Driver

◆ Copy “idct_driver.c”, “fpga_board_ioctl.h”, “fpga_reg_set.h” into “2.4.18-rmk7-pxa1-XSBase255B/drivers/char” directory

■ Lab_5>cd c

■ C>cp fpga_board_ioctl.h fpga_reg_set.h idct_driver.c
/home/2.4.18-rmk7-pxa1-XSBase255B/drivers/char/



A terminal window titled "root@localhost:/home/Co_design/Lab_5/c" showing the following commands and output:

```
inflating: c/idct_driver.c
inflating: c/fpga_board_ioctl.h
inflating: c/fpga_reg_set.h
[root@localhost Lab_5]# ls
2.4.18-rmk7-pxa1-XSBase255B_0330.tar
c
driver.zip
mpeg2dechw_dump.tar
mpeg2_modify_forIDCT2.tar
[root@localhost Lab_5]# cd c
[root@localhost c]# ls
fpga_board_ioctl.h fpga_reg_set.h idct_driver.c
[root@localhost c]# cp fpga_board_ioctl.h fpga_reg_set.h idct_driver.c /home/255/kernel/2.4.18-rmk7-pxa1-XSBase255B/drivers/char/
```

Red boxes highlight the directory change and the copy command. A red arrow points from the text "C>cp fpga_board_ioctl.h fpga_reg_set.h idct_driver.c /home/255/kernel/2.4.18-rmk7-pxa1-XSBase255B/drivers/char/" to the copy command in the terminal.

5.3 IDCT Driver Programming

■ Compilation the Driver

◆ Modify Makefile to Compile the Driver

■ #>cd /home/255/kernel/2.4.18-rmk7-pxa1-XSBase255B/drivers/char

■ char>vi Makefile

```
root@localhost:/home/255/kernel/2.4.18-rmk7-pxa1-XSBase255B/drivers/char

檔案(E) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

Config.in      fpga_board_3.c  machzwd.c      riscom8.h      tty_io.c
conmakehash    fpga_board_4.c  Makefile       riscom8_reg.h  tty_ioctl.c
conmakehash.c  fpga_board_5.c  mem.c          rocket.c        tty_ioctl.o
console.c      fpga_board_6.c  mem.o          rocket_int.h    tty_io.o
console_macros.h fpga_board_6.o  misc.c         rsfl6fmi.h     vc_screen.c
consolemap.c   fpga_board_7.c  misc.o         rtc.h           vc_screen.o
consolemap_deftbl.c fpga_board_7.o mixcomwd.c     rtc-orig.c     vme_scc.c
consolemap_deftbl.o fpga_board_8.c moxa.c         sal100-rtc.c   vt.c
consolemap.o   fpga_board_8.o msbusmouse.c  sal100_wdt.c  vt.o
console.o      fpga_board_9.c mwave          sal111_keyb.c w83877f_wdt.c
cp437.uni     fpga_board_9.o mxser.c        sbc60xxwdt.c  wd50lp.h
cyclades.c    fpga_board_ioctl.h n_hdlc.c       scan_keyb.c   wdt285.c
decserial.c    fpga_reg_set.h  n_r3964.c     scan_keyb.h    wdt977.c
defkeymap.c    ftape           n_tty.c       scc.h          wdt.c
defkeymap.map  gc_kbmap.h      n_tty.o       selection.c    wdt_pci.c
defkeymap.o    gc_keyb.c       nvram.c       selection.o
digil.h        gckeymap.c      nwbutton.c    ser_a2232.c
[root@localhost char]# vi Makefile
```


5.3 IDCT Driver Programming

■ Compilation the Driver

- ◆ Modify obj-\$(CONFIG_FPGA)
- ◆ Add “idct_driver.o” in “Makefile”

■ #char>vi Makefile

```
obj-$(CONFIG_BVME6000_SCC) += generic_serial.o vme_scc.o
obj-$(CONFIG_SERIAL_TX3912) += generic_serial.o serial_tx3912.o
obj-$(CONFIG_ADS7843) += ads7843_ts.o
obj-$(CONFIG_BG_LIGHT) += lcd_adjust.o
#obj-$(CONFIG_RTC4513) += rtc.o
#obj-$(CONFIG_FPGA) += fpga_board_0.o fpga_board_1.o fpga_board_2.o fpga_board_3.o fpga_board_
4.o fpga_board_5.o f_6_read.o

obj-$(CONFIG_FPGA) += fpga_board_7.o fpga_board_8.o fpga_board_9.o fpga_board_10.o f_6_read.o
idct_driver.o
```

- ◆ Push “Esc” button, use “:wq” to save and exit “Makefile”

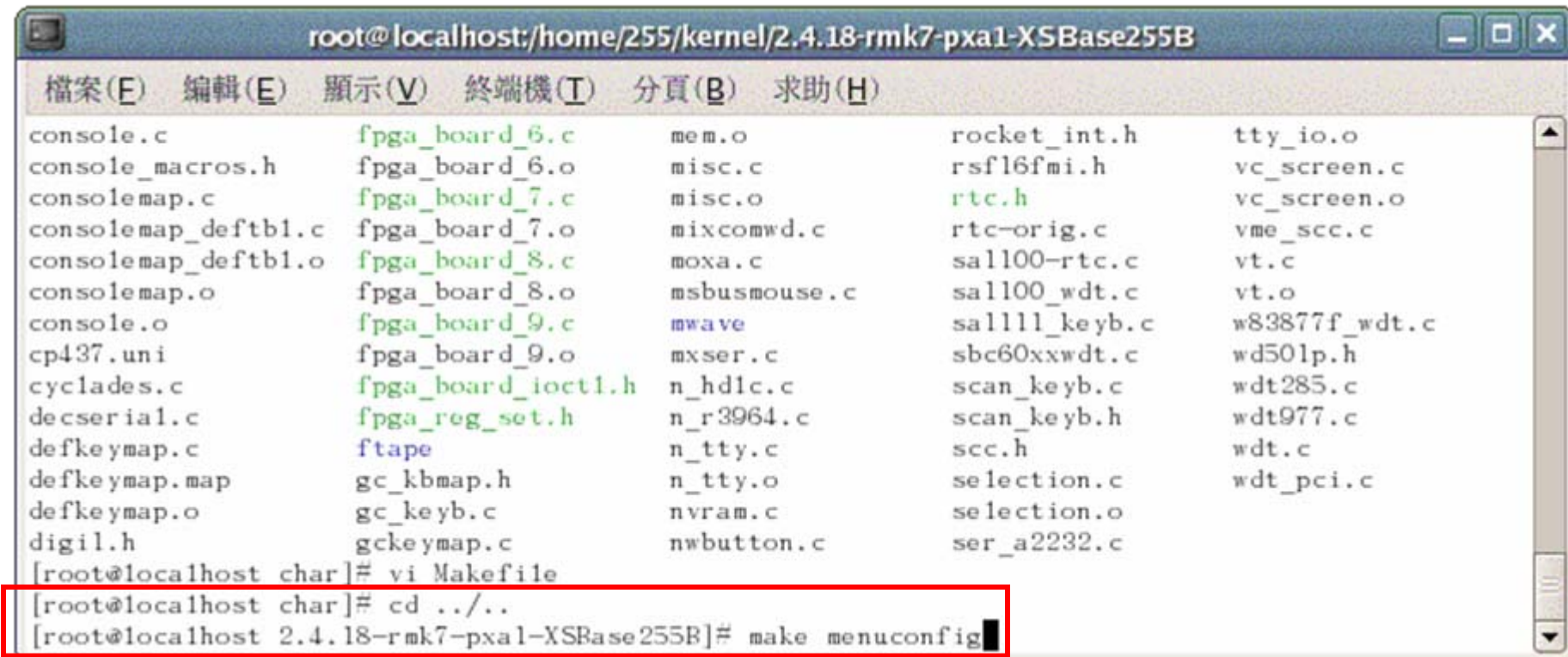
5.3 IDCT Driver Programming

■ Compilation the Driver

◆ #char>cd ../../

■ Top Folder of the Kernel Package.

◆ #2.4.18-rmk7-pxa1-XSBase255B>>make menuconfig

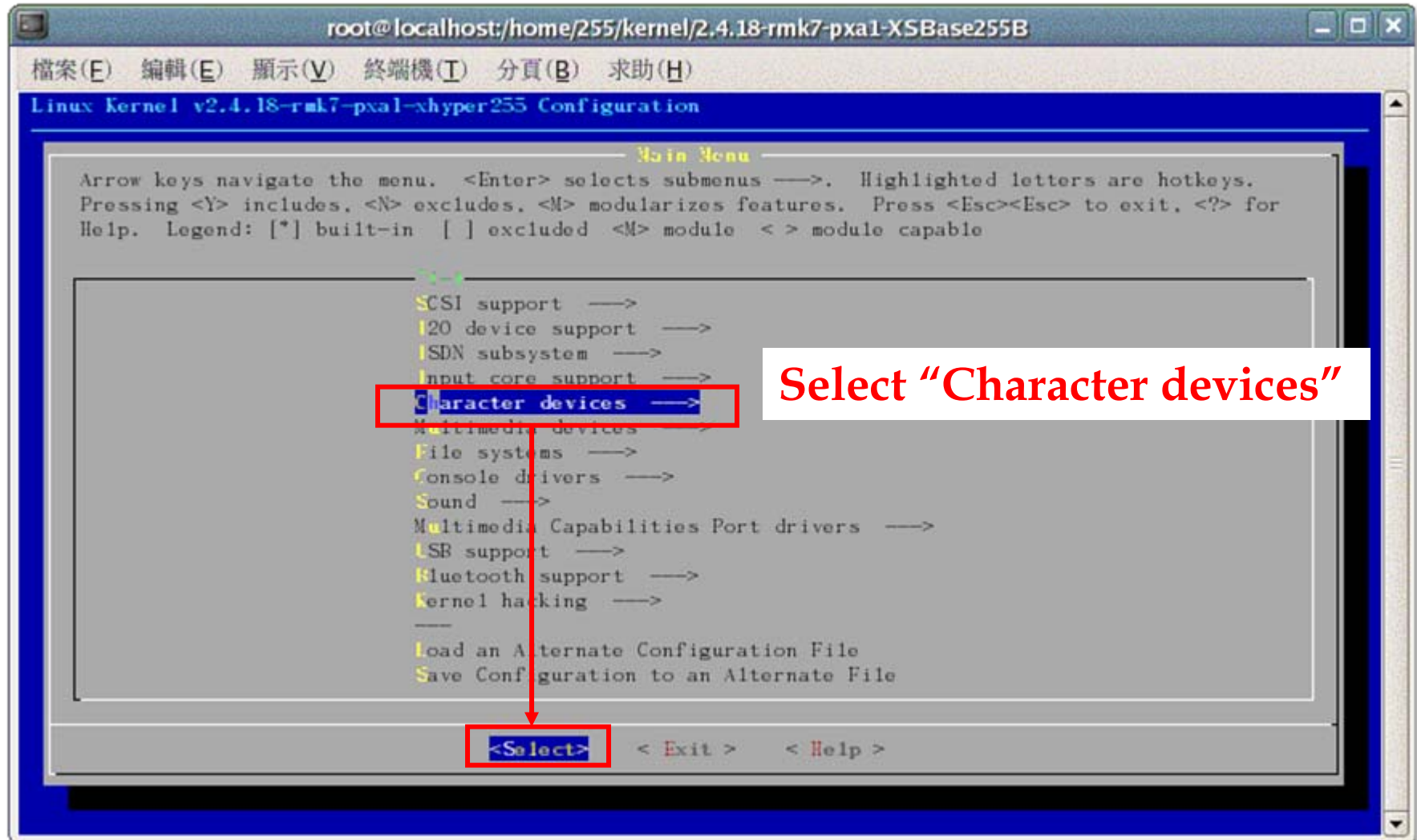


```
root@localhost:/home/255/kernel/2.4.18-rmk7-pxa1-XSBase255B
檔案(E) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)
console.c          fpga_board_6.c    mem.o             rocket_int.h      tty_io.o
console_macros.h   fpga_board_6.o    misc.c            rsfl6fmi.h        vc_screen.c
consolemap.c       fpga_board_7.c    misc.o            rtc.h              vc_screen.o
consolemap_deftb1.c fpga_board_7.o    mixcomwd.c        rtc-orig.c         vme_scc.c
consolemap_deftb1.o fpga_board_8.c    moxa.c            s1100-rtc.c        vt.c
consolemap.o       fpga_board_8.o    msbusmouse.c     s1100_wdt.c        vt.o
console.o          fpga_board_9.c    mwave             s1111_keyb.c       w83877f_wdt.c
cp437.uni          fpga_board_9.o    mxser.c           sbc60xxwdt.c       wd501p.h
cyclades.c         fpga_board_ioct1.h n_hdlc.c          scan_keyb.c        wdt285.c
decserial.c        fpga_reg_set.h   n_r3964.c         scan_keyb.h        wdt977.c
defkeymap.c        ftape             n_tty.c           scc.h              wdt.c
defkeymap.map      gc_kbmap.h        n_tty.o           selection.c         wdt_pci.c
defkeymap.o        gc_keyb.c         nvram.c           selection.o
digil.h            gckeymap.c        nwbutton.c        ser_a2232.c
[root@localhost char]# vi Makefile
[root@localhost char]# cd ../../
[root@localhost 2.4.18-rmk7-pxa1-XSBase255B]# make menuconfig
```


5.3 IDCT Driver Programming

■ Compilation the Driver

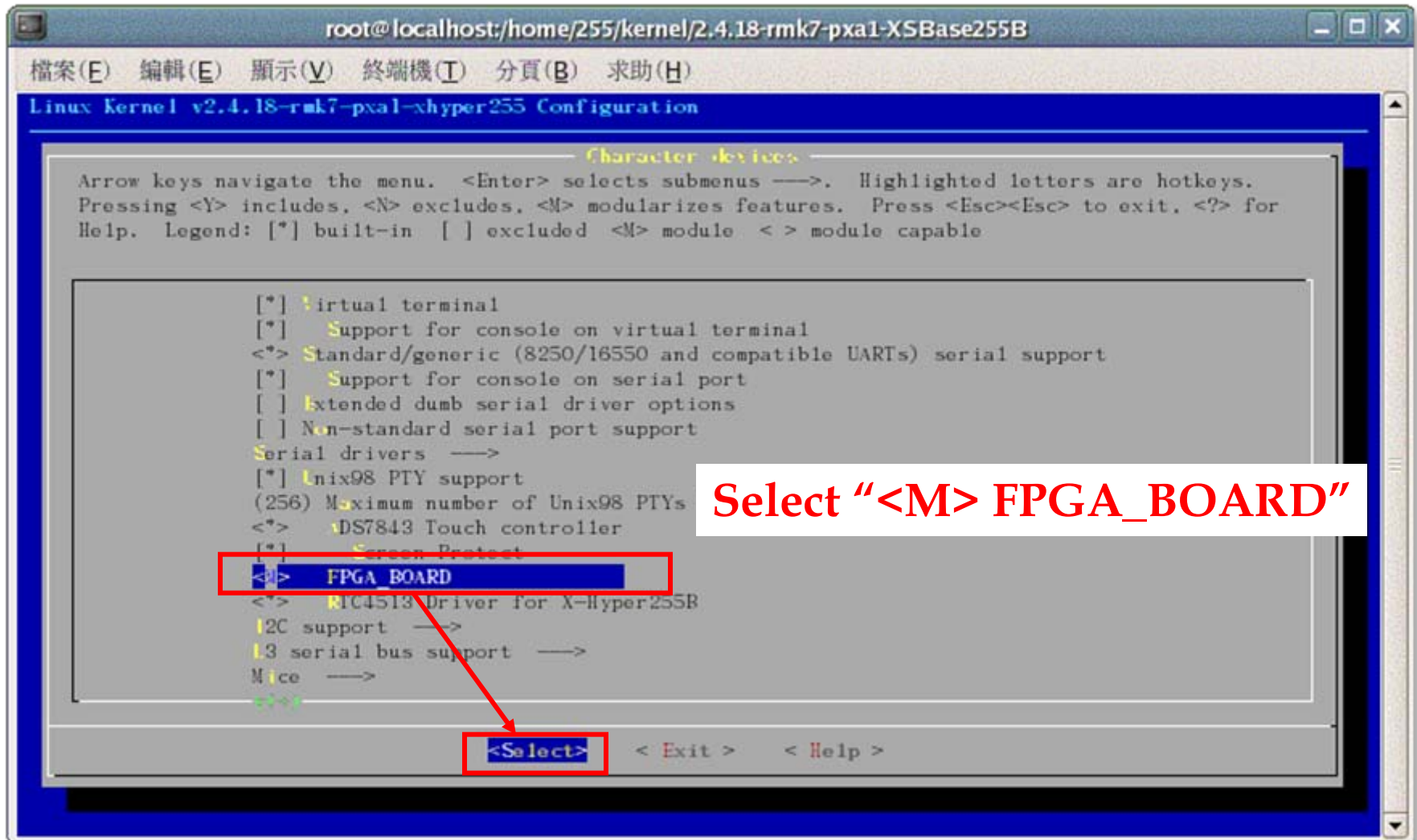
◆ Choose Character Devices



5.3 IDCT Driver Programming

■ Compilation the Driver

◆ Choose Module



5.3 IDCT Driver Programming

■ Compilation the Driver

◆ Compile the Driver

- #2.4.18-rmk7-pxa1-XSBase255B>make modules
- Path :2.4.18-rmk7-pxa1-XSBase255B/drivers/char
- Find the driver -"idct_driver.o"

```
root@localhost:/home/255/kernel/2.4.18-rmk7-pxa1-XSBase255B/drivers/char
檔案(F) 編輯(E) 顯示(V) 終端機(T) 移至(G) 求助(H)
console.c      fpga_board_ioctl.h  pc_keyb.c      specialix_io8.h
console_macros.h fpga_reg_set.h      pcmcia         stallion.c
consolemap.c   ftape              pcwd.c         sxboards.h
cp437.uni      gc_kbmap.h         pcxx.c         sx.c
cyclades.c     gc_keyb.c          pcxx.h         sx.h
decsrial.c     gckeymap.c         ppdev.c        sxwindow.h
defkeymap.c     gckeymap.map       pty.c          synclink.c
defkeymap.map   generic_serial.c   q40_keyb.c     sysrq.c
digi.h          h8.c               qpmouse.c      test.c
digi_bios.h     h8.h              qtronix.c      toshiba.c
digiFepl.h      hp600_keyb.c       qtronixmap.map tpqic02.c
digi_fep.h      hp_keyb.c          random.c       tty_io.c
digi.h          hp_psaux.c         raw.c          tty_ioctl.c
digiPCI.h       i810_rng.c         README.computone vc_screen.c
dn_keyb.c        i810-tco.c         README.cycladesZ vme_scc.c
drm             i810-tco.h         README.cyclomY   vt.c
drm-4.0         i8k.c              README.epca      w83877f_wdt.c
dsl620.c        ib700wdt.c         README.scc       wd501p.h
dsp56k.c        idct_driver.o      rio             wdt285.c
dtlk.c          ip2                riscom8.c       wdt977.c
dz.c            ip2.c              riscom8.h       wdt.c
dz.h            ip2main.c          riscom8_reg.h   wdt_pci.c
ec3104_keyb.c   [root@localhost char]#
```

5.4 Compilation the MPEG-2 Source Code

5.1 MPEG-2 Decoder HW & SW Integration Flow

5.2 Download the Linux Kernel

5.3 IDCT Driver Programming

5.4 Compilation the MPEG-2 Source Code

5.5 Programming IDCT SIP

5.6 System Integrations

5.7 Software Verification and LA Measurement

5.4 Compilation the MPEG-2 Source Code

- ◆ Step 1. Modify MPEG2 decoder source to complete the HW/SW co-design

- Modify files : getpic.c , idct.c

- ◆ getpic.c : mpeg2/src/mpeg2dec/getpic.c

- ◆ File Name : getpic.c

```
-----  
.....  
//Add these header file to implement the HW/SW  
#include <sys/types.h> #include <sys/stat.h>  
#include <fcntl.h>  
#include "fpga_board_ioctl.h"  
#include "fpga_reg_set.h"  
int ffd;  
void Decode_Picture(...)  
{  
.....  
}
```

```
-----
```

5.4 Compilation the MPEG-2 Source Code

◆ Step 2. getpic.c : mpeg2/src/mpeg2dec/getpic.c

◆ File Name : getpic.c

```
-----  
void Decode_Picture(bitstream_framenum,sequence_framenum)  
int bitstream_framenum, sequence_framenum;  
{  
    ffd = open("/dev/fpga_board_6", O_RDWR); //Open driver  
    //The IDCT driver is lined to "/dev/fpga_board_6"  
    .....  
    close(ffd); //Close driver  
}
```

```
-----
```

5.4 Compilation the MPEG-2 Source Code

◆ Step 3. getpic.c : mpeg2/src/mpeg2dec/getpic.c

◆ File Name : getpic.c

```
-----  
static void motion_compensation(...)  
{  
    ...  
    // if (Reference_IDCT_Flag)          //Mask these code  
    //     Reference_IDCT(ld->block[comp]);  
    // else  
    //     Fast_IDCT(ld->block[comp]);  
    put_idct_FPGA(ld->block[comp],ffd);  
    //Add source code to call the modify IDCT function to implement HW/SW  
    ...  
}
```

5.4 Compilation the MPEG-2 Source Code

◆ Step 4. getpic.c : mpeg2/src/mpeg2dec/getpic.c

◆ File Name : getpic.c

```
-----  
void put_idct_FPGA(short *block,int ffd)  
{  
    int y = 0;  
    int buffer[32];  
    int count;  
    for(y = 0; y < 8; y++)  
    {  
        //Merge data to transfer into the hardware  
        buffer[(0 + 4 * y)]=(block[0 + y] & 0xffff)|(block[8 + y] << 16);  
        buffer[(1 + 4 * y)]=(block[16 + y] & 0xffff)|(block[24 + y] << 16);  
        buffer[(2 + 4 * y)]=(block[32 + y] & 0xffff)|(block[40 + y] << 16);  
        buffer[(3 + 4 * y)]=(block[48 + y] & 0xffff)|(block[56 + y] << 16);  
    }  
    write(ffd, &buffer[0], 128);  
    for(y=0; y <= 4000; y++)  
        count =2;  
    .....  
}
```

5.4 Compilation the MPEG-2 Source Code

■ Modify the Software for SW/HW

◆ Step 5. getpic.c : mpeg2/src/mpeg2dec/getpic.c

◆ File Name : getpic.c

```
-----  
{  
.....  
read (ffd, &buffer[0], 128); //Read the data from the hardware  
for(y = 0; y < 8; y++)  
{  
    block[0 + (8 * y) ] = (short)(buffer[(0 + 4 * y)] & 0xffff);  
    block[1 + (8 * y) ] = (short)(buffer[(0 + 4 * y)] >>16);  
    block[2 + (8 * y) ] = (short)(buffer[(1 + 4 * y)] & 0xffff);  
    block[3 + (8 * y) ] = (short)(buffer[(1 + 4 * y)] >>16);  
    block[4 + (8 * y) ] = (short)(buffer[(2 + 4 * y)] & 0xffff);  
    block[5 + (8 * y) ] = (short)(buffer[(2 + 4 * y)] >>16);  
    block[6 + (8 * y) ] = (short)(buffer[(3 + 4 * y)] & 0xffff);  
    block[7 + (8 * y)] = (short)(buffer[(3 + 4 * y) ] >>16);  
}  
.....  
}  
-----
```

5.4 Compilation the MPEG-2 Source Code

- ◆ Step 6. Compile MPEG-2 Source Code for ARM
 - Modify Makefile
 - Path : /mpeg2/Makefile
 - Method 1
 - CC = gcc → Absolute Patch/arm-linux-gcc
 - **Modify the Makefile with the path of the compiler tool**
 - Example
 - CC = /home/toolchain/bin/arm-linux-gcc

5.4 Compilation the MPEG-2 Source Code

■ Modify Makefile with Method 1

- ◆ Push the Button “Esc” to exit Makefile
- ◆ Type “:wq!” to Quit Makefile and Save

```
# GNU gcc
CC = /usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc
CFLAGS = -O2

all: mpeg2decode mpeg2encode

mpeg2decode:
— INSERT —
```

Press “Esc”

```
all: mpeg2decode mpeg2encode

mpeg2decode:
:wq!
```

5.4 Compilation the MPEG-2 Source Code

■ Modify Makefile with Method 1

◆ Step 7. Get the MPEG-2 decoder execute file

■ #mpeg2>make clean; make

```
getbits.c          getvlc.h          motion.c          mpeg2d
getbits.o          getvlc.o          motion.o          mpeg2d
getblk.c          global.h          mpeg2dec.c       mpeg2d
[root@localhost mpeg2dec]# cd ..
[root@localhost src]# ls
foreman.m2v  mpeg2dec  mpeg2enc  xdpinfo
[root@localhost src]# cd ..
[root@localhost mpeg2]# ls
doc  Makefile  par  README  src  verify
[root@localhost mpeg2]# make clean;make
```

■ #mpeg2>cd src/mpeg2dec/

■ #mpeg2>ls

5.4 Compilation the MPEG-2 Source Code

■ Modify Makefile with Method 1

■ Get **mpeg2decode**

```
.o puthdr.o putmpg.o putvlc.o putbits.o motion.o predict.o readpic.o writepic.o transf
rtize.o ratectl.o stats.o -lm
make[1]: Leaving directory `/frank/ffff/mpeg2/src/mpeg2enc'
[root@localhost mpeg2]# ls
doc Makefile par README src verify
[root@localhost mpeg2]# cd src/mpeg2dec/
[root@localhost mpeg2dec]# ls
CHANGES          getblk.o          idct.c            mpeg2dec.h        mpeg2decode_hw
config.h          gethdr.c          idct.c.bak       mpeg2dec.o        mpeg2decodesof
display.c         gethdr.o          idct.o           mpeg2decode       README
display.o         getpic.c          idctref.c        mpeg2decode_0402_2ip.out recon.c
EXAMPLES         getpic.c.bak      idctref.o        mpeg2decode0402.out recon.o
fpga_board_ioctl.h getpic.o          IEEE1180         mpeg2decode0430   SPATIAL.DOC
fpga_reg_set.h   getvlc.c          Makefile         mpeg2decode0430-4000 spatscal.c
getbits.c        getvlc.h          motion.c         mpeg2decodels     spatscal.o
getbits.o        getvlc.o          motion.o         mpeg2decode3slow  store.c
getblk.c         global.h          mpeg2dec.c      mpeg2decodedef    store.c.bak
[root@localhost mpeg2dec]#
```

5.4 Compilation the MPEG-2 Source Code

■ Modify Makefile with Method 2

◆ Step 6. Compile MPEG-2 Source Code for ARM

■ Modify Makefile

■ Path : /mpeg2/Makefile

■ Method 2

● CC = gcc → arm-linux-gcc

- Modify the compiler for ARM
- Need export the environment variable in the Linux

● Example

- Modify the Makefile

CC= gcc → arm-linux-gcc

- Exit the Makefile
- Export the environment variable in the Linux and use “make” to comple

5.4 Compilation the MPEG-2 Source Code

■ Modify Makefile with Method 2

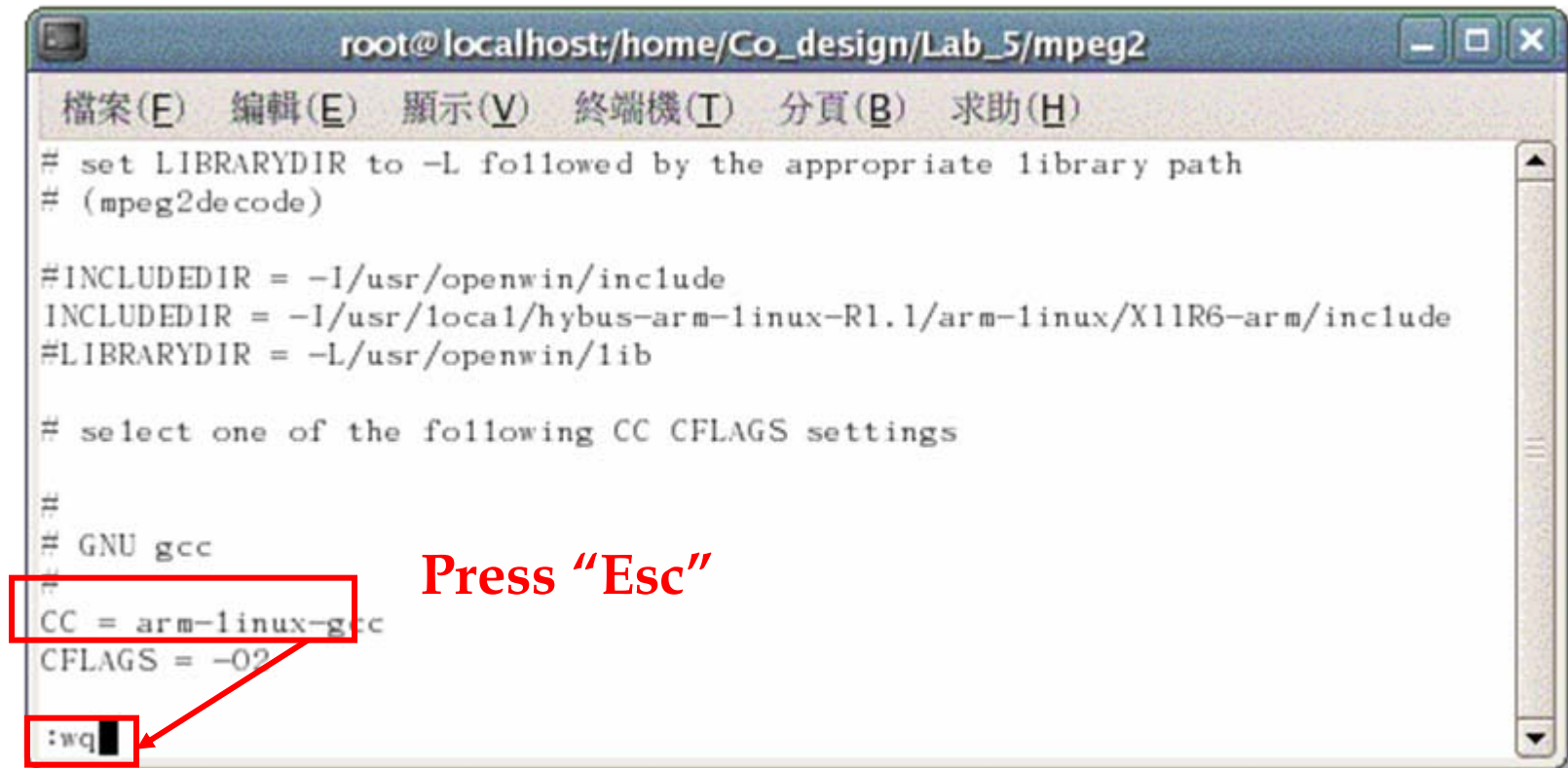
◆ #mpeg2>vi Makefile

```
[root@localhost frank]# cd mpeg2
[root@localhost mpeg2]# ls
doc  Makefile  par  README  src  verify
[root@localhost mpeg2]# ls
doc  Makefile  par  README  src  verify
[root@localhost mpeg2]# vi Makefile
```

5.4 Compilation the MPEG-2 Source Code

■ Modify Makefile with Method 2

- ◆ Modify “Makefile” : CC = arm-linux-gcc
- ◆ Press “Esc” button.
- ◆ Use “:wq” the save and exit.



```
root@localhost:/home/Co_design/Lab_5/mpeg2
檔案(E) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)
# set LIBRARYDIR to -L followed by the appropriate library path
# (mpeg2decode)

#INCLUDEDIR = -I/usr/openwin/include
INCLUDEDIR = -I/usr/local/hybus-arm-linux-R1.1/arm-linux/X11R6-arm/include
#LIBRARYDIR = -L/usr/openwin/lib

# select one of the following CC CFLAGS settings

#
# GNU gcc
#
CC = arm-linux-gcc
CFLAGS = -O2

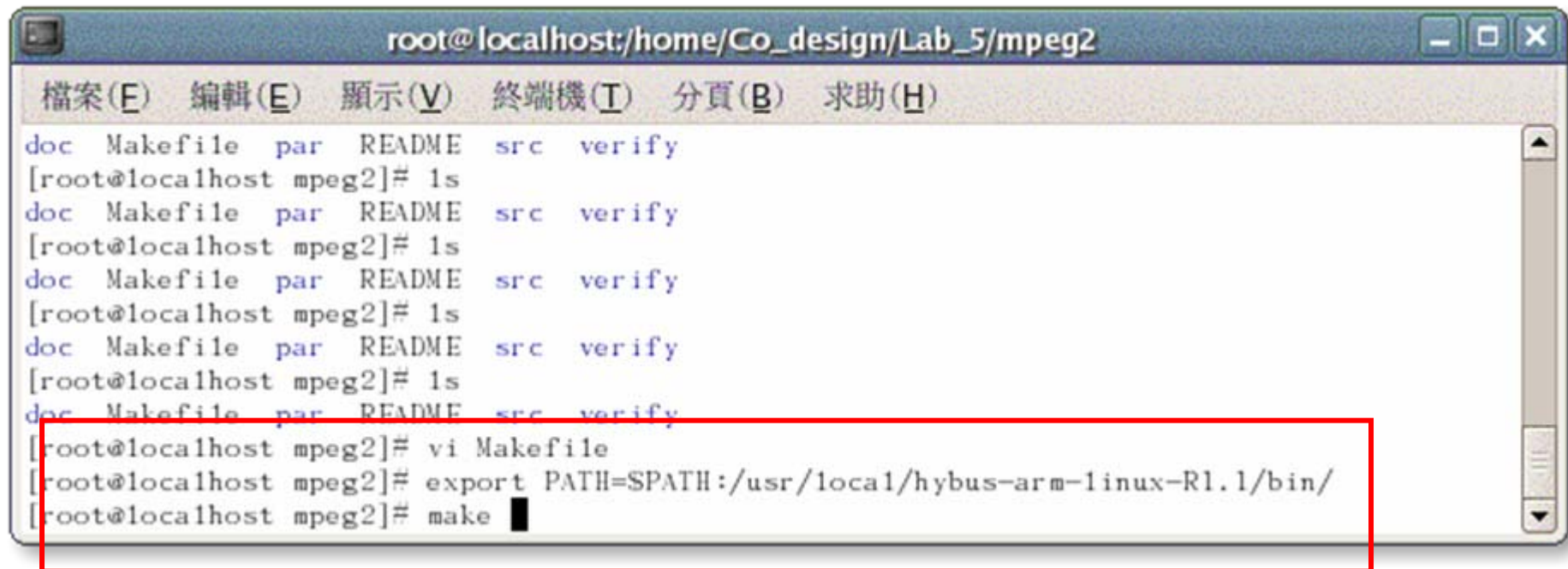
:wq
```

Press “Esc”

5.4 Compilation the MPEG-2 Source Code

■ Modify Makefile with Method 2

- ◆ #mpeg2>export PATH=\$PATH:/usr/local/hybus-arm-linux-R1.1/bin/
- ◆ #mpeg2>make
- ◆ Compile the Source Code



```
root@localhost:/home/Co_design/Lab_5/mpeg2
檔案(E) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)
doc Makefile par README src verify
[root@localhost mpeg2]# ls
doc Makefile par README src verify
[root@localhost mpeg2]# ls
doc Makefile par README src verify
[root@localhost mpeg2]# ls
doc Makefile par README src verify
[root@localhost mpeg2]# ls
doc Makefile par README src verify
[root@localhost mpeg2]# ls
doc Makefile par README src verify
[root@localhost mpeg2]# vi Makefile
[root@localhost mpeg2]# export PATH=$PATH:/usr/local/hybus-arm-linux-R1.1/bin/
[root@localhost mpeg2]# make
```

5.4 Compilation the MPEG-2 Source Code

■ Modify Makefile with Method 2

◆ Get the MPEG-2 decoder file

■ #mpeg2>cd src/mpeg2dec/

■ #mpeg2>ls

■ Get **mpeg2decode**

```
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/Co design/Lab_5/mpeg2/src/mpeg2enc'
[root@localhost mpeg2]# cd src/mpeg2dec/
[root@localhost mpeg2dec]# ls
CHANGES          getpic.o          mpeg2dec.h        spatscal.c
config.h          getvlc.c          mpeg2dec.o        spatscal.o
display.c         getvlc.h          mpeg2decode       store.c
display.o         getvlc.o          mpeg2decode_0402_2ip.out store.c.bak
EXAMPLES         global.h          mpeg2decode0402.out store.o
fpga_board_ioctl.h idct.c            mpeg2decode0430   subspic.c
fpga_reg_set.h   idct.c.bak       mpeg2decode0430-4000 subspic.o
getbits.c        idct.o           mpeg2decode1s     systems.c
getbits.o        idctref.c        mpeg2decode3slow  systems.o
getblk.c         idctref.o        mpeg2decodef      TODO
getblk.o         IEEE1180         mpeg2decodesofe   verify.c
gethdr.c         Makefile         README            verify.o
gethdr.o         motion.c         recon.c           yuv2rgb.c
getpic.c         motion.o         recon.o           yuv2rgb.o
getpic.c.bak     mpeg2dec.c       SPATIAL.DOC
[root@localhost mpeg2dec]#
```

5.5 Programming IDCT SIP

5.1 MPEG-2 Decoder HW & SW Integration Flow

5.2 Download the Linux Kernel

5.3 IDCT Driver Programming

5.4 Compilation the MPEG-2 Source Code

5.5 Programming IDCT SIP

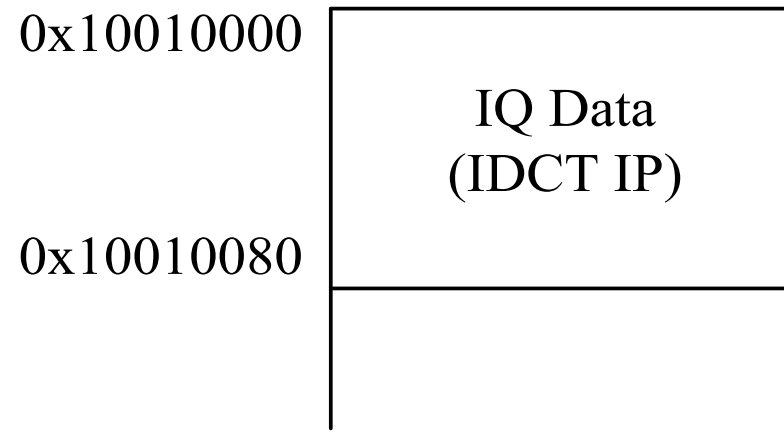
5.6 System Integrations

5.7 Software Verification and LA Measurement

5.5 Programming IDCT SIP

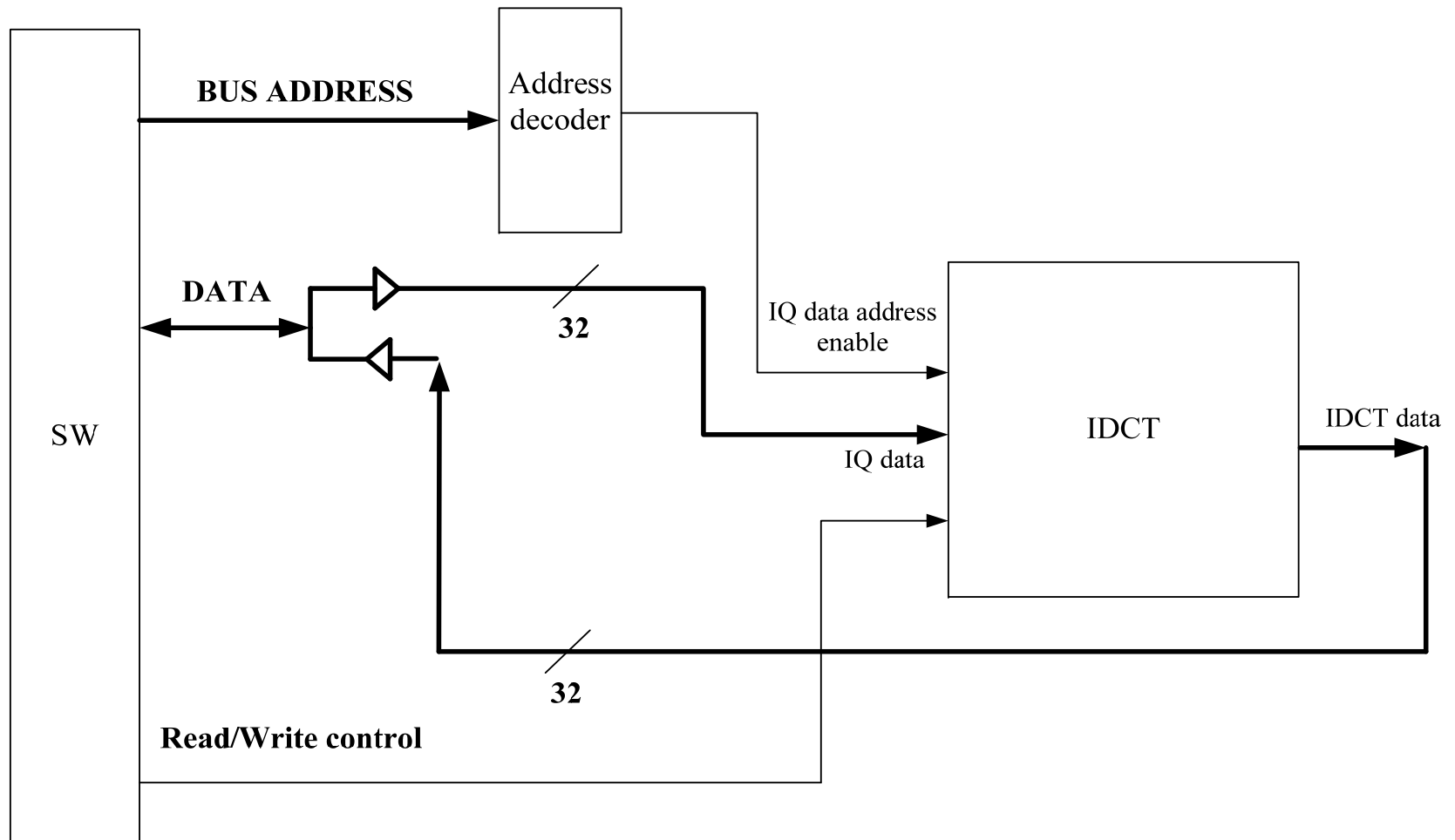
■ System Memory Map

◆ System Memory Map



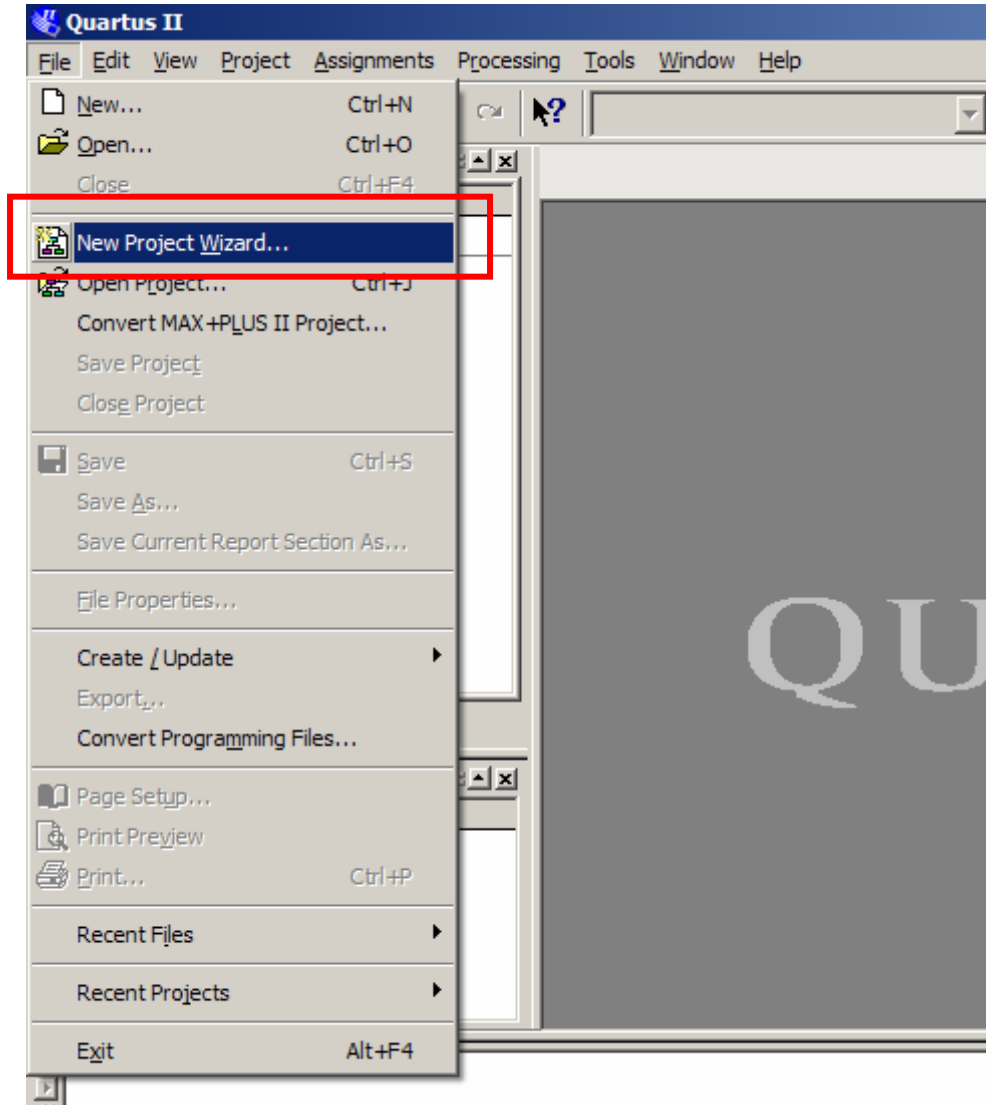
5.5 Programming IDCT SIP

■ MPEG-2 HW System



5.5 Programming IDCT SIP

◆ Step 1. File/New/New Project Wizard



5.5 Programming IDCT SIP

- ◆ Step 2.Type the Name for this Project. And type “Next>”.

New Project Wizard: Directory, Name, Top-Level Entity [page 1 of 5]

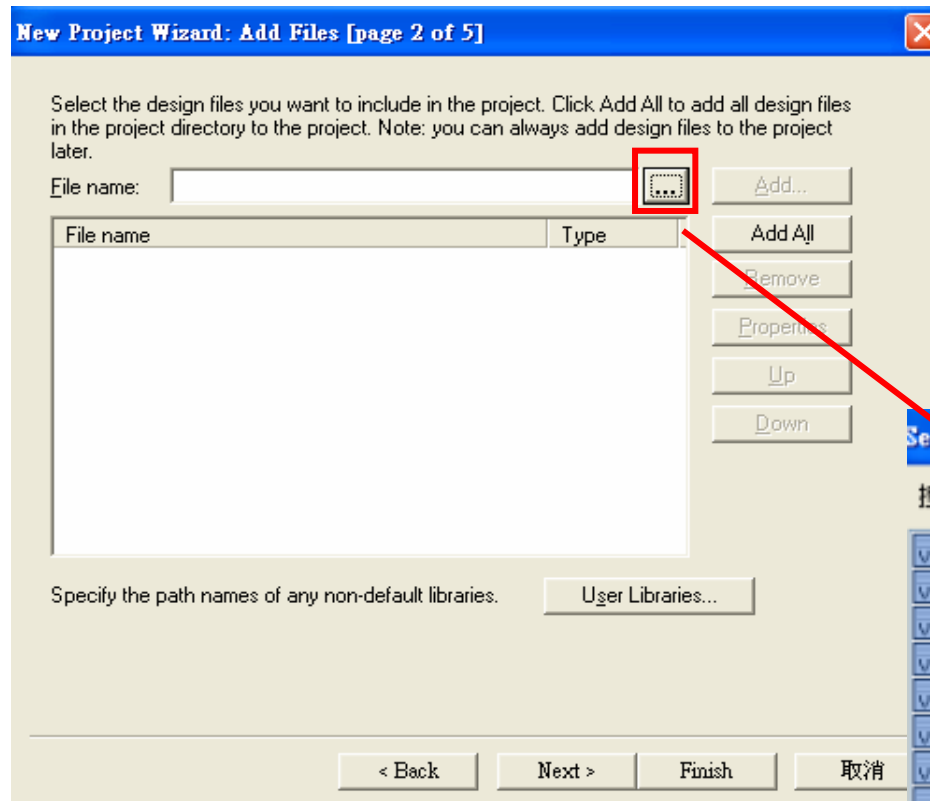
What is the working directory for this project?
 The directory for this project.

What is the name of this project?
 Type the name for this project.

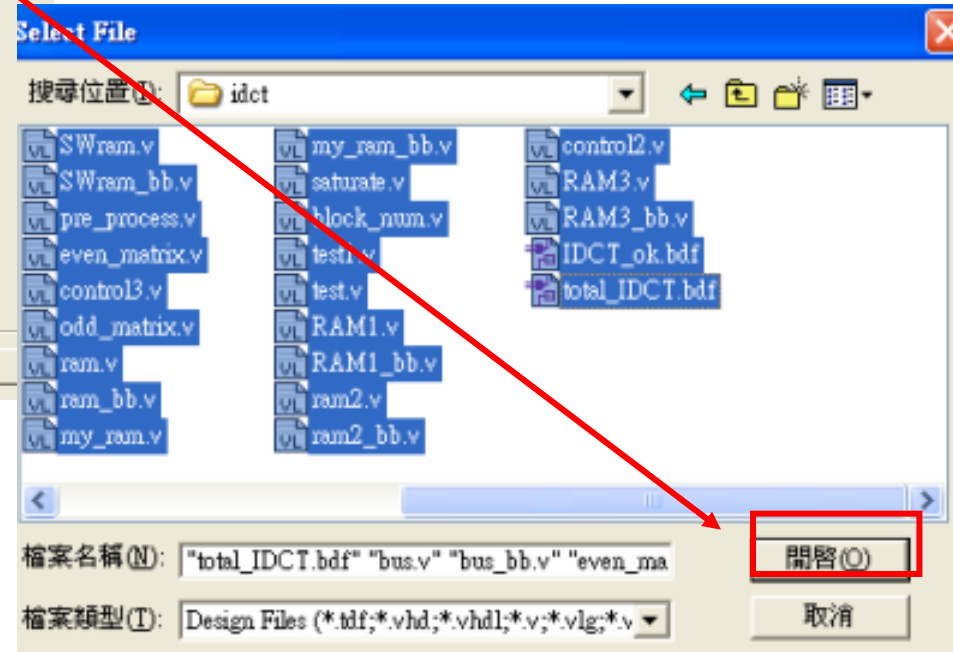
What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.
 Type the name for this project.

5.5 Programming IDCT SIP

◆ Step 3. Add the Previous IP into this Project.

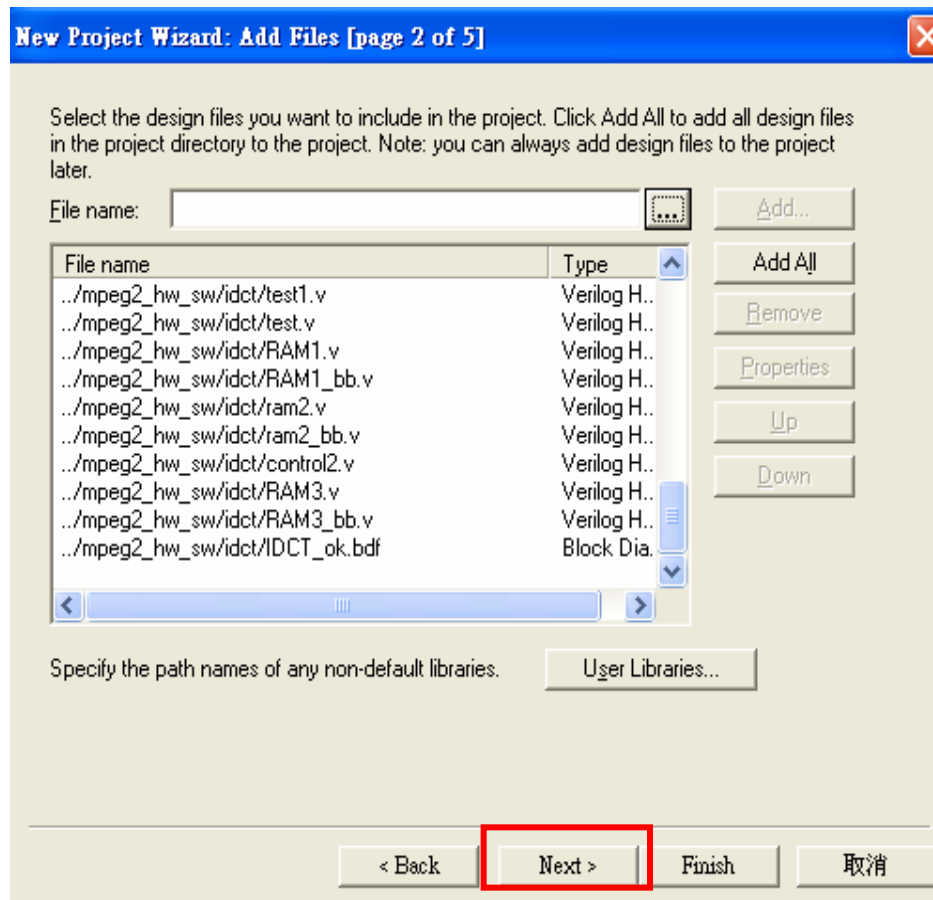


Click , select the files of the previous IP into this project.



5.5 Programming IDCT SIP

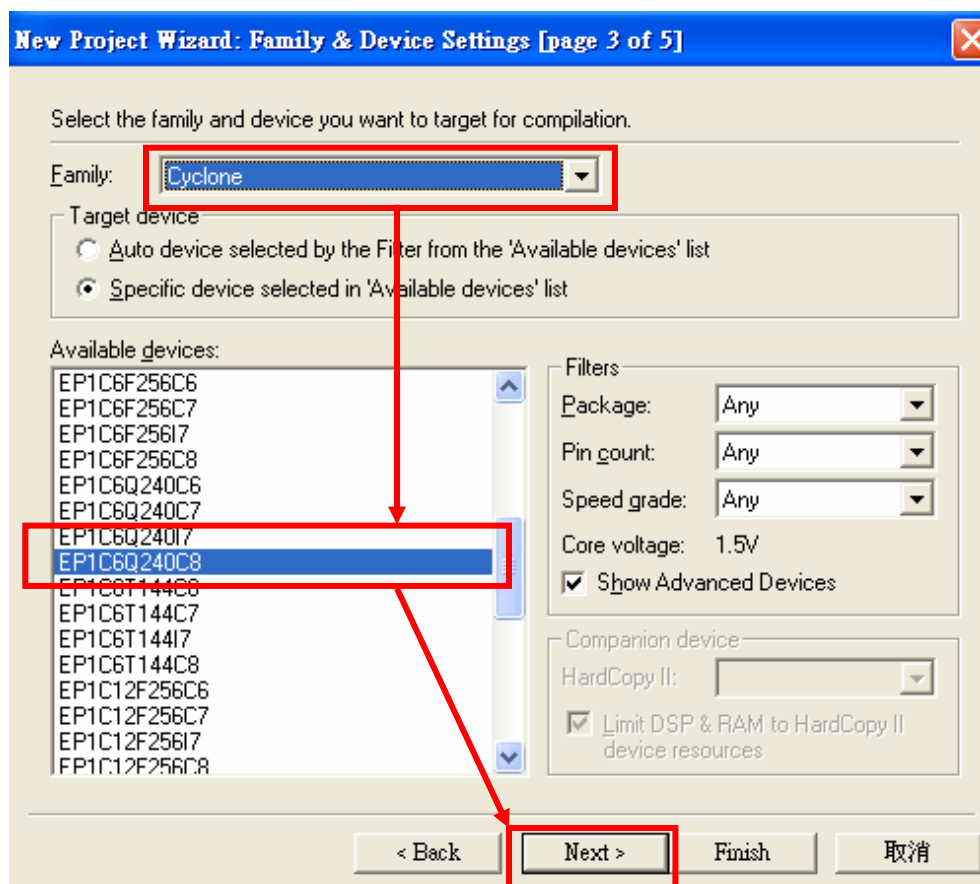
◆ Step 4. The Selected Files add into this Project.



5.5 Programming IDCT SIP

◆ Step 5. Select Family as “**Cyclone**”.

Select Available Devices as “**EP1C6Q240C8**”



5.5 Programming IDCT SIP

◆ Step 6. Click “Next>” to Continue.

New Project Wizard: EDA Tool Settings [page 4 of 5]

Specify the other EDA tools -- in addition to the Quartus II software -- used with the project.

☐ EDA design entry / synthesis tool: Not available

☐ EDA simulation tool: Not available

☐ EDA timing analysis tool: Not available

< Back **Next >** Finish

New Project Wizard: Summary [page 5 of 5]

When you click Finish, the project will be created with the following settings:

Project directory:
H:/SOC_Lab2/MPEG2/mpeg2_hw_swtest/

Project name: mpeg2_hw_swtest

Top-level design entity: mpeg2_hw_swtest

Number of files added: 37

Number of user libraries added: 0

Device assignments:

Family name: Cyclone

Device: EP1C6Q240C8

EDA tools:

Design entry/synthesis: <None>

Simulation: <None>

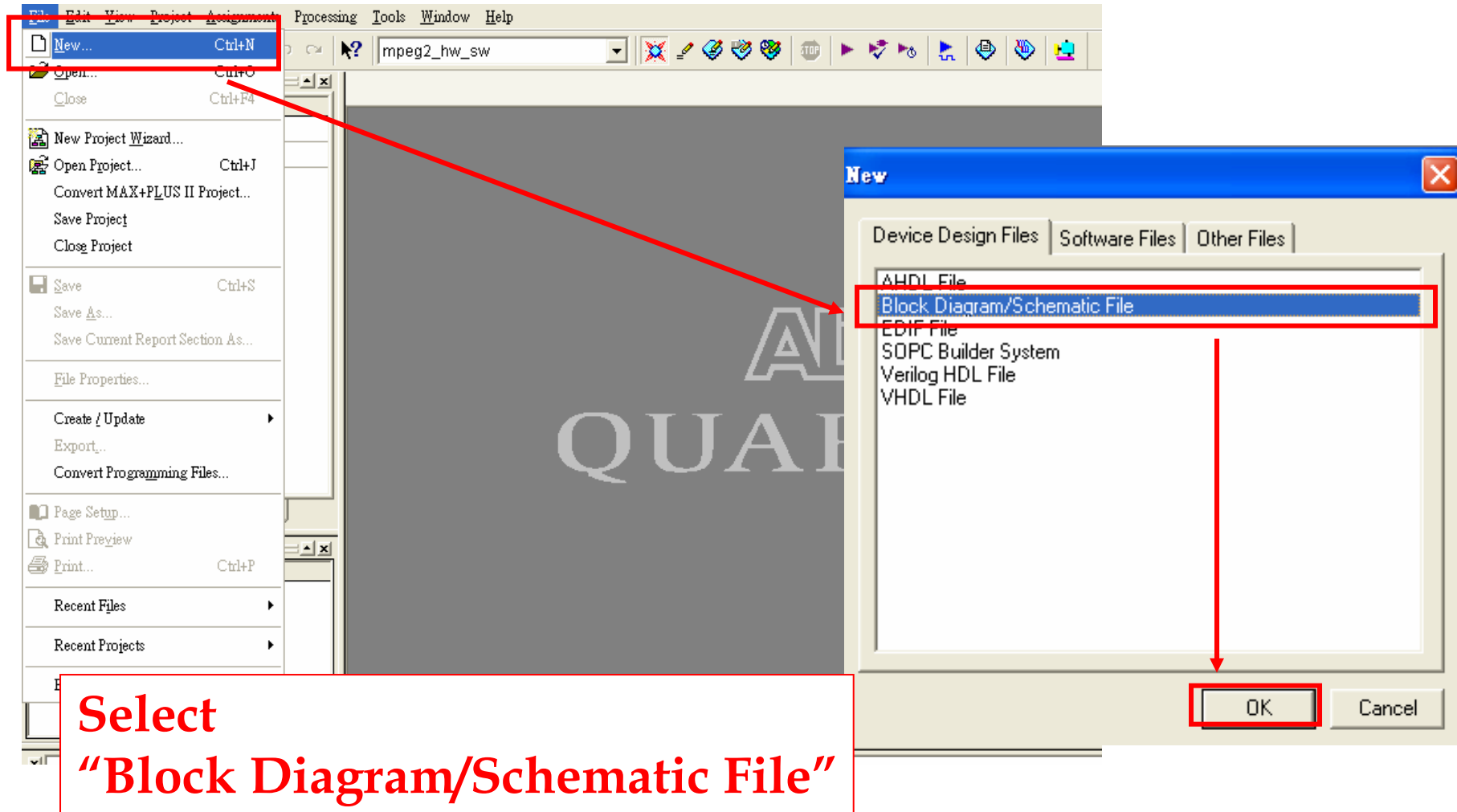
Timing analysis: <None>

< Back Next > **Finish** 取消

◆ Step 7. Click “Finish”. The New Project is Ready.

5.5 Programming IDCT SIP

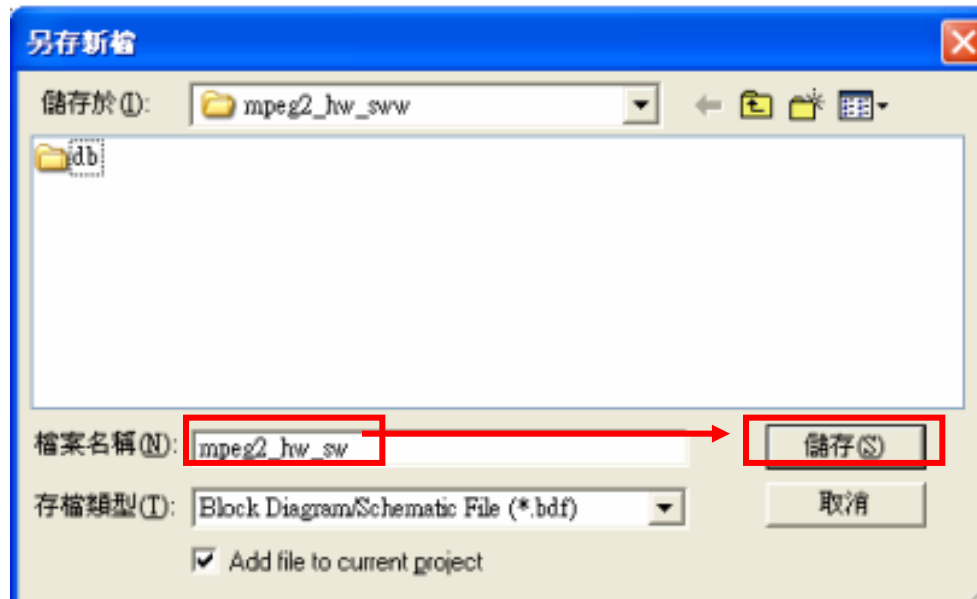
◆ Step 8. Open New “Block Diagram/Schematic File” “File/New”



5.5 Programming IDCT SIP

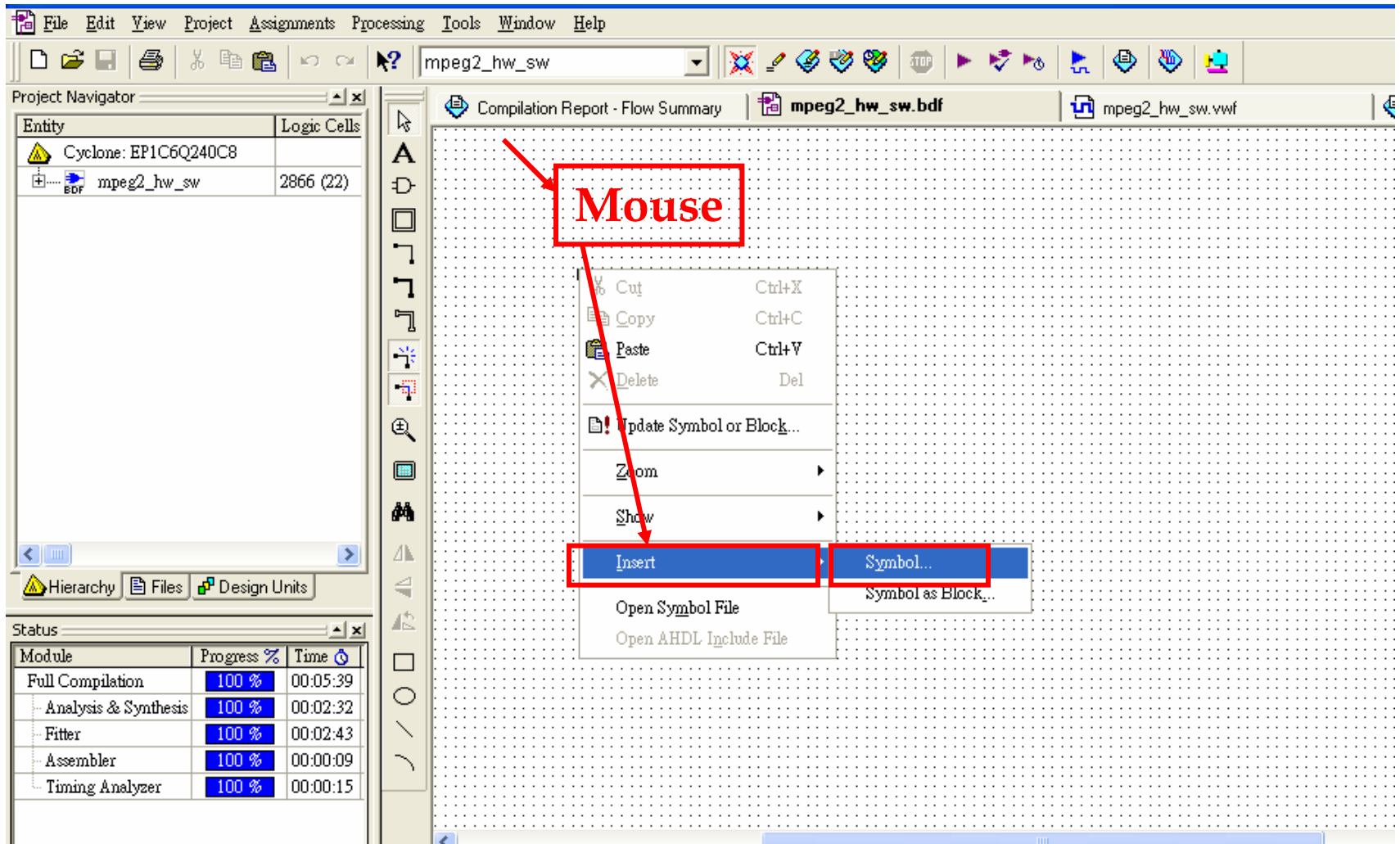
◆ Step 9. Save New File.

The Name of File is the Same as the Name of this Project.



5.5 Programming IDCT SIP

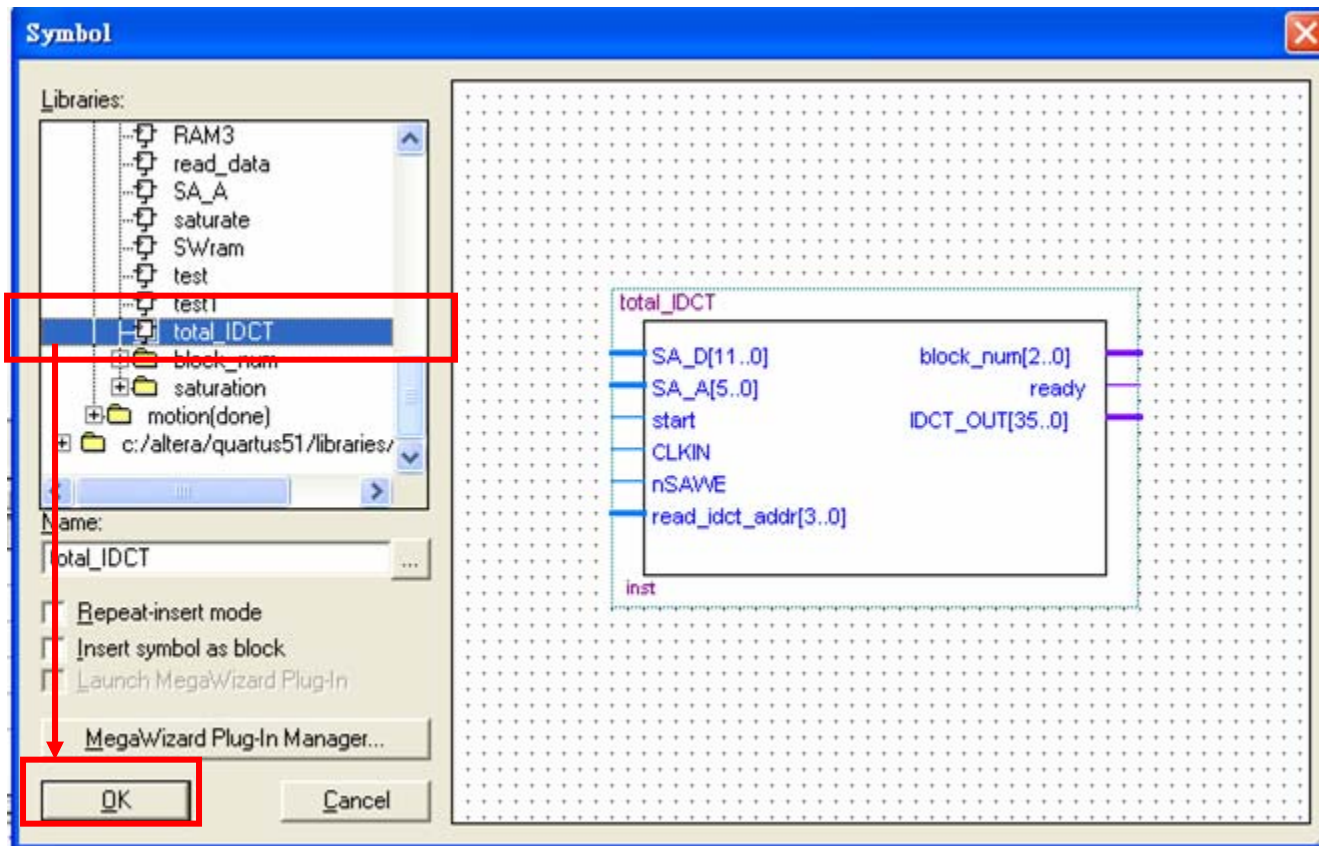
- ◆ Step 10. Click the Right Button of the Genius mouse.
“Insert/ Symbol...”



5.5 Programming IDCT SIP

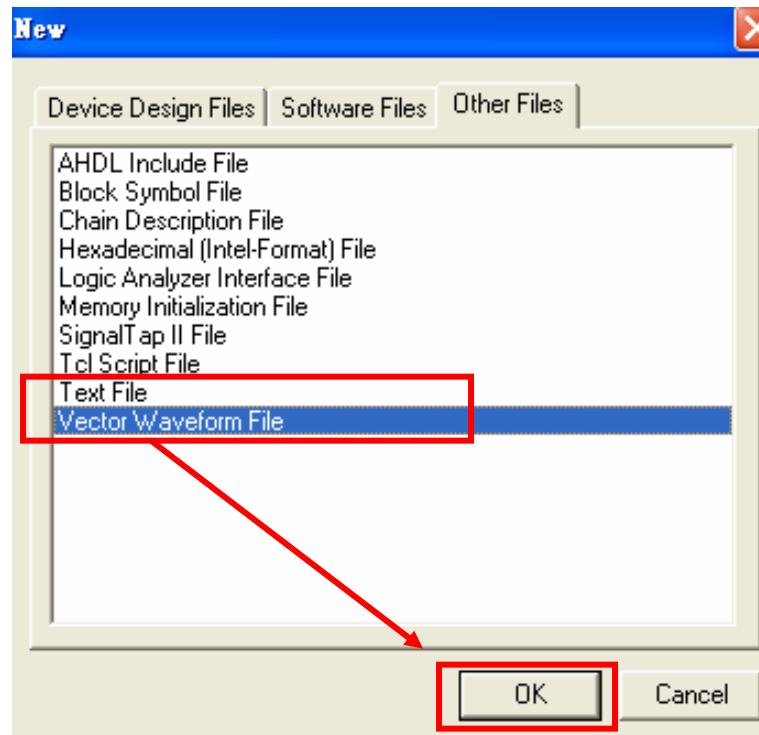
- ◆ Step 11. Select the Symbol of the Previous IPs.

Select the Top of IDCT Design : total IDCT



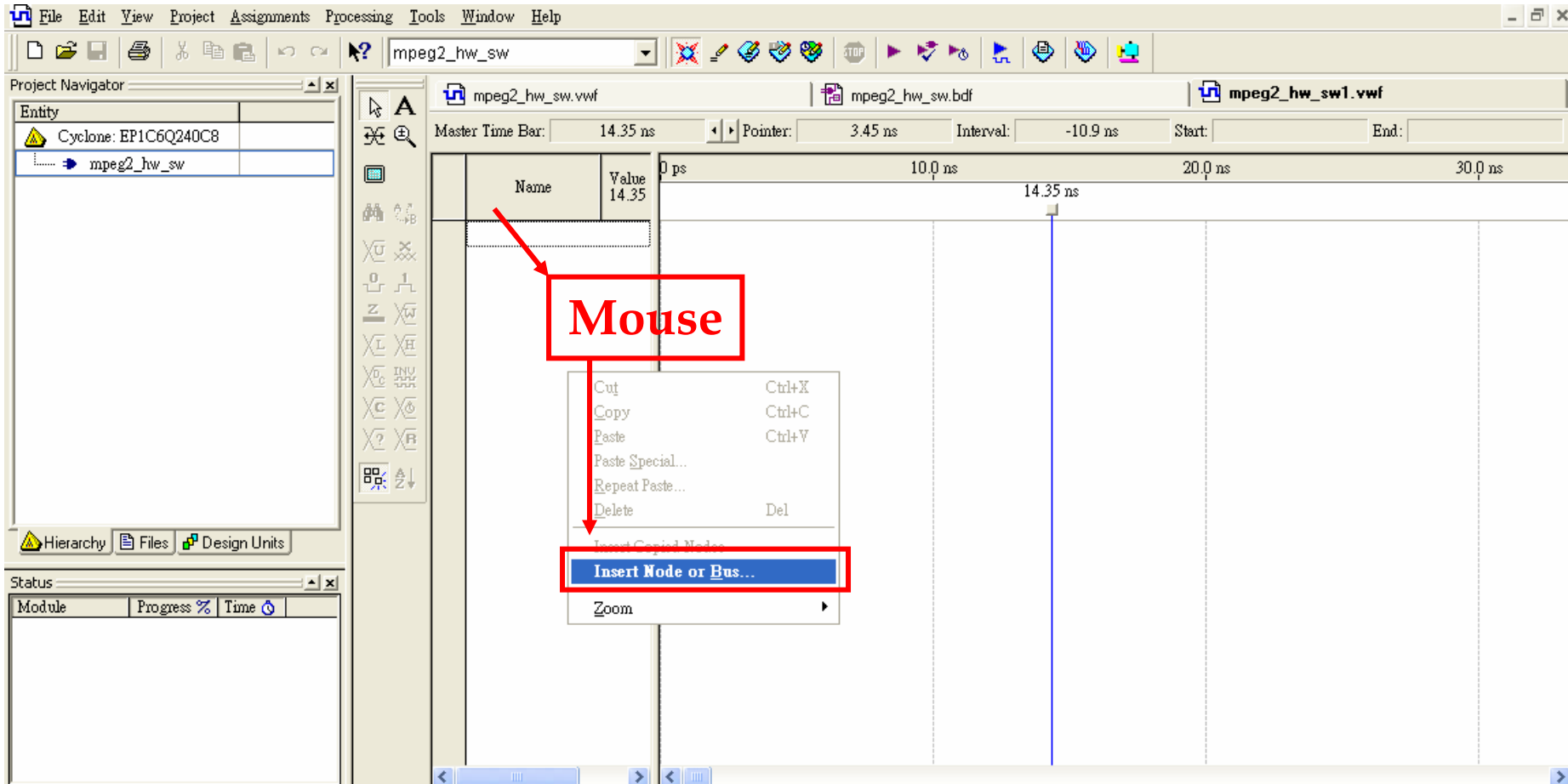
5.5 Programming IDCT SIP

- ◆ Step 12. Click "File/New" ,Open the "Vector Waveform File" to Simulate the Waveform.



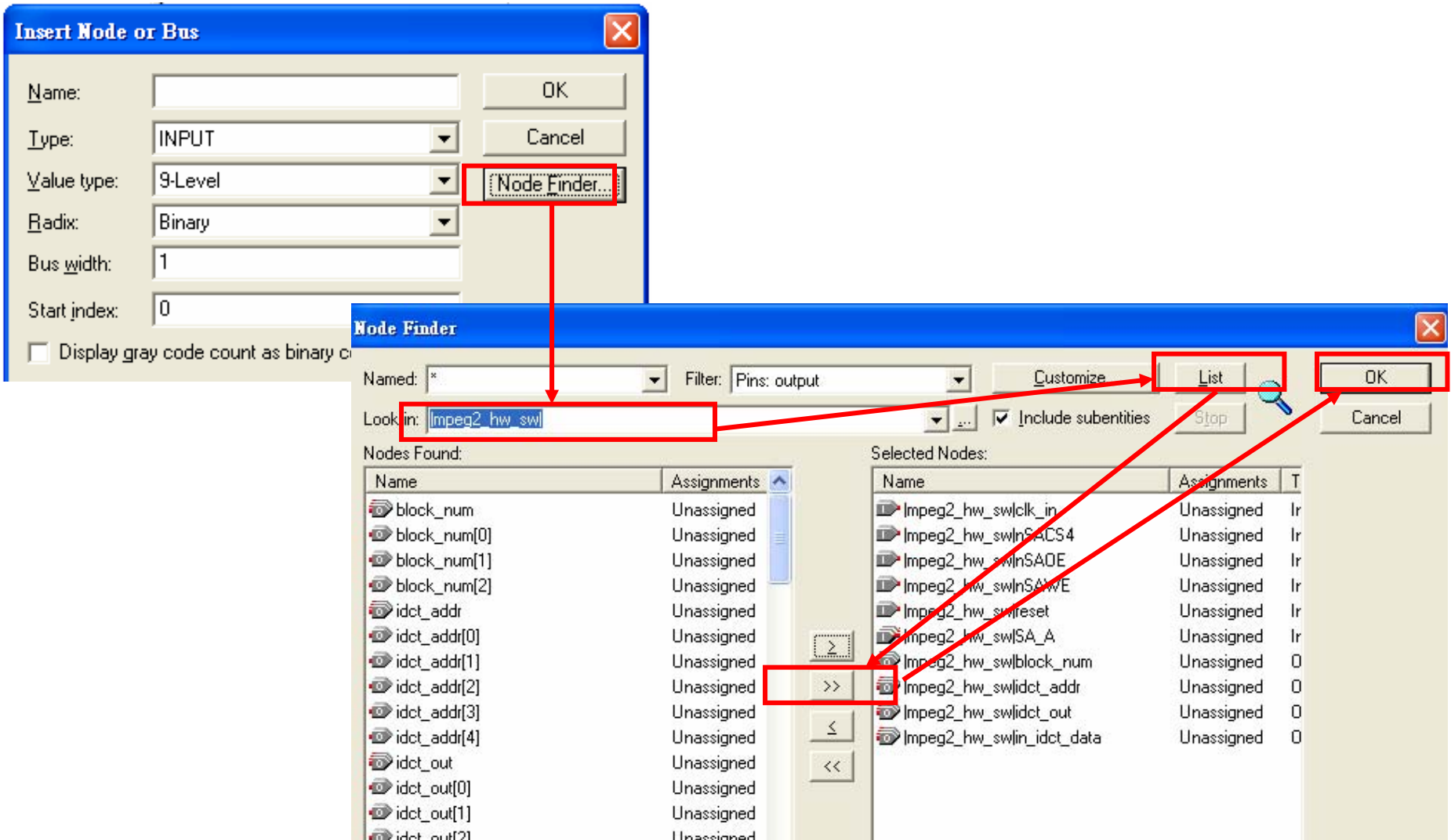
5.5 Programming IDCT SIP

- ◆ Step 13. Click the Right key of the Genius Mouse .
Click “Insert Nod or Bus...”



5.5 Programming IDCT SIP

- ◆ Step 14. Click “Node Finder”. Select the Inputs and Outputs to Simulate.



5.5 Programming IDCT SIP

◆ Step 15. Click the Button to Simulate.

The screenshot displays the Xilinx ISE software interface during a simulation. The top menu bar includes File, Edit, View, Project, Assignments, Processing, Tools, Window, and Help. The toolbar contains icons for file operations, simulation, and other functions. The 'Start Simulation' button, represented by a blue play icon, is highlighted with a red box and an arrow pointing to it.

The Project Navigator on the left shows the project hierarchy with 'Entity' and 'Logic Cells' tabs. The 'Entity' tab is active, showing 'Cyclone: EP1C6Q240C8' and 'mpeg2_hw_sw' (2866 (22)).

The main window displays the 'Compilation Report - Flow Summary' and 'mpeg2_hw_sw.bdf'. The 'Start Simulation' button is also visible in the top right corner. The waveform viewer shows a list of signals on the left, including 'nsACS4', 'nsAOE', 'nsAWE', 'clk_in', 'reset', 'SA_A', 'SA_D', 'idct_addr', 'idct_data', 'idct_out', 'in_idct_addr', 'iq_addr', 'pin_name', 'ready', and 'block_num'. The signals are plotted on a grid with time markers from 650.0 ns to 890.0 ns. The 'SA_D' signal is highlighted with a red box.

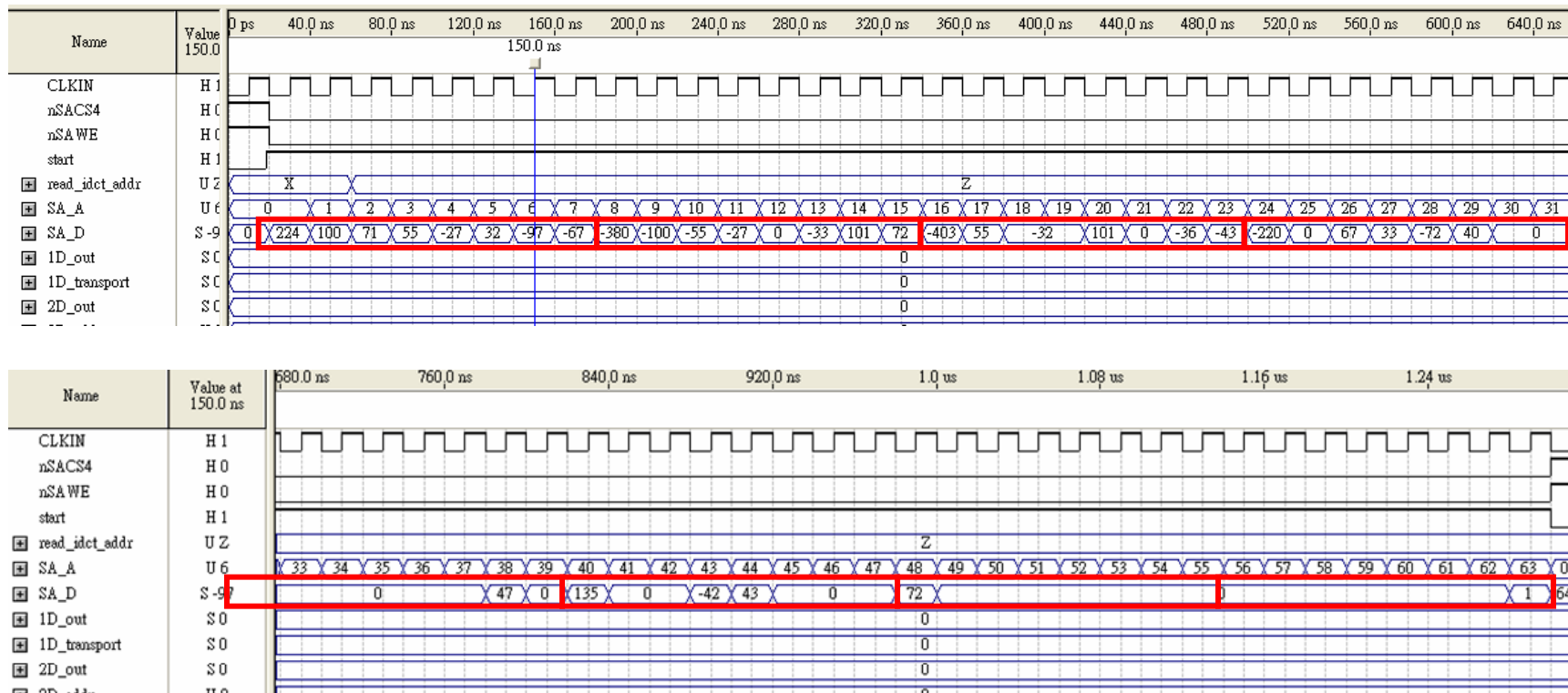
The Status window at the bottom left shows the 'Module' and 'Progress %' for the 'Simulator' module, which is at 58% progress and has a time of 00:00:45.

5.5 Programming IDCT SIP

■ Waveform simulation

◆ IP Operation Result

■ IQ_data

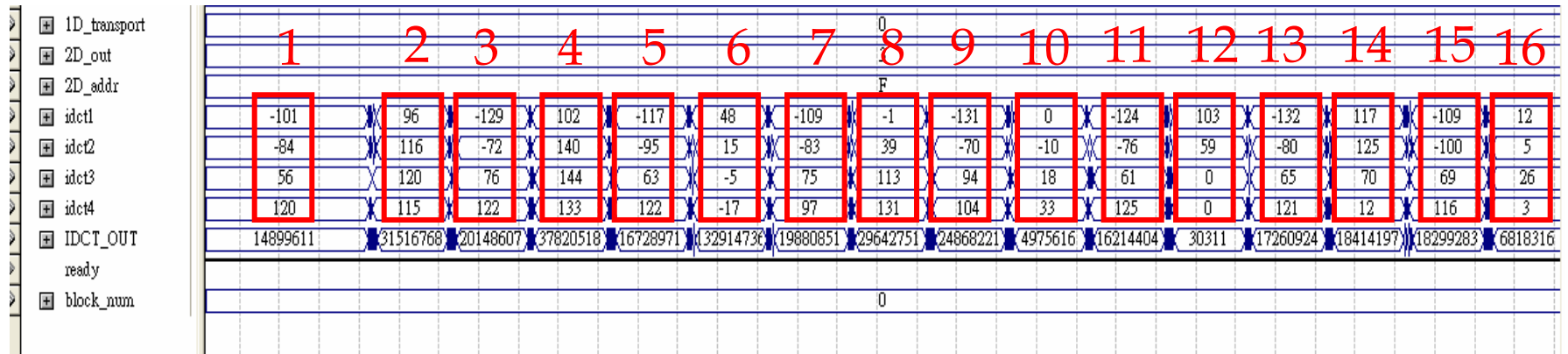


5.5 Programming IDCT SIP

■ Waveform simulation

◆ IP Operation Result

■ IDCT-IP



5.5 Programming IDCT SIP

■ Download IDCT hardware

◆ Hardware Design

- IDCT Hardware Programming File : idct.pof

◆ Download the IDCT Hardware

- Refer to Lab. 3

5.6 System Integrations

5.1 MPEG-2 Decoder HW & SW Integration Flow

5.2 Download the Linux Kernel

5.3 IDCT Driver Programming

5.4 Compilation the MPEG-2 Source Code

5.5 Programming IDCT SIP

5.6 System Integrations

5.7 Software Verification and LA Measurement

5.6 System Integration

- ◆ NFS environment
 - Mount Linux operation system to execute files
- ◆ Execute on PXA255 platform
 - Integrate the hardware and the software

5.6 System Integration

◆ Prepared design files:

■ Software design

- MPEG-2 decoder execute file : mpeg2decode
- Driver : idct_driver.o

■ Hardware

- IDCT hardware : idct.pof

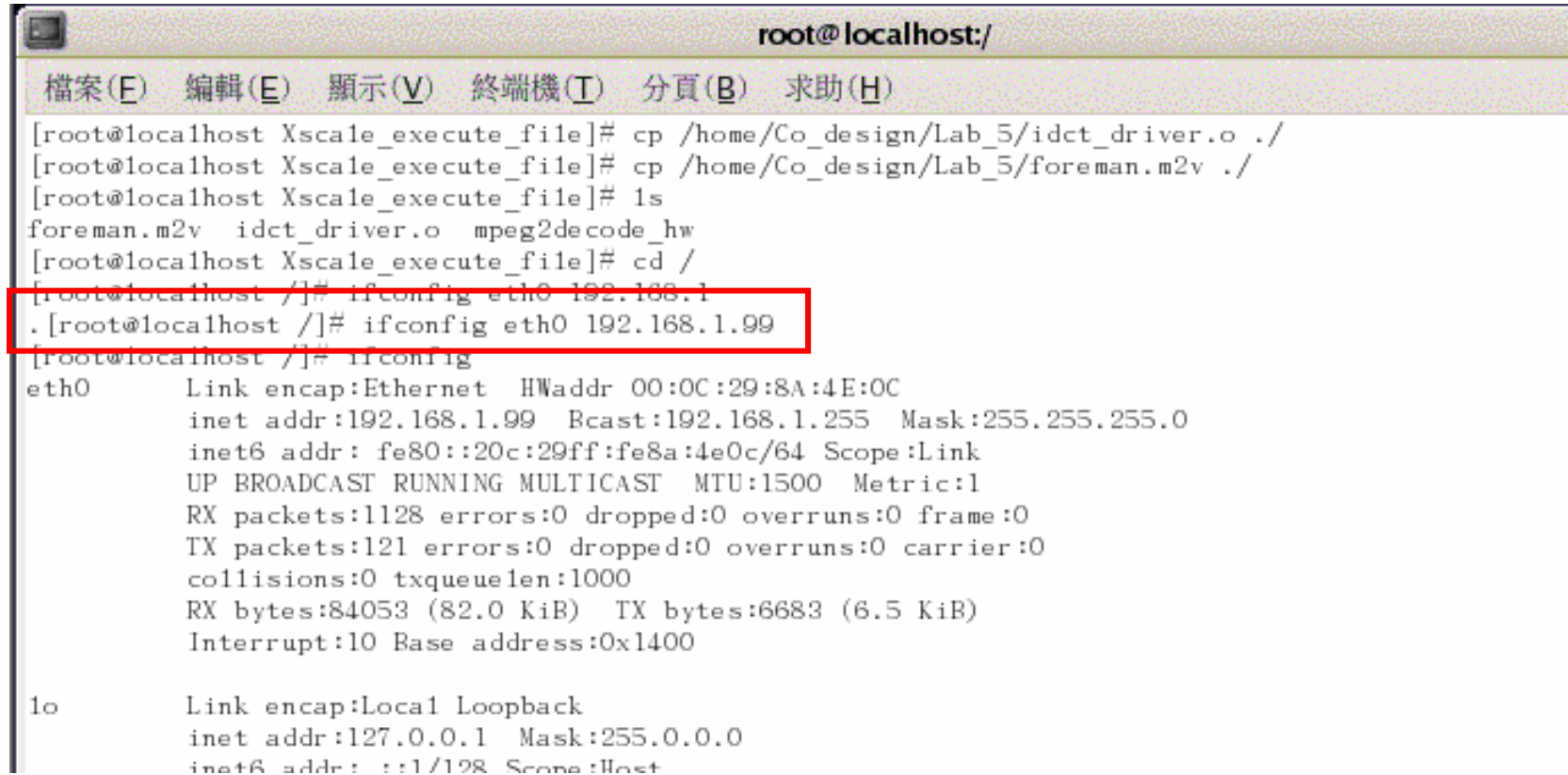
5.6 System Integration

■ NFS environment

◆ Step 1. Setup Network

■ #>ifconfig eth0 192.168.1.99

■ #>ifconfig



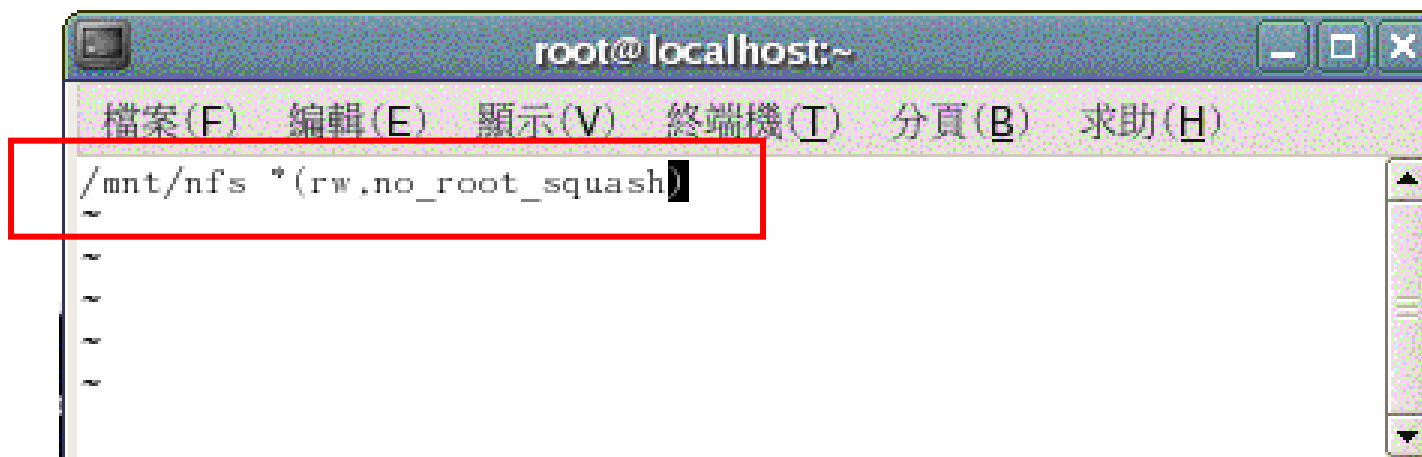
```
root@localhost:/  
檔案(E) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)  
[root@localhost Xscale_execute_file]# cp /home/Co_design/Lab_5/idct_driver.o ./  
[root@localhost Xscale_execute_file]# cp /home/Co_design/Lab_5/foreman.m2v ./  
[root@localhost Xscale_execute_file]# ls  
foreman.m2v idct_driver.o mpeg2decode_hw  
[root@localhost Xscale_execute_file]# cd /  
[root@localhost /]# ifconfig eth0 192.168.1.  
.[root@localhost /]# ifconfig eth0 192.168.1.99  
[root@localhost /]# ifconfig  
eth0      Link encap:Ethernet  HWaddr 00:0C:29:8A:4E:0C  
          inet addr:192.168.1.99  Bcast:192.168.1.255  Mask:255.255.255.0  
          inet6 addr: fe80::20c:29ff:fe8a:4e0c/64  Scope:Link  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:1128 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:121 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:1000  
          RX bytes:84053 (82.0 KiB)  TX bytes:6683 (6.5 KiB)  
          Interrupt:10 Base address:0x1400  
  
lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          inet6 addr: ::1/128  Scope:Host
```

5.6 System Integration

■ NFS environment

◆ Step 2. Setup NFS in VMware

- PATH:/mnt/nfs
- Create Folder : nfs, Xscale_execute_tile
- PATH:/mnt/nfs/Xscale_execute_tile
- #>vi /etc/export
- Add “/mnt/nfs *(rw,no_root_squash)” in the File



The screenshot shows a terminal window titled "root@localhost:~". The menu bar includes "檔案(F)", "編輯(E)", "顯示(V)", "終端機(T)", "分頁(B)", and "求助(H)". The command `/mnt/nfs *(rw,no_root_squash)` is entered and highlighted with a red rectangle. The terminal also shows several empty lines below the command.

```
root@localhost:~  
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)  
/mnt/nfs *(rw,no_root_squash)  
~  
~  
~  
~  
~
```

5.6 System Integration

■ NFS environment

◆ Step 3. Restart the NFS Service

■ #>/etc/rc.d/init.d/nfs restart

```
[root@localhost ~]# vi /etc/exports
[root@localhost ~]# /etc/rc.d/init.d/nfs restart
關閉 NFS mountd:
停止 NFS 系統程式:
關閉 NFS 磁碟配額:
關閉 NFS 服務:
啓動 NFS 服務:
啓動 NFS quotas:
啓動 NFS 系統程式:
啓動 NFS mountd:
[root@localhost ~]#
```

```
[ 確定 ]
[ 確定 ]
[ 確定 ]
[ 確定 ]
[ 確定 ]
[ 確定 ]
[ 確定 ]
[ 確定 ]
```

5.6 System Integration

■ NFS environment

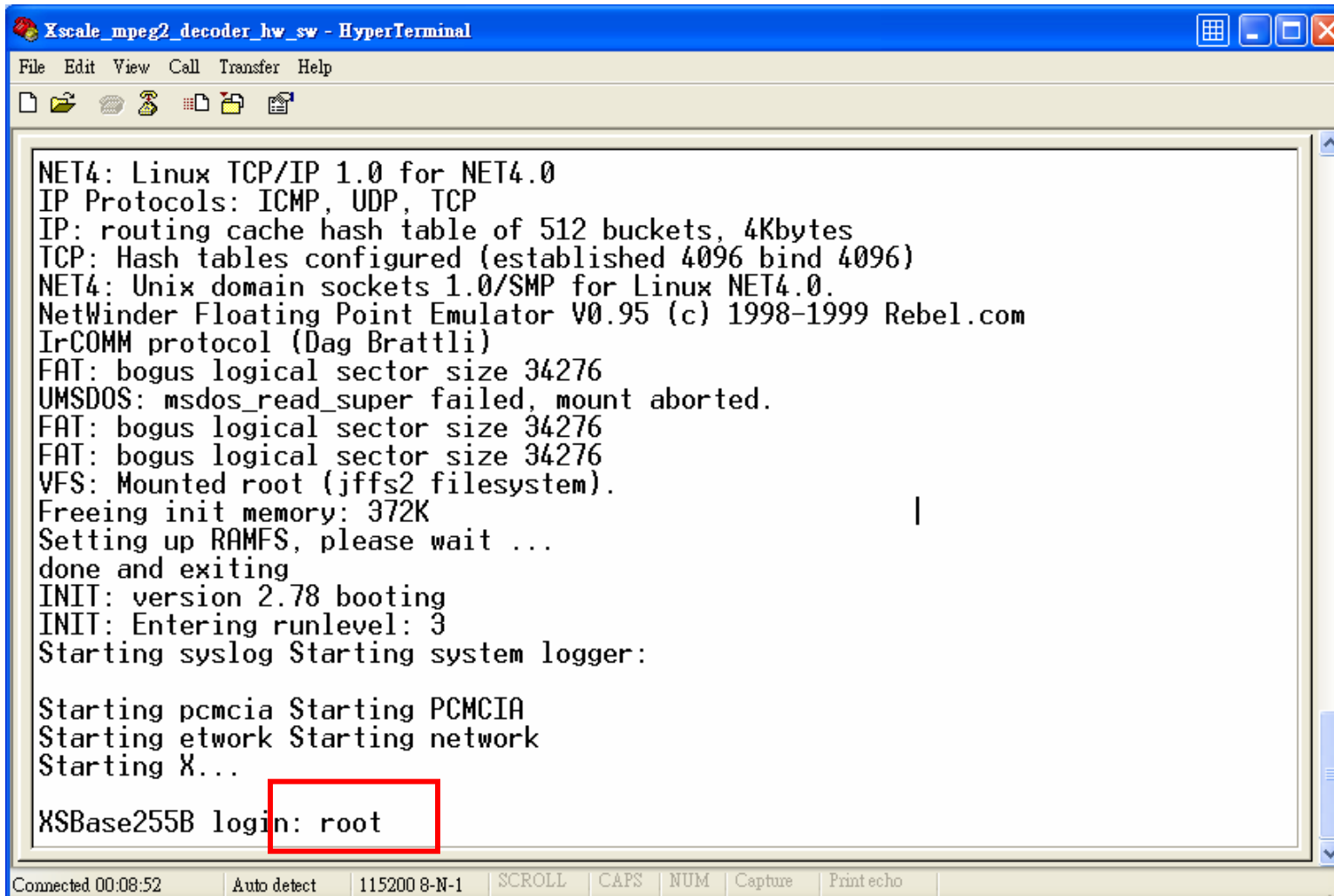
- ◆ Step 4. Copy “mpeg2decode_hw”, “idct_driver.o”, “foreman.m2v” into Xscale_execute_file folder
 - #>cd /mnt/nfs/Xscale_execute_file/
 - Xscale_execute_file>cp /home/Co_design_Lab5/mpeg2decode_hw ./
 - Xscale_execute_file>cp /home/Co_design_Lab5/idct_driver.o ./
 - Xscale_execute_file>cp /home/Co_design_Lab5/foreman.m2v ./

```
[root@localhost nfs]# cd Xscale_execute_file/
[root@localhost Xscale_execute_file]# ls
[root@localhost Xscale_execute_file]# cd /mnt/nfs/Xscale_execute_file/
[root@localhost Xscale_execute_file]# ls
[root@localhost Xscale_execute_file]# cp /home/Co_design/Lab_5/mpeg2decode_hw ./
[root@localhost Xscale_execute_file]# cp /home/Co_design/Lab_5/idct_driver.o ./
[root@localhost Xscale_execute_file]# cp /home/Co_design/Lab_5/foreman.m2v ./
[root@localhost Xscale_execute_file]# ls
foreman.m2v idct_driver.o mpeg2decode_hw
[root@localhost Xscale_execute_file]#
```

5.6 System Integration

■ Execute on PXA255 platform

◆ Step 1 : Power on the platform



```
Xscale_mpeg2_decoder_hw_sw - HyperTerminal
File Edit View Call Transfer Help

NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 4096 bind 4096)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
NetWinder Floating Point Emulator V0.95 (c) 1998-1999 Rebel.com
IrCOMM protocol (Dag Brattli)
FAT: bogus logical sector size 34276
UMSDOS: msdos_read_super failed, mount aborted.
FAT: bogus logical sector size 34276
FAT: bogus logical sector size 34276
VFS: Mounted root (jffs2 filesystem).
Freeing init memory: 372K
Setting up RAMFS, please wait ...
done and exiting
INIT: version 2.78 booting
INIT: Entering runlevel: 3
Starting syslog Starting system logger:

Starting pcmcia Starting PCMCIA
Starting etwork Starting network
Starting X...

XSBa255B login: root
```

5.6 System Integration

■ Execute on PXA255 platform

◆ Step 2 : Make the Device Node

■ #root>cd /dev

■ #root>mknod fpga_board_6 c 72 0

```
INIT: version 2.78 booting
INIT: Entering runlevel: 3
Starting syslog Starting system logger:
```

```
Starting pcmcia Starting PCMCIA
Starting etwork Starting network
Starting X...
```

```
XSBBase255B login: root
```

```
[root@XSBBase255B /root]$cd /dev
```

```
[root@XSBBase255B /dev]$mknod fpga_board_6 c 72 0
```

```
[root@XSBBase255B /dev]$ls -al fpga_board_6
```

```
crw-r--r--  1 root  root    72,  0 Feb 29 15:58 fpga board 6
```

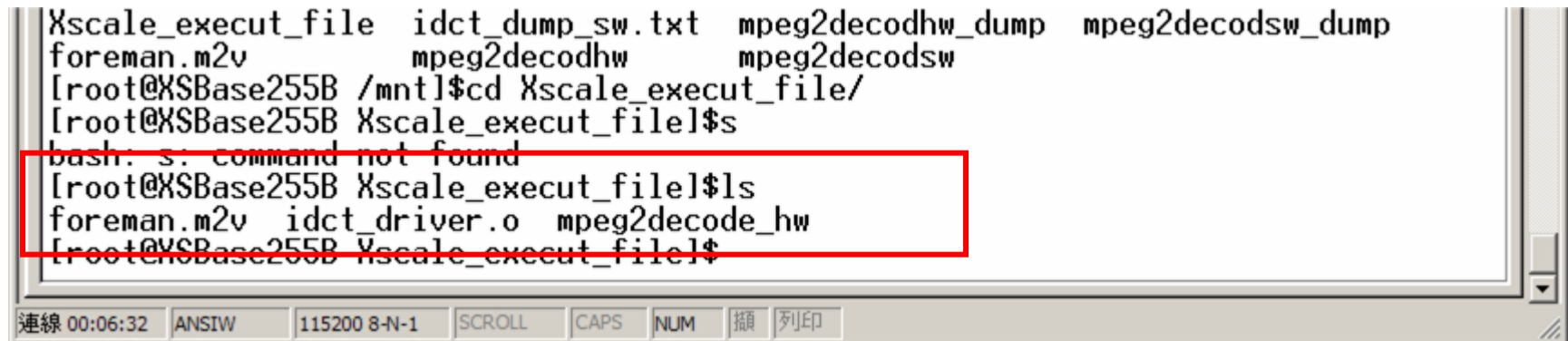
```
[root@XSBBase255B /dev]$_
```

5.6 System Integration

■ Execute on PXA255 platform

◆ Step 3. Mount the NFS service

- #root>ifconfig eth0 192.168.1.30
- #root>mount 192.168.1.99:/mnt/nfs /mnt
- #root>cd /mnt
- #mnt>cd Xscale_execute_file/
- #Xscale_execute_file>ls

A terminal window screenshot showing the execution of commands on a PXA255 platform. The terminal output shows the successful mounting of an NFS share and the listing of files in the mounted directory. A red rectangle highlights the 'ls' command and its output.

```
Xscale_execut_file  idct_dump_sw.txt  mpeg2decodhw_dump  mpeg2decodsw_dump
foreman.m2v         mpeg2decodhw       mpeg2decodsw
[root@XSBase255B /mnt]$cd Xscale_execut_file/
[root@XSBase255B Xscale_execut_file]$s
bash: s: command not found
[root@XSBase255B Xscale_execut_file]$ls
foreman.m2v  idct_driver.o  mpeg2decode_hw
[root@XSBase255B Xscale_execut_file]$
```

連線 00:06:32 ANSIW 115200 8-N-1 SCROLL CAPS NUM 擷 列印

Get idct_driver.o (driver), mpeg2decode_hw ,foreman.m2v

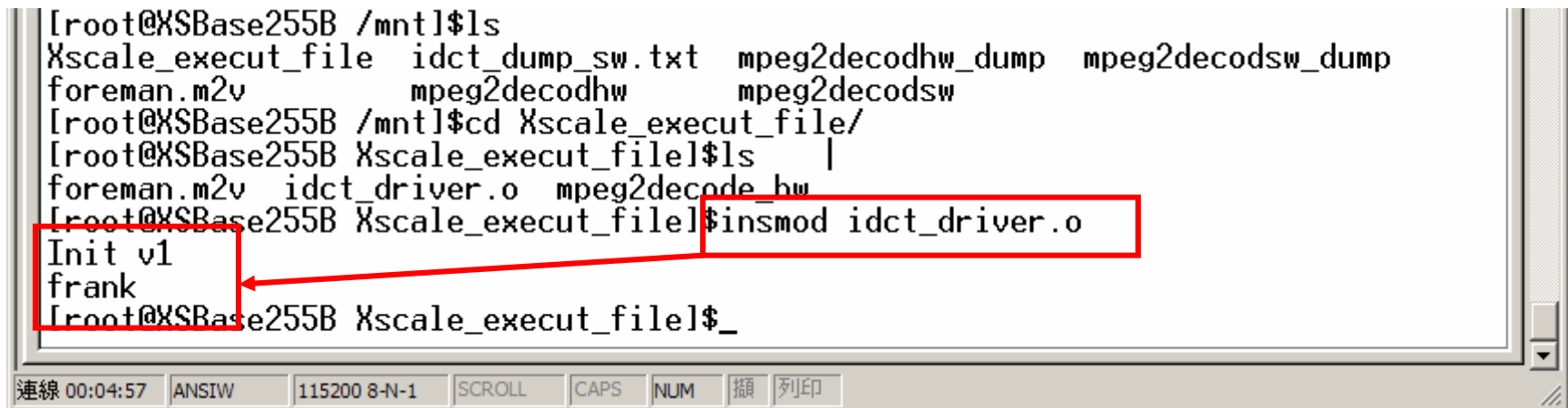
5.6 System Integration

■ Execute on PXA255 platform

◆ Step 4 : Insmod the driver

■ #Xscale_execute_file> insmod idct_driver.o

```
[root@XSBASE255B /mnt]$ls
Xscale_execut_file  idct_dump_sw.txt  mpeg2decodhw_dump  mpeg2decodsw_dump
foreman.m2v         mpeg2decodhw         mpeg2decodsw
[root@XSBASE255B /mnt]$cd Xscale_execut_file/
[root@XSBASE255B Xscale_execut_file]$ls
foreman.m2v  idct_driver.o  mpeg2decode_hw
[root@XSBASE255B Xscale_execut_file]$insmod idct_driver.o
Init v1
frank
[root@XSBASE255B Xscale_execut_file]$_
```



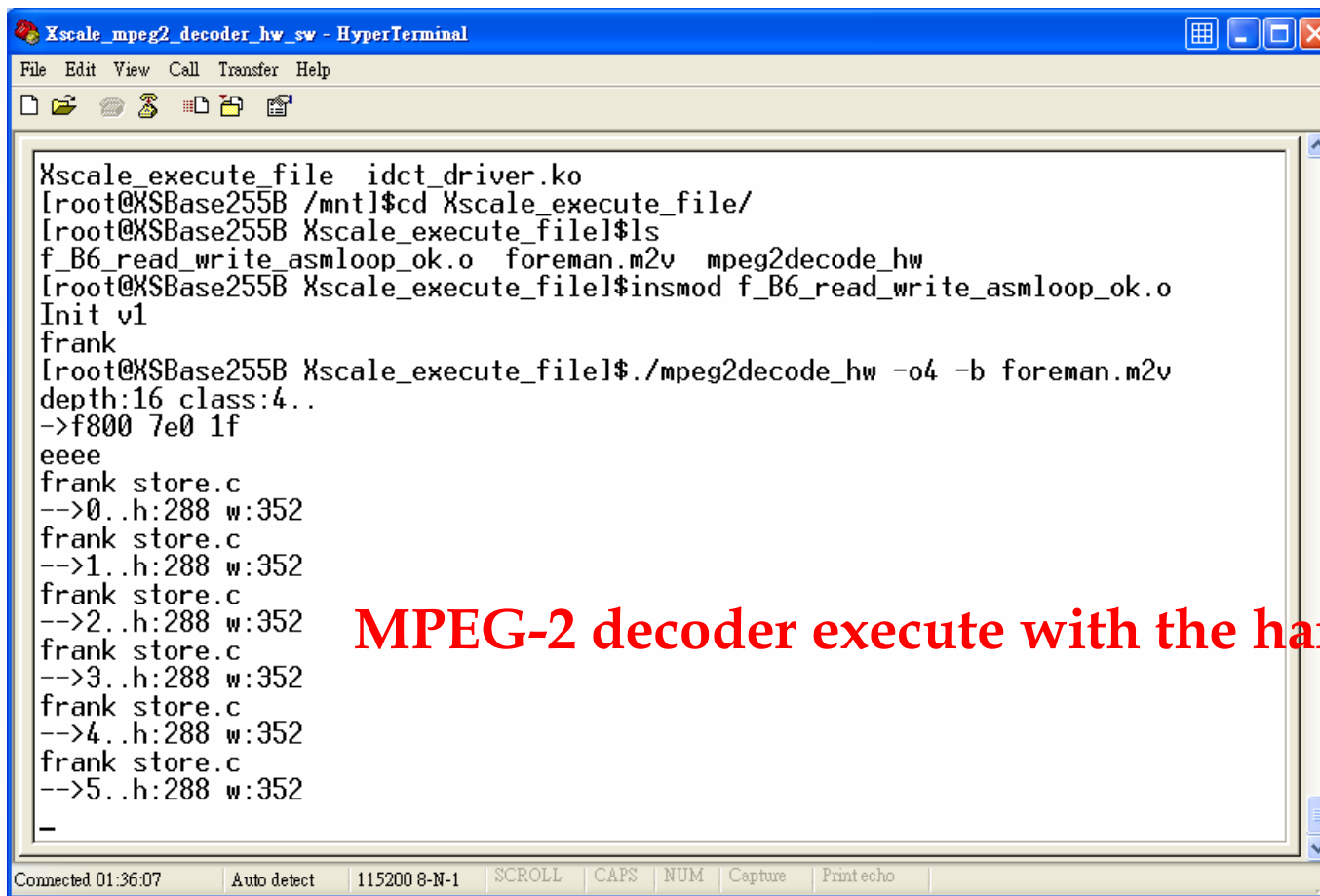
Insmode driver success!!

5.6 System Integration

■ Execute on PXA255 platform

◆ Step 5 : execute the MPEG-2 decoder file

■ #Xscale_execute_file>./mpeg2decode_hw -o4 -b foreman.m2v



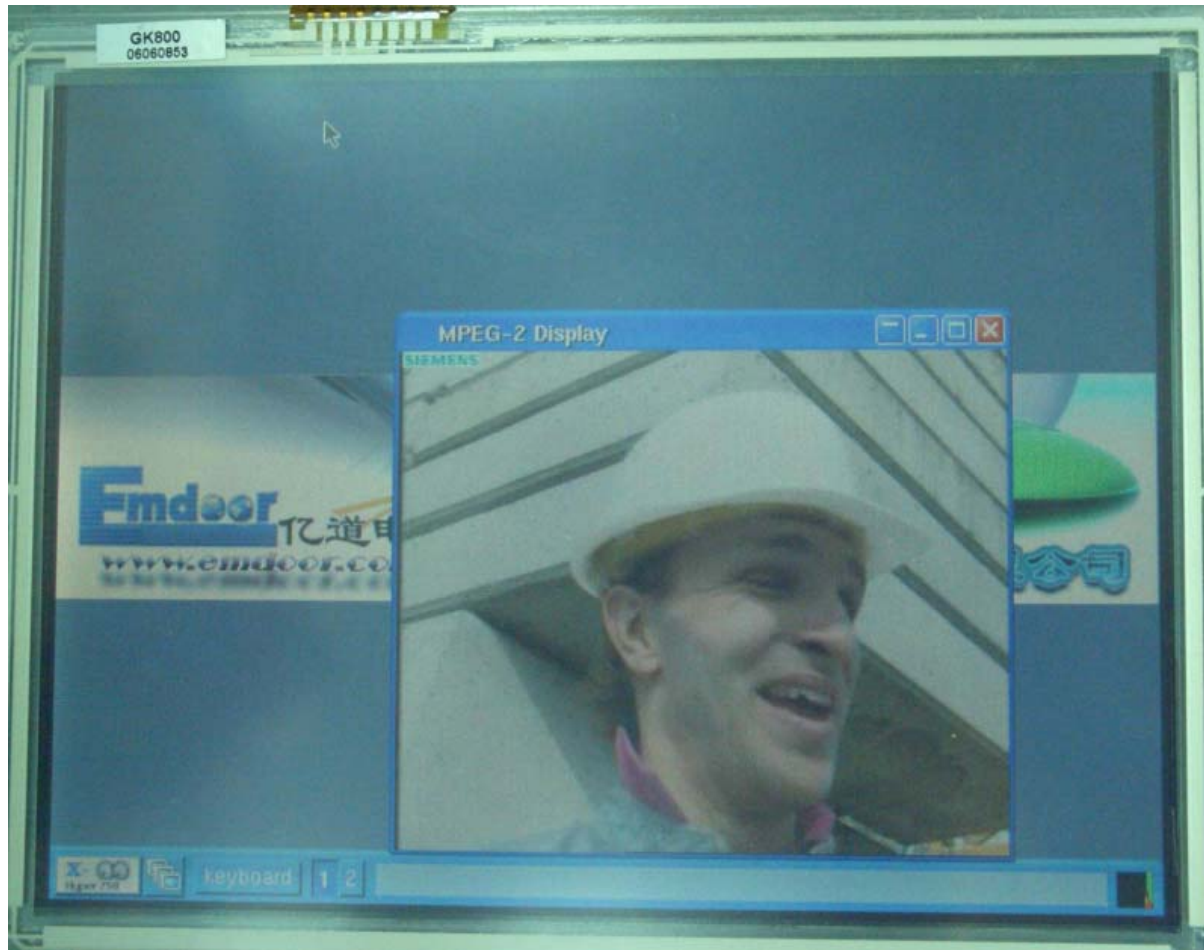
```
Xscale_mpeg2_decoder_hw_sw - HyperTerminal
File Edit View Call Transfer Help
Xscale_execute_file idct_driver.ko
[root@XSBASE255B /mnt1]$cd Xscale_execute_file/
[root@XSBASE255B Xscale_execute_file]$ls
f_B6_read_write_asmlong_ok.o foreman.m2v mpeg2decode_hw
[root@XSBASE255B Xscale_execute_file]$insmod f_B6_read_write_asmlong_ok.o
Init v1
frank
[root@XSBASE255B Xscale_execute_file]$./mpeg2decode_hw -o4 -b foreman.m2v
depth:16 class:4..
->f800 7e0 1f
eeee
frank store.c
-->0..h:288 w:352
frank store.c
-->1..h:288 w:352
frank store.c
-->2..h:288 w:352
frank store.c
-->3..h:288 w:352
frank store.c
-->4..h:288 w:352
frank store.c
-->5..h:288 w:352
-
```

MPEG-2 decoder execute with the hardware.

5.6 System Integration

■ Execute on PXA255 platform

◆ Display on TFT LCD



5.7 Software Verification and LA Measurement

5.1 MPEG-2 Decoder HW & SW Integration Flow

5.2 Download the Linux Kernel

5.3 IDCT Driver Programming

5.4 Compilation the MPEG-2 Source Code

5.5 Programming IDCT SIP

5.6 System Integrations

5.7 Software Verification and LA Measurement

5.7 Software Verification and LA Measurement

◆ Software Verification

- Modify Software to Dump IDCT Data
- Dump IDCT Data with Software Version
- Dump IDCT Data with SW/HW Version

◆ LA Measurement

- Use Logic Analyser
- Physical Waveform

5.7 Software Verification and LA Measurement

■ Modify software to dump IDCT data(1/3)

◆ getpic.c : mpeg2/src/mpeg2dec/getpic.c

◆ File Name : getpic.c

```
-----  
int idct_mb_count=0;      //MB count  
int idct_block_count;    //8x8 block count  
FILE *idct_dump_ptr;     //Dump file
```

```
static void motion_compensation(...)
```

```
...  
{  
    ...  
    int di;  
    int dj;  
    short *idct_dump;  
    ...  
}
```

```
-----
```

5.7 Software Verification and LA Measurement

■ Modify software to dump IDCT data(2/3)

◆ getpic.c : mpeg2/src/mpeg2dec/getpic.c

◆ File Name : getpic.c

```
-----  
static void motion_compensation(...)  
{  
    ...  
    idct_block_count=0; //Open the file to dump data  
    idct_dump_ptr=fopen("idct_dump_sw.txt","ab");  
    /* copy or add block data into picture */  
    for (comp=0; comp<block_count; comp++)  
    {  
        ...  
    }  
    idct_mb_count=idct_mb_count+1;  
    // printf("s"); //Make sure the function call  
    fclose(idct_dump_ptr);  
}  
-----
```

5.7 Software Verification and LA Measurement

■ Modify software to dump IDCT data(3/3)

◆ File Name : getpic.c (mpeg2/src/mpeg2dec/getpic.c)

```
-----  
for (comp=0; comp<block_count; comp++)  
{  
    Fast_IDCT(ld->block[comp]);  
    idct_dump=ld->block[comp];          //Get the IDCT data  
    fprintf(idct_dump_ptr,"idct_mb_count=%d",idct_mb_count);//MB number  
    fprintf(idct_dump_ptr,"idct_block_count=%d\n",idct_block_count);  
    for(dj=0;dj<8;dj++)  
    {  
        for(di=0;di<8;di++)  
        {  
            //Dump data into the output file  
            fprintf(idct_dump_ptr,"idct[%d]=%d ;",8*dj+di,idct_dump[8*dj+di]);  
        }  
        fprintf(idct_dump_ptr,"\n");  
    }  
    idct_block_count++;  
    fprintf(idct_dump_ptr,"\n\n\n");  
    Add_Block(comp,bx,by,dct_type,(macroblock_type & MACROBLOCK_INTRA)==0);  
}  
-----
```


5.7 Software Verification and LA Measurement

◆ Modify MPEG-2 Source Code

- Compile the MPEG-2 Source Code
- Get the File that Dump Data

```
[root@XSBase255B /mnt]$ls
Xscale_execut_file  idct_dump_sw.txt  mpeg2decodhw_dump  mpeg2decodsw_dump
foreman.m2v         mpeg2decodhw      mpeg2decodsw
[root@XSBase255B /mnt]$./mpeg2decodhw_dump -o4 -b foreman.m2v
```

連線 02:59:59 ANSIW 115200 8-N-1 SCROLL CAPS NUM 擷 列印

XSase255B>./mpeg2decodhw_dump -o4 -b foreman.m2v

```
[root@XSBase255B /mnt]$ls
Xscale_execut_file  idct_dump_hw.txt  mpeg2decodhw      mpeg2decodsw
foreman.m2v         idct_dump_sw.txt  mpeg2decodhw_dump  mpeg2decodsw_dump
[root@XSBase255B /mnt]$
```

連線 03:14:41 ANSIW 115200 8-N-1 SCROLL CAPS NUM 擷 列印

Get the dump file : idct_dump_hw.txt

5.7 Software Verification and LA Measurement

■ Dump IDCT data

◆ Dump the Data with the Software Version and Hardware Version

idct_dump_sw.txt

```
idct_mb_count=0 idct_block_count=2
idct[0]=-139 ;idct[1]=-68 ;idct[2]=61 ;idct[3]=134 ;idct[4]=120 ;idct[5]=115 ;idct[6]=126 ;idct[7]=109 ;
idct[8]=-124 ;idct[9]=-90 ;idct[10]=-29 ;idct[11]=32 ;idct[12]=58 ;idct[13]=36 ;idct[14]=8 ;idct[15]=-1 ;
idct[16]=-97 ;idct[17]=-76 ;idct[18]=-26 ;idct[19]=31 ;idct[20]=49 ;idct[21]=28 ;idct[22]=23 ;idct[23]=45 ;
idct[24]=-122 ;idct[25]=-105 ;idct[26]=-24 ;idct[27]=40 ;idct[28]=34 ;idct[29]=42 ;idct[30]=84 ;idct[31]=102 ;
idct[32]=-112 ;idct[33]=-102 ;idct[34]=-34 ;idct[35]=31 ;idct[36]=52 ;idct[37]=84 ;idct[38]=108 ;idct[39]=89 ;
idct[40]=-117 ;idct[41]=-87 ;idct[42]=-42 ;idct[43]=14 ;idct[44]=70 ;idct[45]=96 ;idct[46]=87 ;idct[47]=69 ;
idct[48]=-114 ;idct[49]=-84 ;idct[50]=-31 ;idct[51]=34 ;idct[52]=83 ;idct[53]=97 ;idct[54]=92 ;idct[55]=91 ;
idct[56]=-95 ;idct[57]=-107 ;idct[58]=-41 ;idct[59]=47 ;idct[60]=76 ;idct[61]=88 ;idct[62]=98 ;idct[63]=82 ;
```

idct_dump_hw.txt

```
idct_mb_count=0 idct_block_count=2
idct[0]=-142 ;idct[1]=-68 ;idct[2]=58 ;idct[3]=132 ;idct[4]=118 ;idct[5]=114 ;idct[6]=126 ;idct[7]=108 ;
idct[8]=-126 ;idct[9]=-90 ;idct[10]=-30 ;idct[11]=31 ;idct[12]=55 ;idct[13]=36 ;idct[14]=8 ;idct[15]=0 ;
idct[16]=-99 ;idct[17]=-76 ;idct[18]=-26 ;idct[19]=30 ;idct[20]=48 ;idct[21]=28 ;idct[22]=22 ;idct[23]=43 ;
idct[24]=-123 ;idct[25]=-105 ;idct[26]=-26 ;idct[27]=39 ;idct[28]=33 ;idct[29]=42 ;idct[30]=83 ;idct[31]=101 ;
idct[32]=-113 ;idct[33]=-101 ;idct[34]=-35 ;idct[35]=31 ;idct[36]=51 ;idct[37]=83 ;idct[38]=107 ;idct[39]=89 ;
idct[40]=-118 ;idct[41]=-86 ;idct[42]=-44 ;idct[43]=13 ;idct[44]=69 ;idct[45]=96 ;idct[46]=86 ;idct[47]=68 ;
idct[48]=-115 ;idct[49]=-84 ;idct[50]=-32 ;idct[51]=33 ;idct[52]=81 ;idct[53]=96 ;idct[54]=90 ;idct[55]=89 ;
idct[56]=-96 ;idct[57]=-106 ;idct[58]=-42 ;idct[59]=47 ;idct[60]=75 ;idct[61]=88 ;idct[62]=96 ;idct[63]=80 ;
```

5.7 Software Verification and LA Measurement

■ Dump IDCT data

◆ Compare Data with the Software Version and Hardware Version

■ Check OK

IDCT with Software Version

```
1
2
3 idct_mb_count=0 idct_block_count=1
4 idct[0]=100 ;idct[1]=132 ;idct[2]=126 ;idct[3]=121 ;idct[4]=133 ;idct[5]
5 idct[8]=125 ;idct[9]=130 ;idct[10]=125 ;idct[11]=132 ;idct[12]=128 ;idc
6 idct[16]=131 ;idct[17]=85 ;idct[18]=4 ;idct[19]=53 ;idct[20]=125 ;idct[
7 idct[24]=120 ;idct[25]=86 ;idct[26]=-28 ;idct[27]=32 ;idct[28]=141 ;idc
8 idct[32]=107 ;idct[33]=104 ;idct[34]=10 ;idct[35]=46 ;idct[36]=116 ;idc
9 idct[40]=65 ;idct[41]=92 ;idct[42]=-7 ;idct[43]=39 ;idct[44]=134 ;idct[
0 idct[48]=51 ;idct[49]=103 ;idct[50]=-6 ;idct[51]=34 ;idct[52]=129 ;idct
1 idct[56]=45 ;idct[57]=88 ;idct[58]=9 ;idct[59]=58 ;idct[60]=126 ;idct[6
2
3
4
5 idct_mb_count=0 idct_block_count=2
6 idct[0]=-139 ;idct[1]=-68 ;idct[2]=61 ;idct[3]=134 ;idct[4]=120 ;idct[5]
7 idct[8]=-124 ;idct[9]=-90 ;idct[10]=-29 ;idct[11]=32 ;idct[12]=58 ;idct
8 idct[16]=-97 ;idct[17]=-76 ;idct[18]=-26 ;idct[19]=31 ;idct[20]=49 ;idc
9 idct[24]=-122 ;idct[25]=-105 ;idct[26]=-24 ;idct[27]=40 ;idct[28]=34 ;id
0 idct[32]=-112 ;idct[33]=-102 ;idct[34]=-34 ;idct[35]=31 ;idct[36]=52 ;id
1 idct[40]=-117 ;idct[41]=-87 ;idct[42]=-42 ;idct[43]=14 ;idct[44]=70 ;idc
2 idct[48]=-114 ;idct[49]=-84 ;idct[50]=-31 ;idct[51]=34 ;idct[52]=83 ;idc
3 idct[56]=-95 ;idct[57]=-107 ;idct[58]=-41 ;idct[59]=47 ;idct[60]=76 ;idc
4
5
```

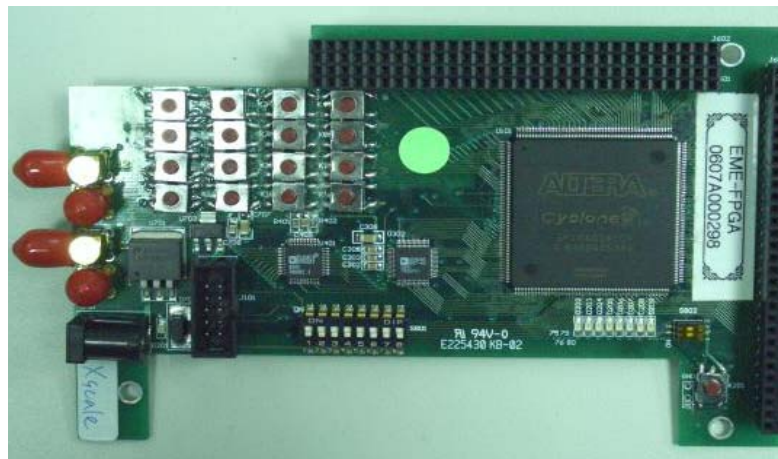
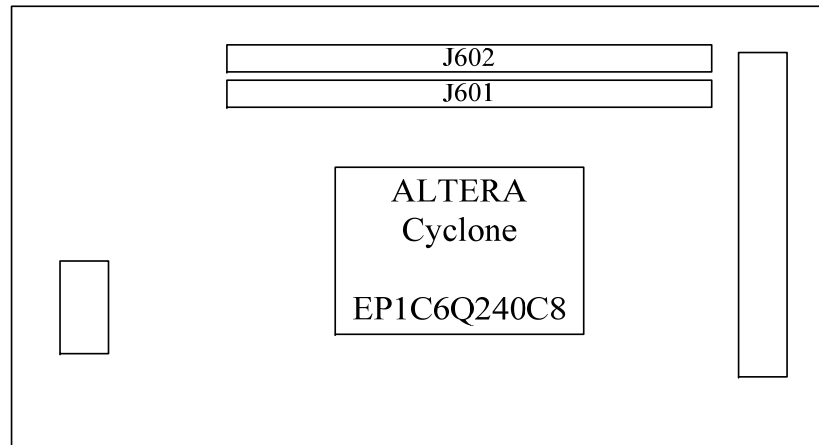
IDCT with Hardware Version

```
11
12
13 idct_mb_count=0 idct_block_count=1
14 idct[0]=97 ;idct[1]=129 ;idct[2]=124 ;idct[3]=118 ;idct[4]=132 ;idct[5]
15 idct[8]=121 ;idct[9]=129 ;idct[10]=123 ;idct[11]=129 ;idct[12]=127 ;idc
16 idct[16]=126 ;idct[17]=84 ;idct[18]=4 ;idct[19]=50 ;idct[20]=124 ;idct[
17 idct[24]=114 ;idct[25]=84 ;idct[26]=-30 ;idct[27]=29 ;idct[28]=139 ;idc
18 idct[32]=105 ;idct[33]=102 ;idct[34]=10 ;idct[35]=44 ;idct[36]=116 ;idc
19 idct[40]=64 ;idct[41]=91 ;idct[42]=-8 ;idct[43]=37 ;idct[44]=133 ;idct[
20 idct[48]=49 ;idct[49]=101 ;idct[50]=-7 ;idct[51]=32 ;idct[52]=128 ;idct
21 idct[56]=43 ;idct[57]=88 ;idct[58]=7 ;idct[59]=56 ;idct[60]=124 ;idct[6
22
23
24
25 idct_mb_count=0 idct_block_count=2
26 idct[0]=-142 ;idct[1]=-68 ;idct[2]=58 ;idct[3]=132 ;idct[4]=118 ;idct[5]
27 idct[8]=-126 ;idct[9]=-90 ;idct[10]=-30 ;idct[11]=31 ;idct[12]=55 ;idct
28 idct[16]=-99 ;idct[17]=-76 ;idct[18]=-26 ;idct[19]=30 ;idct[20]=48 ;idc
29 idct[24]=-123 ;idct[25]=-105 ;idct[26]=-26 ;idct[27]=39 ;idct[28]=33 ;id
30 idct[32]=-113 ;idct[33]=-101 ;idct[34]=-35 ;idct[35]=31 ;idct[36]=51 ;id
31 idct[40]=-118 ;idct[41]=-86 ;idct[42]=-44 ;idct[43]=13 ;idct[44]=69 ;idc
32 idct[48]=-115 ;idct[49]=-84 ;idct[50]=-32 ;idct[51]=33 ;idct[52]=81 ;idc
33 idct[56]=-96 ;idct[57]=-106 ;idct[58]=-42 ;idct[59]=47 ;idct[60]=75 ;idc
34
35
36
```

5.7 Software Verification and LA Measurement

■ LA Measurement

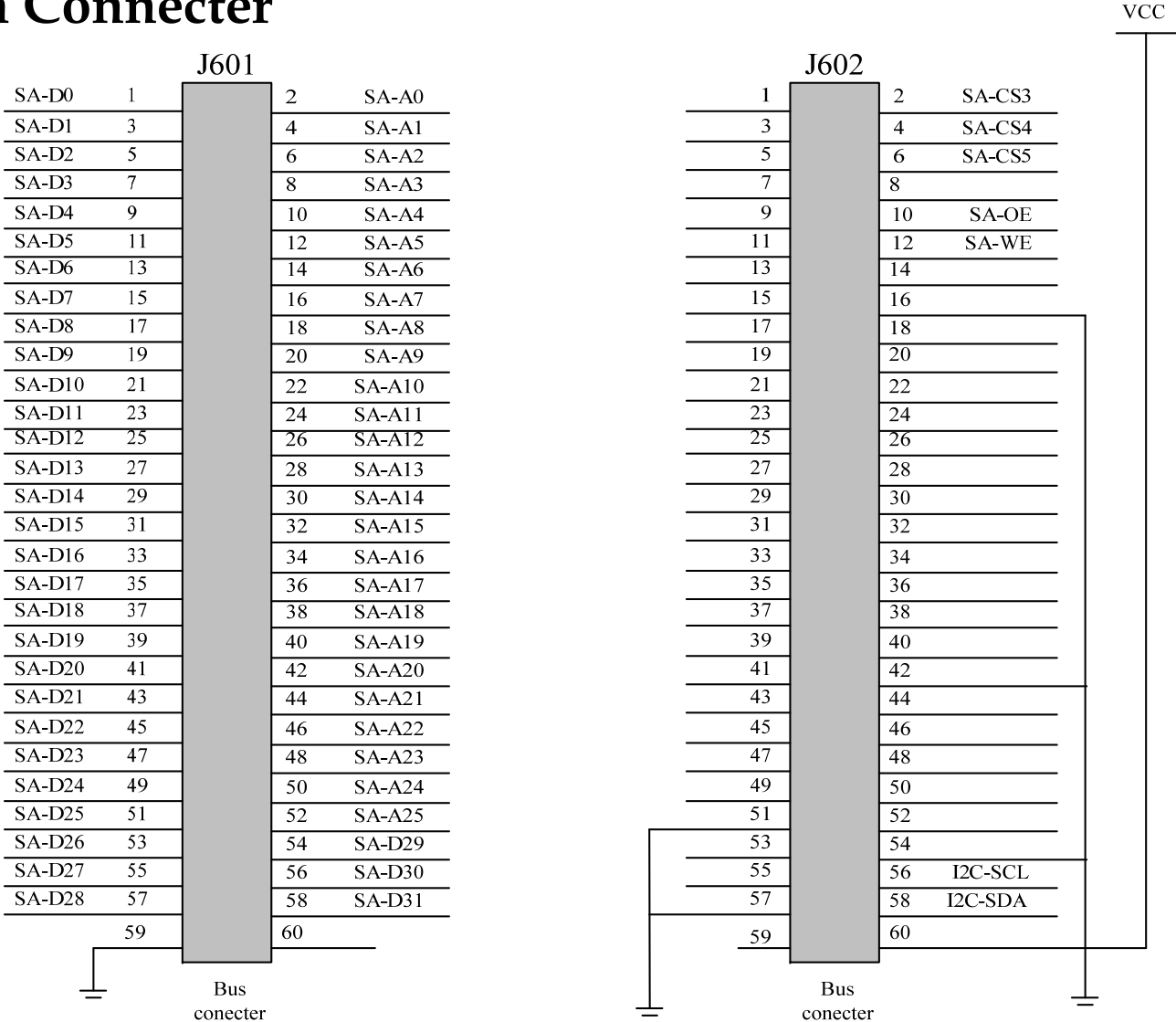
◆ FPGA Daughter Board



5.7 Software Verification and LA Measurement

■ LA Measurement

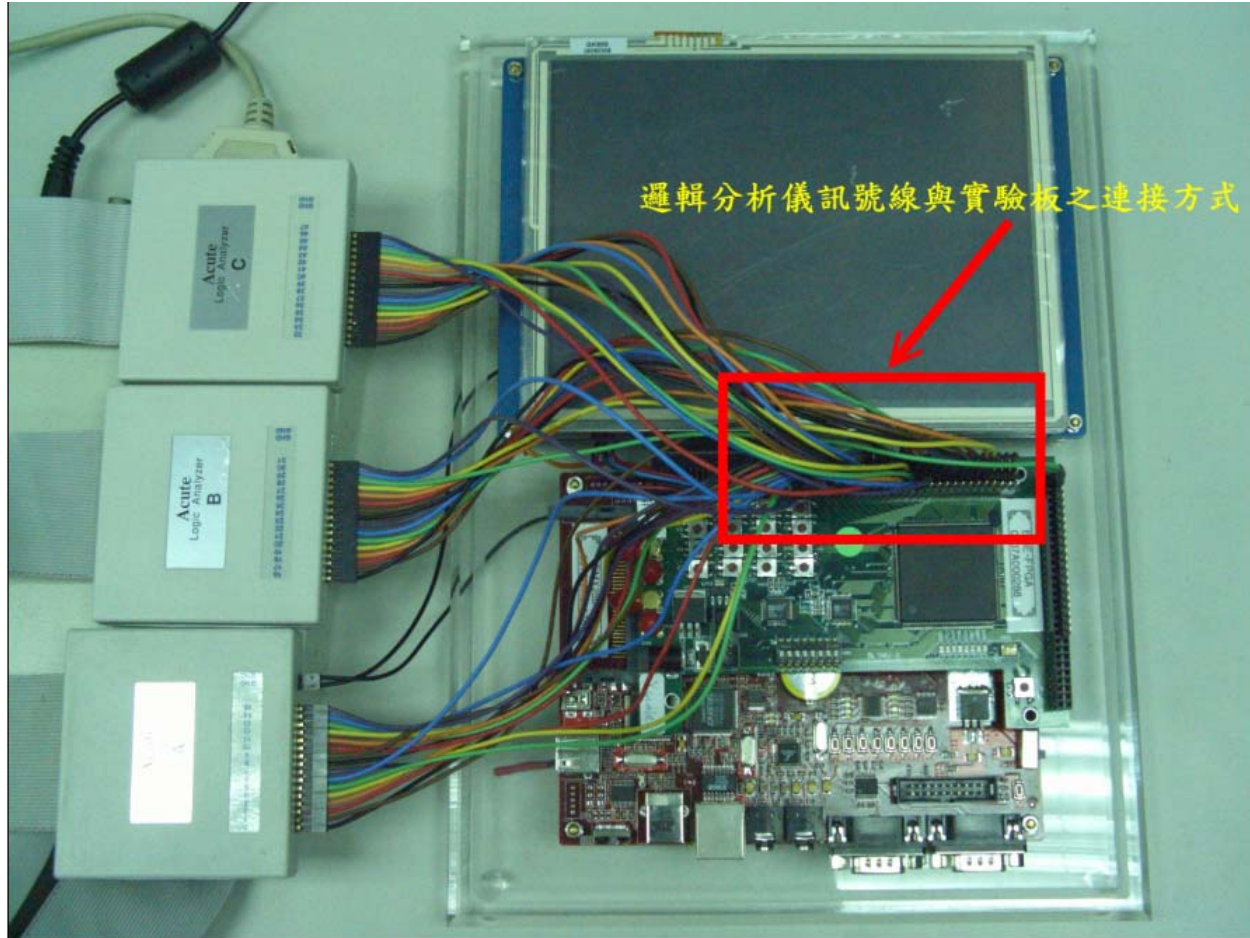
◆ Bus Pin Connector



5.7 Software Verification and LA Measurement

■ LA Measurement

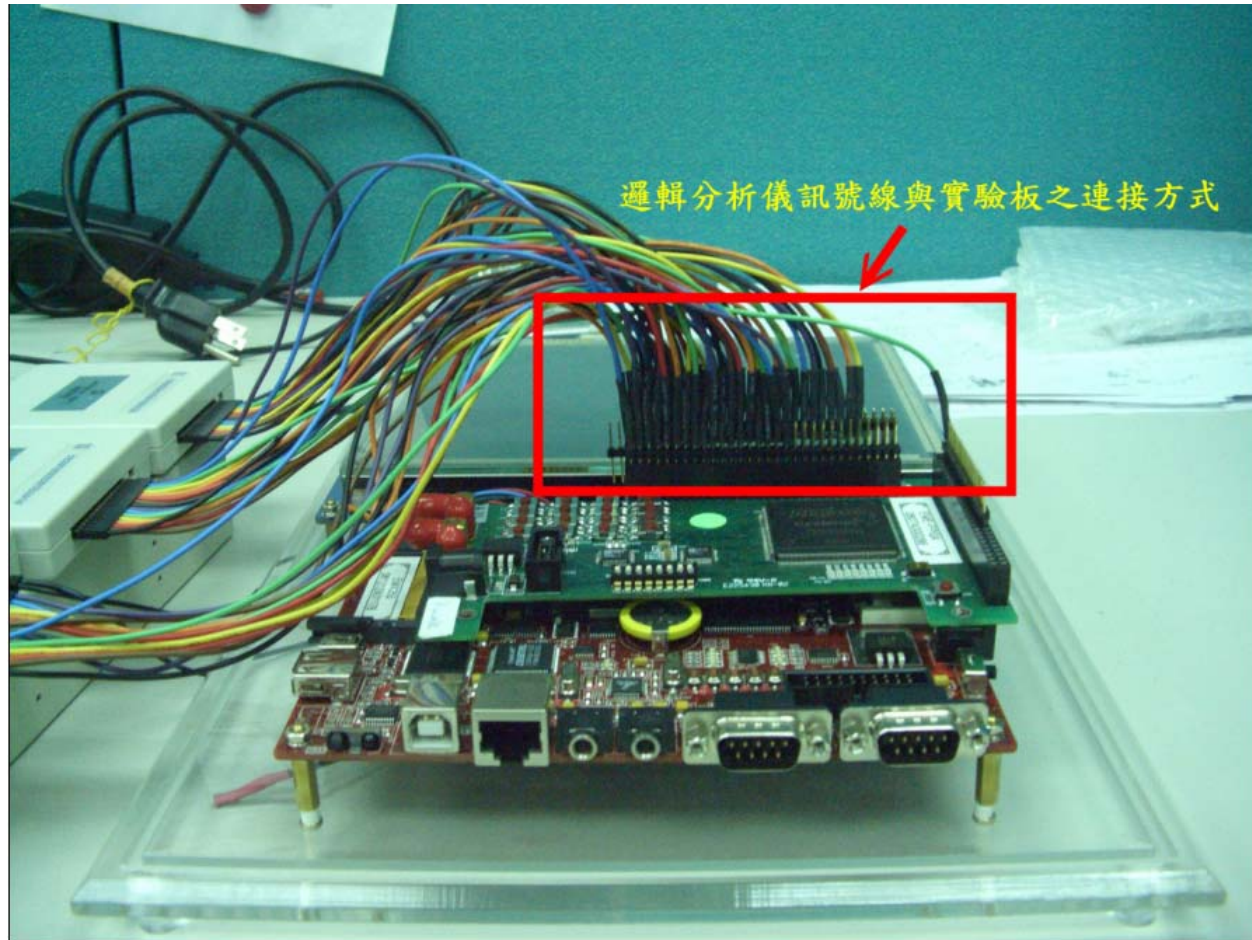
- ◆ Connect the Signals on PXA255 Platform to the Logic Analyser



5.7 Software Verification and LA Measurement

■ LA Measurement

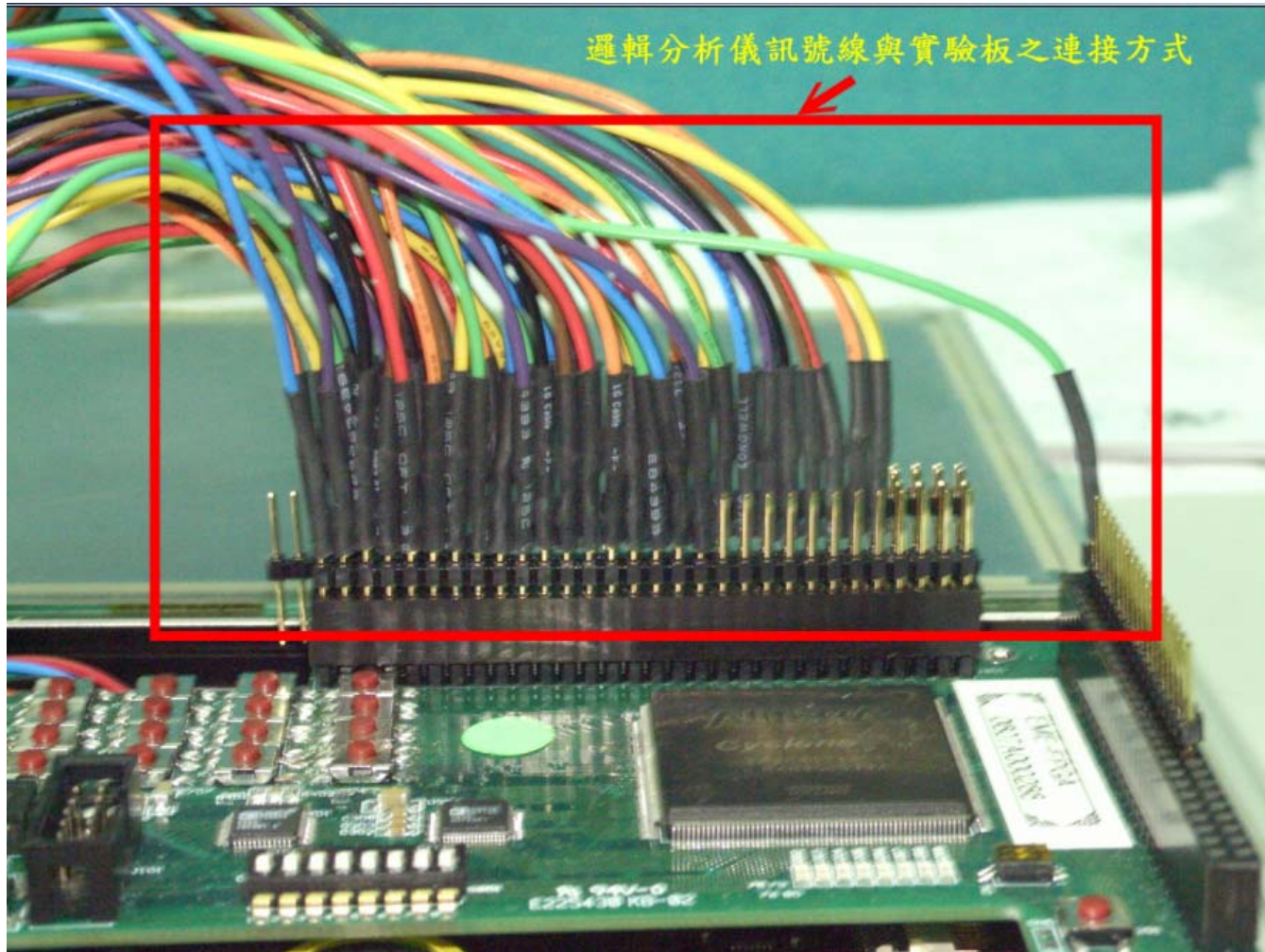
- ◆ Connect the Signals on PXA255 Platform to the Logic Analyser



5.7 Software Verification and LA Measurement

■ LA Measurement

- ◆ Connect the Signals on PXA255 Platform to the Logic Analyser



5.7 Software Verification and LA Measurement

■ LA Measurement

◆ Use Logic Analyser

■ Open the PC-based Logic Analyser

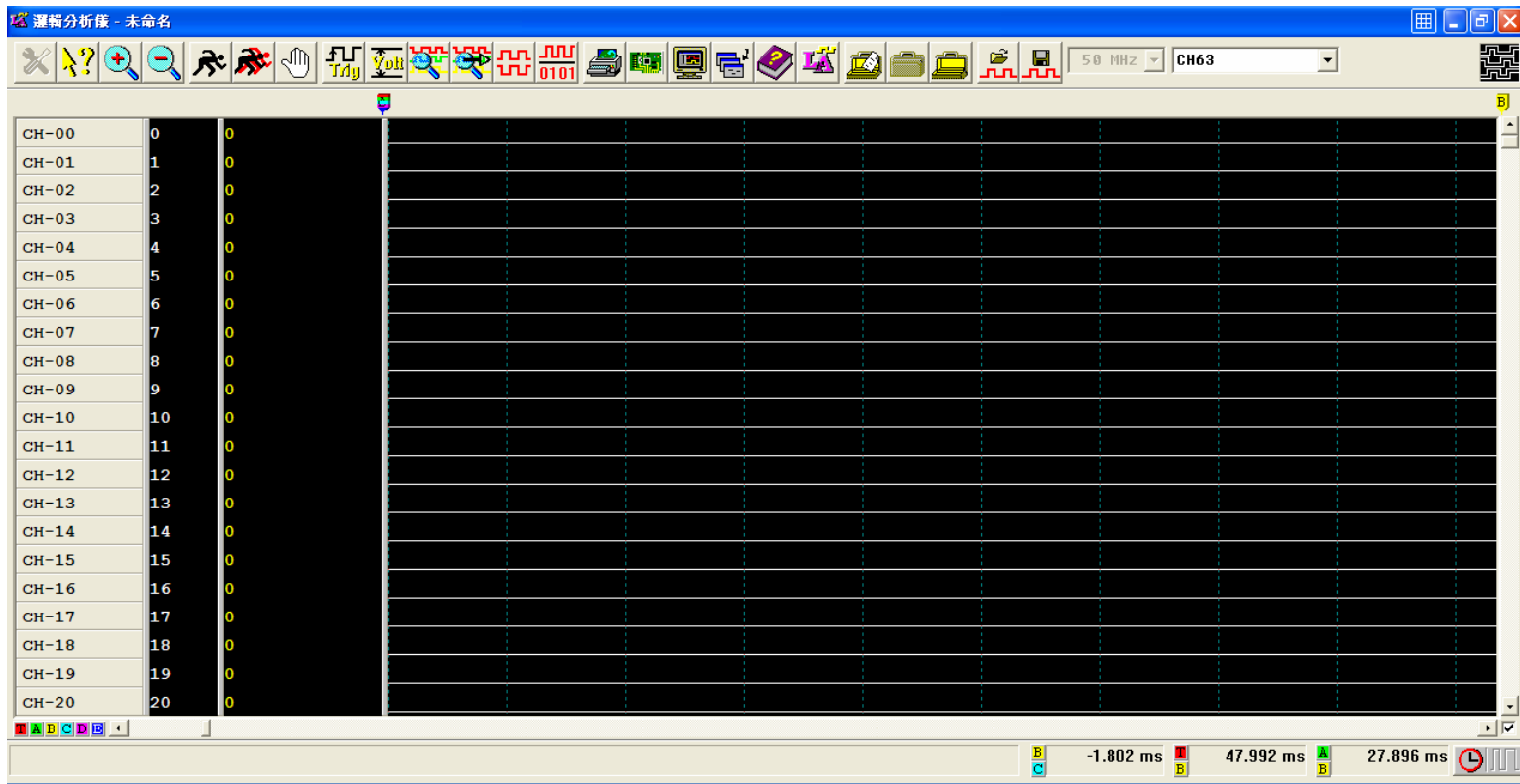


5.7 Software Verification and LA Measurement

■ LA Measurement

◆ The PC-based Logic Analyser

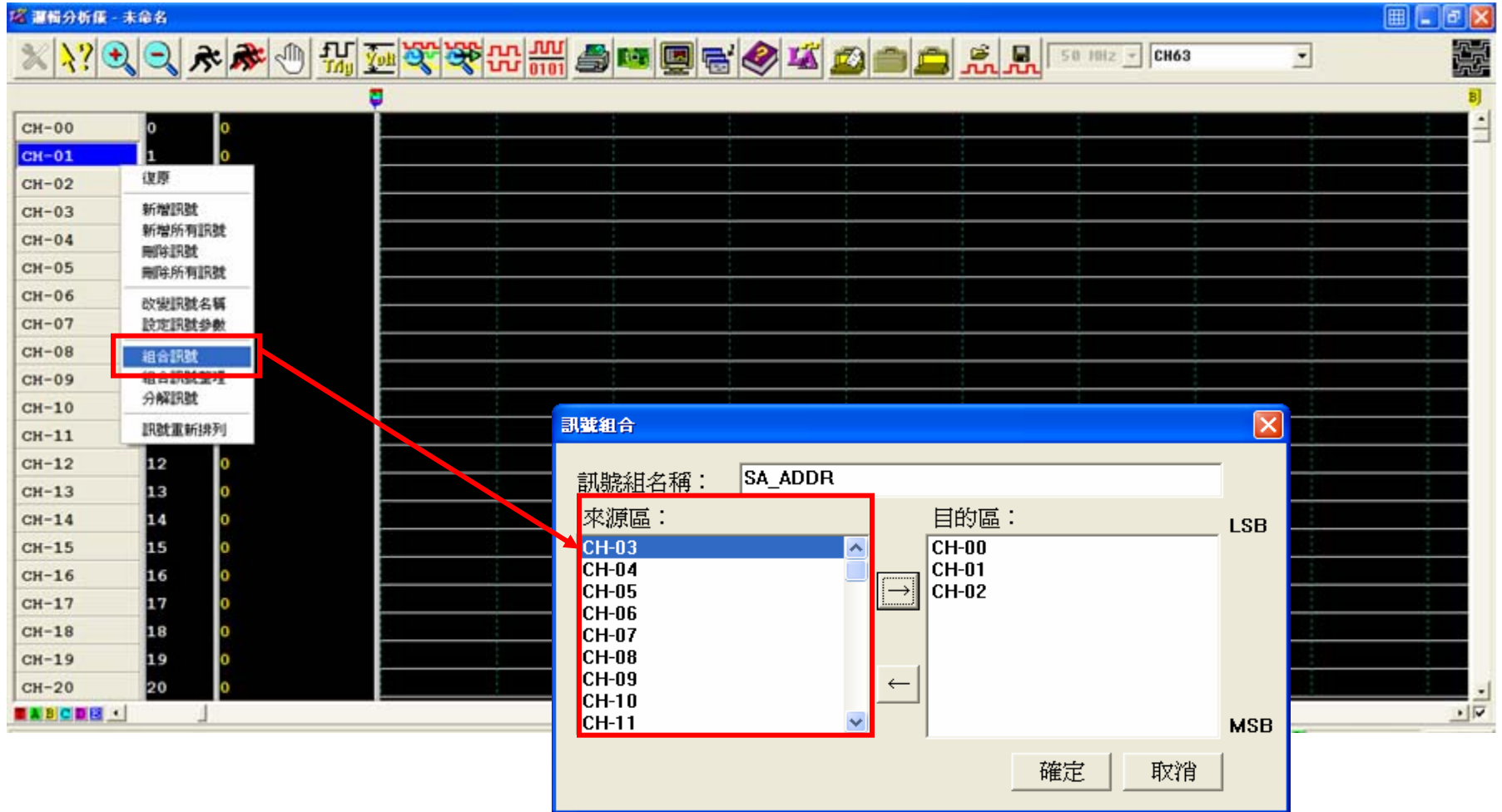
■ GUI Interface



5.7 Software Verification and LA Measurement

■ LA Measurement

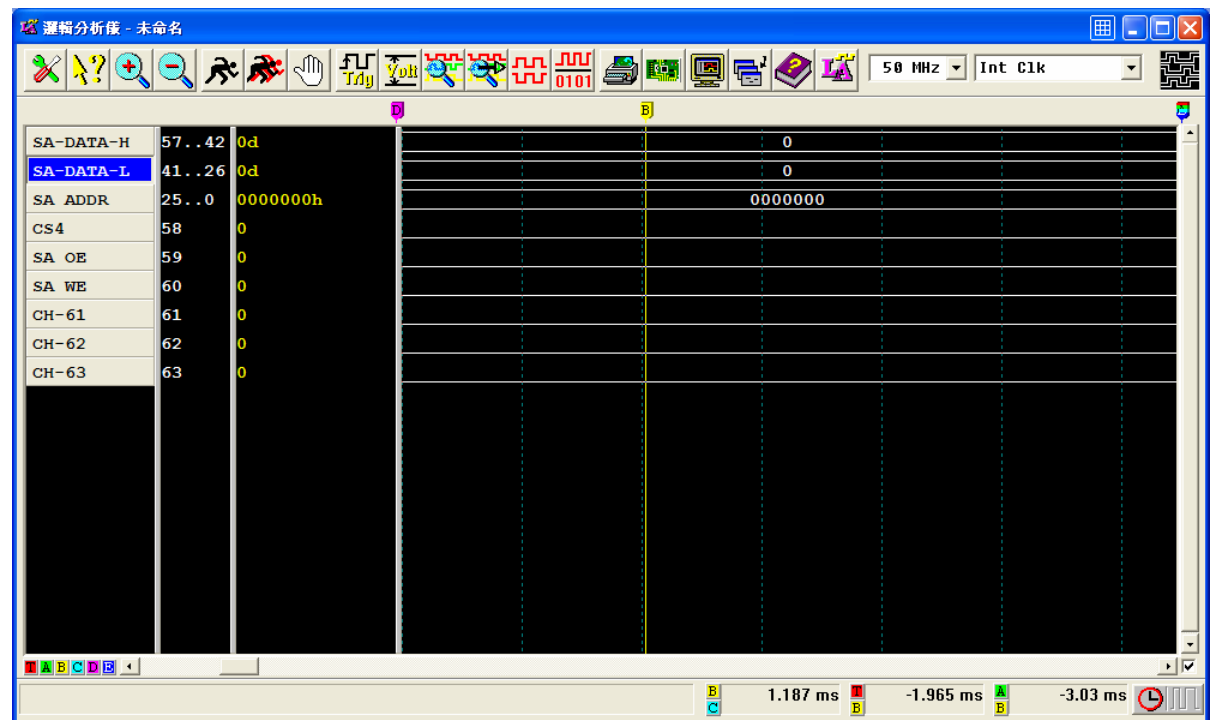
◆ Assign the Name to Each Signal



5.7 Software Verification and LA Measurement

■ LA Measurement

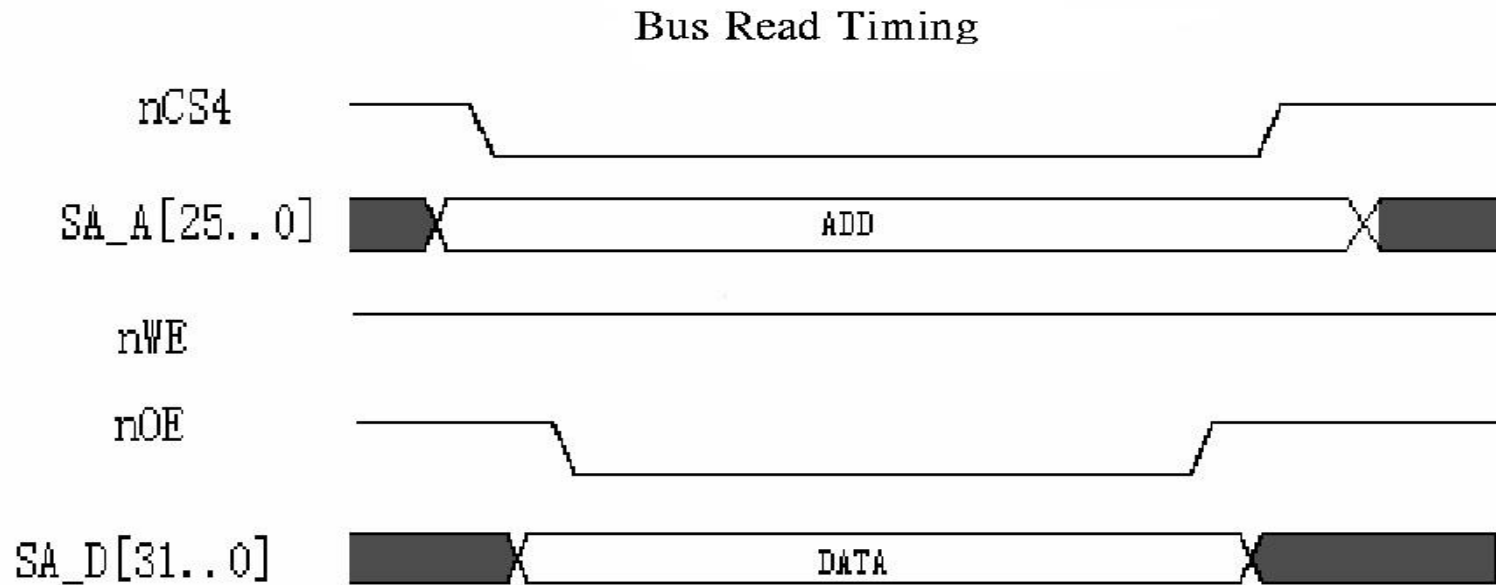
- ◆ Set Group for the Signals
 - SA-ADDR :Pin0~Pin25
 - SA-DATA-L:Pin26~Pin41
 - SA-DATA-H:Pin42~Pin57
 - CS4:Pin58
 - SA-WE:Pin59
 - SA-OE:Pin60



5.7 Software Verification and LA Measurement

■ LA Measurement

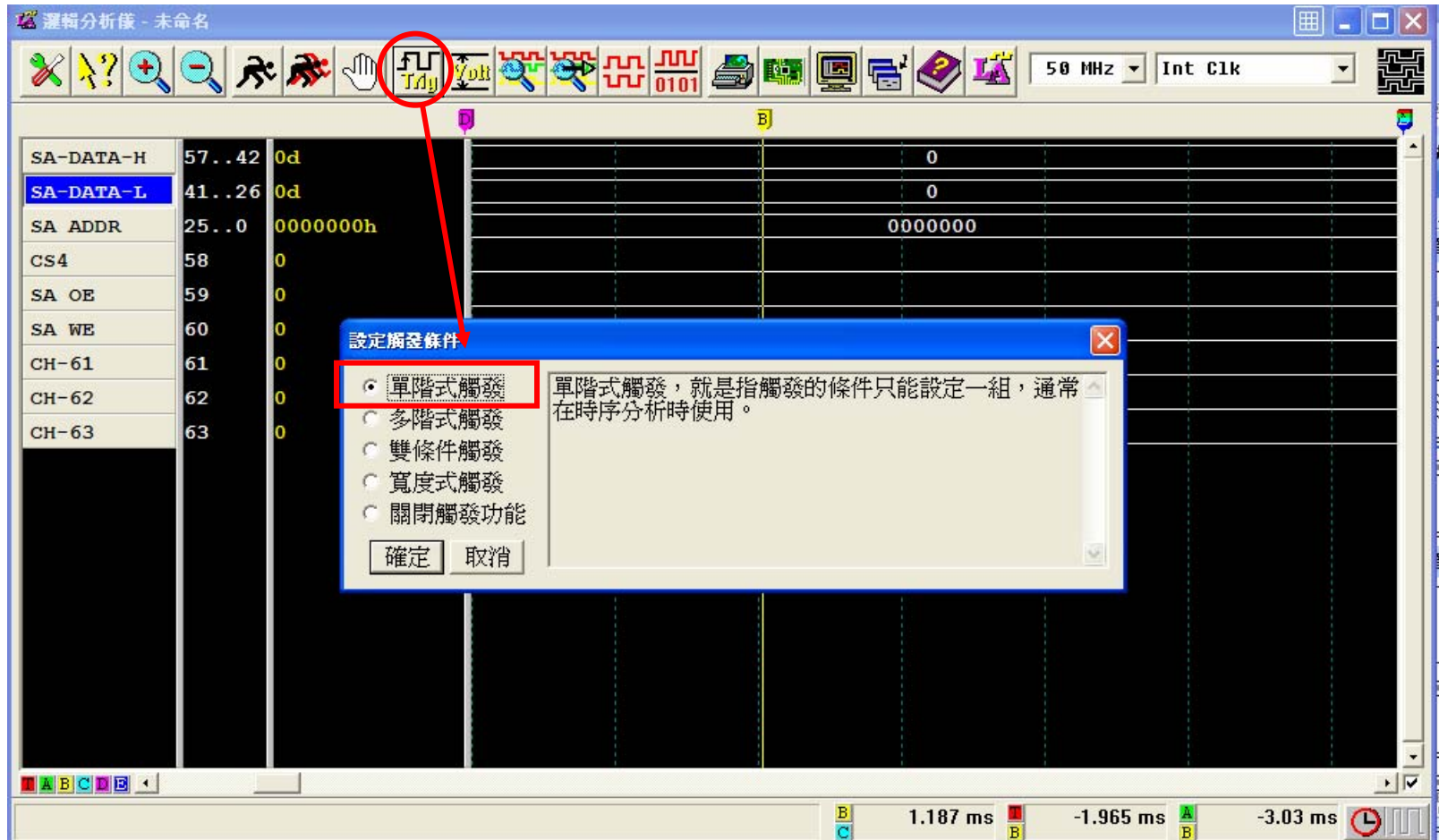
◆ The Read Timing



5.7 Software Verification and LA Measurement

■ LA Measurement

◆ Set the Condition to Trigger



5.7 Software Verification and LA Measurement

■ LA Measurement

◆ Set the Condition to Trigger

- CS4 Negative Edge

- SA ADDR = 0x10000

Trigger 參數設定 ~ 方式二

☐ 啟動 Pre-Trigger

訊號名稱: SA_OE 條件: 0

觸發條件: 數值: 0

Label	value	type
CS4 SA ADDR	↓ 0010000	HEX

觸發寬度設定

☐ > ☐ < 寬度: 0 ns 觸發寬度為 0ns

延遲觸發: 0 ns

Pass Count: 1 確定 取消 清除 <<

5.7 Software Verification and LA Measurement

■ LA Measurement

◆ Run. Click the icon

SA-DATA-H 57..42 0d
SA-DATA-L 41..26 0d
SA ADDR 25..0 0000000h
CS4 58 0
SA OE 59 0
SA WE 60 0
CH-61 61 0
CH-62 62 0
CH-63 63 0

SA-DATA-H 57..42 0d
SA-DATA-L 41..26 0d
SA ADDR 25..0 0000000h
CS4 58 0
SA OE 59 0
SA WE 60 0
CH-61 61 0
CH-62 62 0
CH-63 63 0

Wait the Trigger Condition

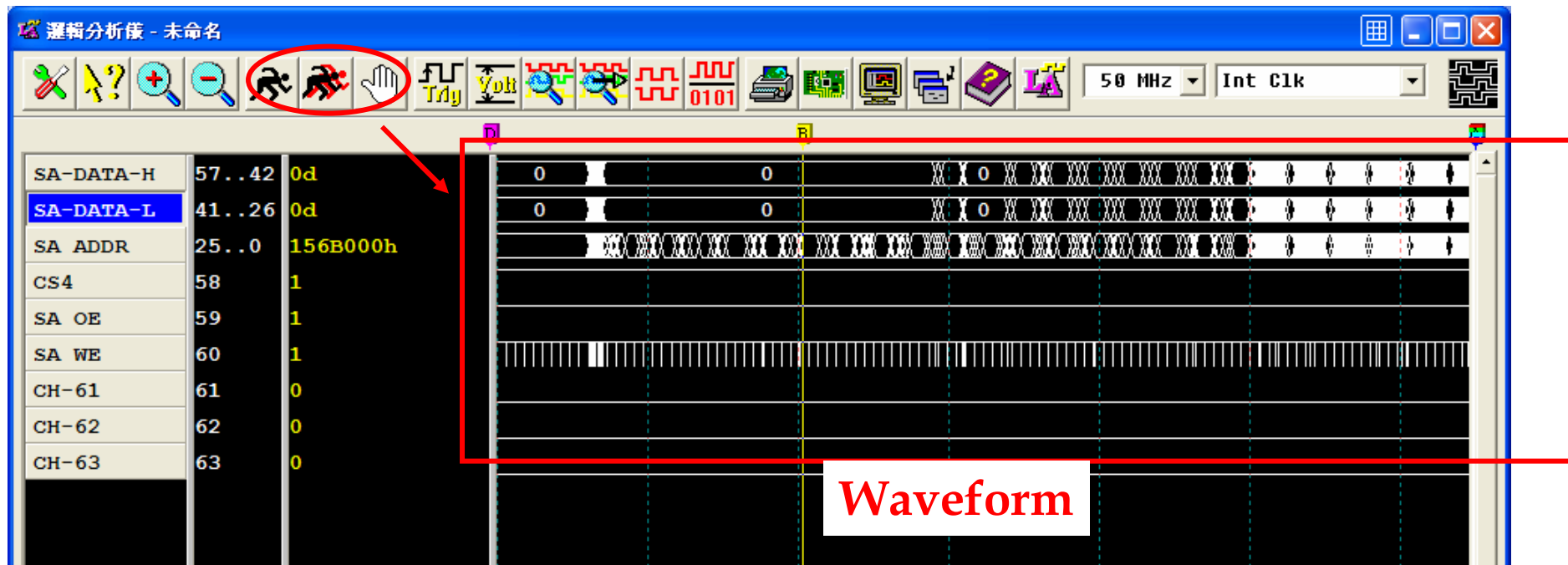
Capture..... 1.187 ms -1.965 ms -3.03 ms

5.7 Software Verification and LA Measurement

■ LA Measurement

◆ The Trigger Condition

- Execute the Software Application. Get the Waveform.



5.7 Software Verification and LA Measurement

■ LA Measurement

◆ Set the Condition to Find the Waveform

Find the Signals You Want to See

搜尋特定波形

訊號名稱: SA-DATA-L 條件: Dec 刪除 新增

搜尋條件: 數值: 97

Signal	Value	Type
CS4	0	
SA_ADDR	0010000	HEX
SA_OE	0	
SA-DATA-L	97	DEC

搜尋起始點: 緩衝區開頭

搜尋結束點: 緩衝區尾端

Pass Count: 0 確定 取消 清除 內定條件

5.7 Software Verification and LA Measurement

■ LA Measurement

◆ Find the Data Compare with the Dump Data .Correct!!

