# Lab 4

## Laboratory for Software Optimizations

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

# Outline

4.1 Preparing

4.2 MPEG-2 Decoder Software

4.3 IDCT Source Code

4.4 IDCT Fast Algorithm - 1

4.5 IDCT Fast Algorithm – 2

4.6 Insert Assembly Code in C Language

# 4.1 Preparing

4.1 Preparing

4.2 MPEG-2 Decoder Software

4.3 IDCT Source Code

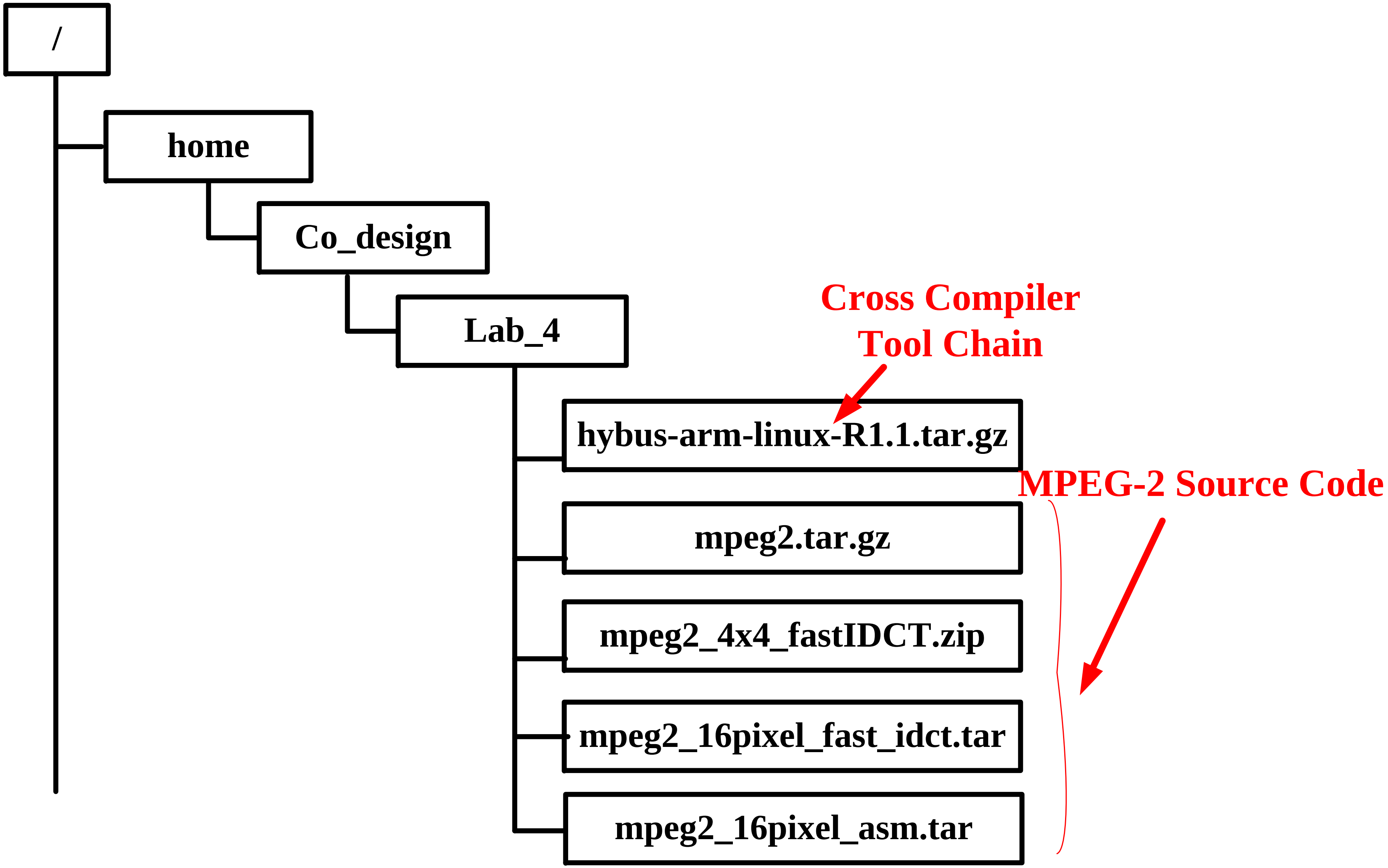
4.4 IDCT Fast Algorithm - 1

4.5 IDCT Fast Algorithm – 2

4.6 Insert Assembly Code in C Language

# 4.1 Preparing

## ■ File Directories



/

home

Co_design

Lab_4

hybus-arm-linux-R1.1.tar.gz

mpeg2.tar.gz

mpeg2_4x4_fastIDCT.zip

mpeg2_16pixel_fast_idct.tar

mpeg2_16pixel_asm.tar

Cross Compiler Tool Chain

MPEG-2 Source Code

# 4.1 Preparing

## ■ Source Code Files

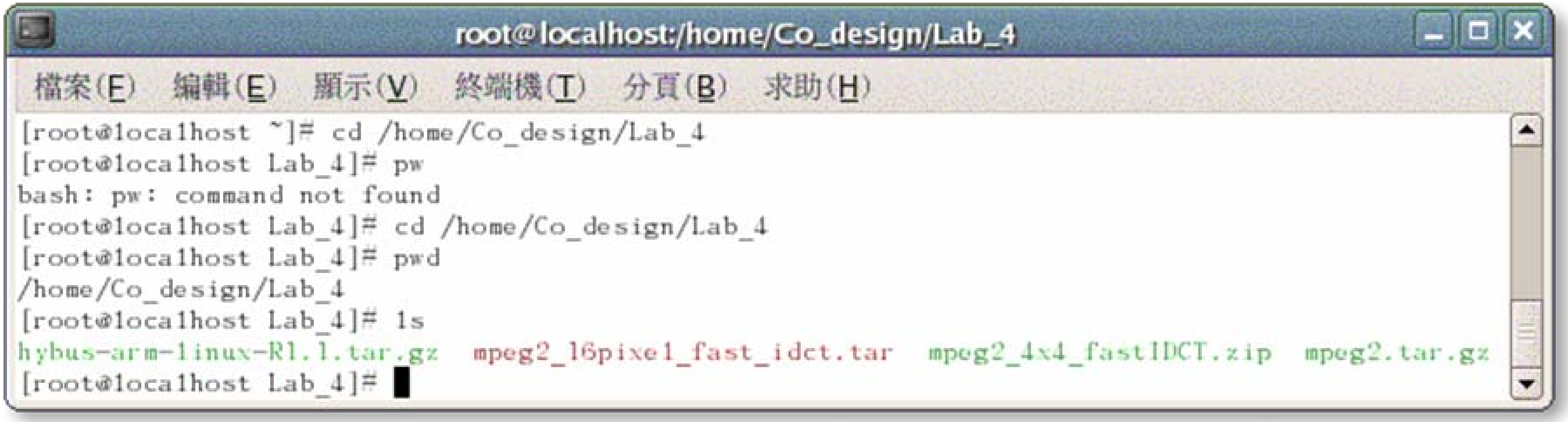
- hybus-arm-linux-R1.1.tar.gz
  - Cross Compiler Tool Chain
- mpeg2.tar.gz (4.2)
  - The Original MPEG-2 Decoder Source Code
- mpeg2_4x4_fastIDCT.zip (4.4)
  - The Source Code after Modify about 4x4 IDCT
  - IDCT Fast Algorithm - 1: 4x4 IDCT
- mpeg2_16pixel_fast_idct.tar (4.5)
  - The Source Code after Modify about 16 Pixels IDCT
  - IDCT Fast Algorithm - 2: 16 pixels IDCT
- mpeg2_16pixel_asm.tar (4.6)
  - The Source Code after Assembly Coding

# 4.1 Preparing

## Source Code Files

- Prepare the files
  - />cd /home/Co_design/Lab_4/
  - Lab_4>ls

  See the source code files about LAB4



root@localhost:/home/Co_design/Lab_4
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
[root@localhost ~]# cd /home/Co_design/Lab_4
[root@localhost Lab_4]# pw
bash: pw: command not found
[root@localhost Lab_4]# cd /home/Co_design/Lab_4
[root@localhost Lab_4]# pwd
/home/Co_design/Lab_4
[root@localhost Lab_4]# ls
hybus-arm-linux-R1.1.tar.gz  mpeg2_16pixel_fast_idct.tar  mpeg2_4x4_fastIDCT.zip  mpeg2.tar.gz
[root@localhost Lab_4]# █
```

# 4.2 MPEG-2 Decoder Software

4.1 Preparing

4.2 MPEG-2 Decoder Software

4.3 IDCT Source Code

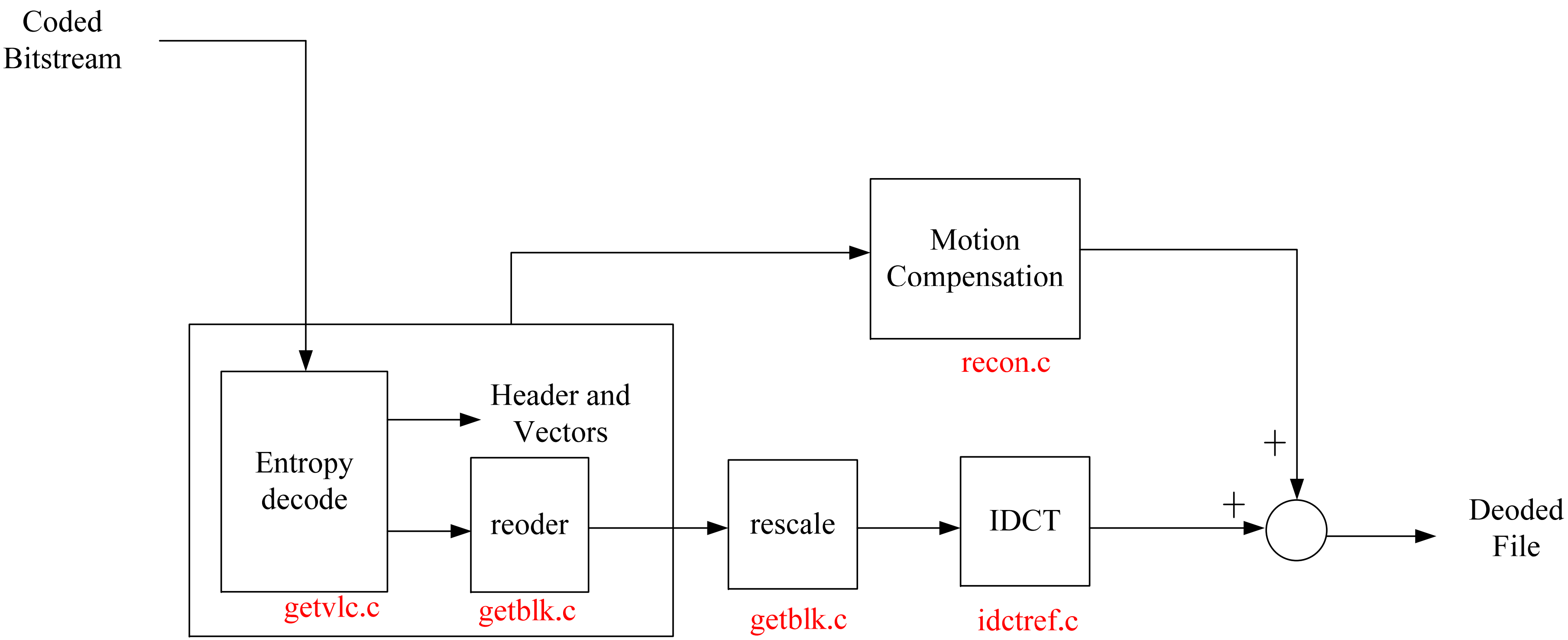
4.4 IDCT Fast Algorithm - 1

4.5 IDCT Fast Algorithm – 2

4.6 Insert Assembly Code in C Language

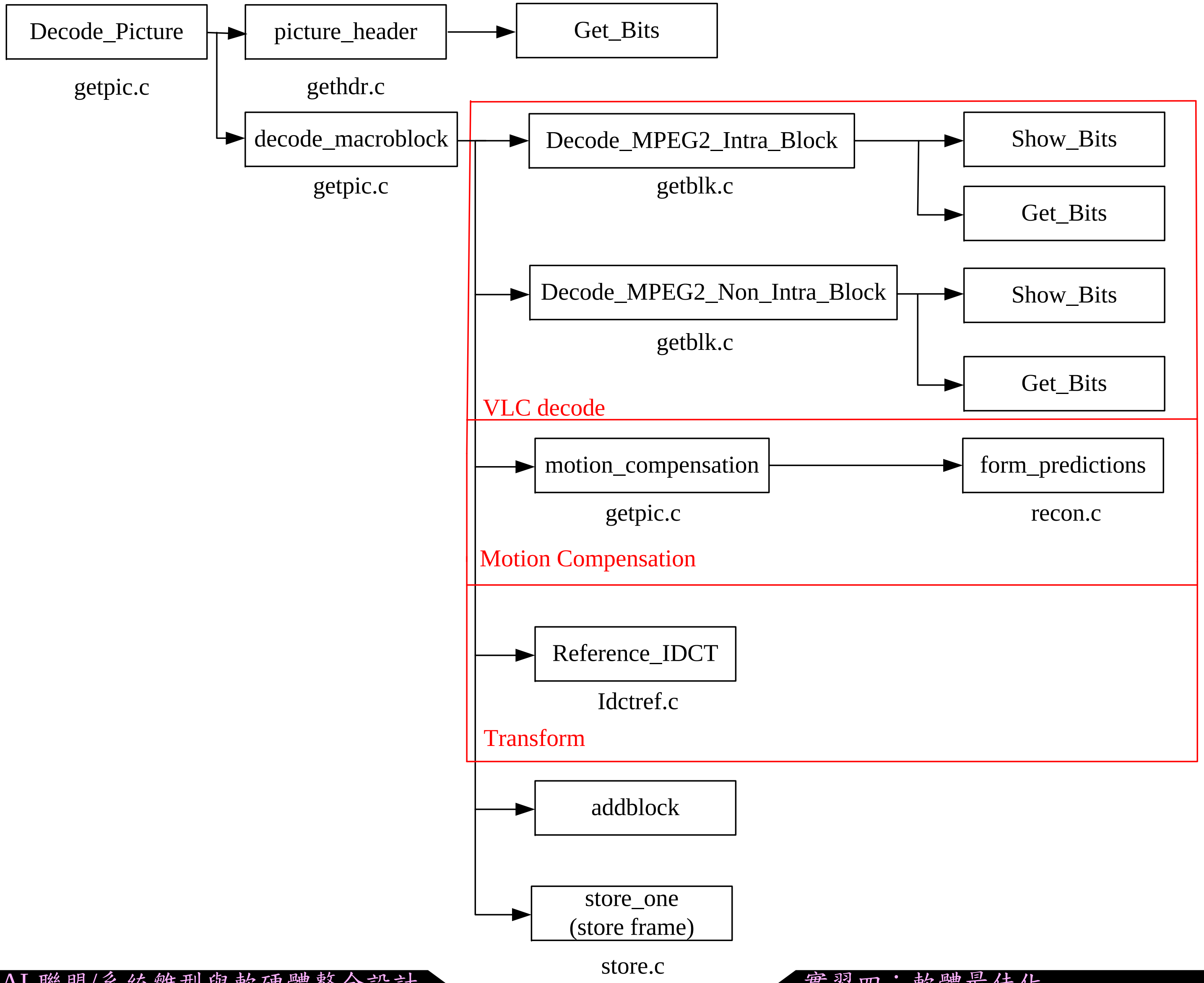
# 4.2 MPEG-2 Decoder Software

## ■ MPEG-2 Decoder Flow



# 4.2 MPEG-2 Decoder Software

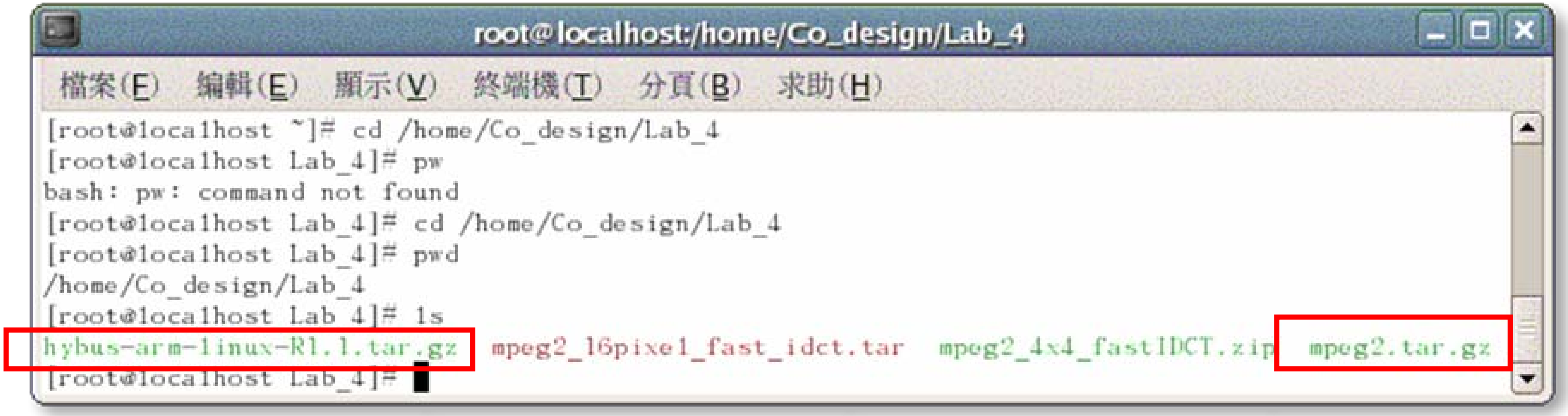
## ■ MPEG-2 Decoder Flow



# 4.2 MPEG-2 Decoder Software

## Setup Environment

- Get the MPEG-2 Decoder Source Code Package
  - Path:/home/Co_design/Lab_4/


root@localhost:/home/Co_design/Lab_4
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

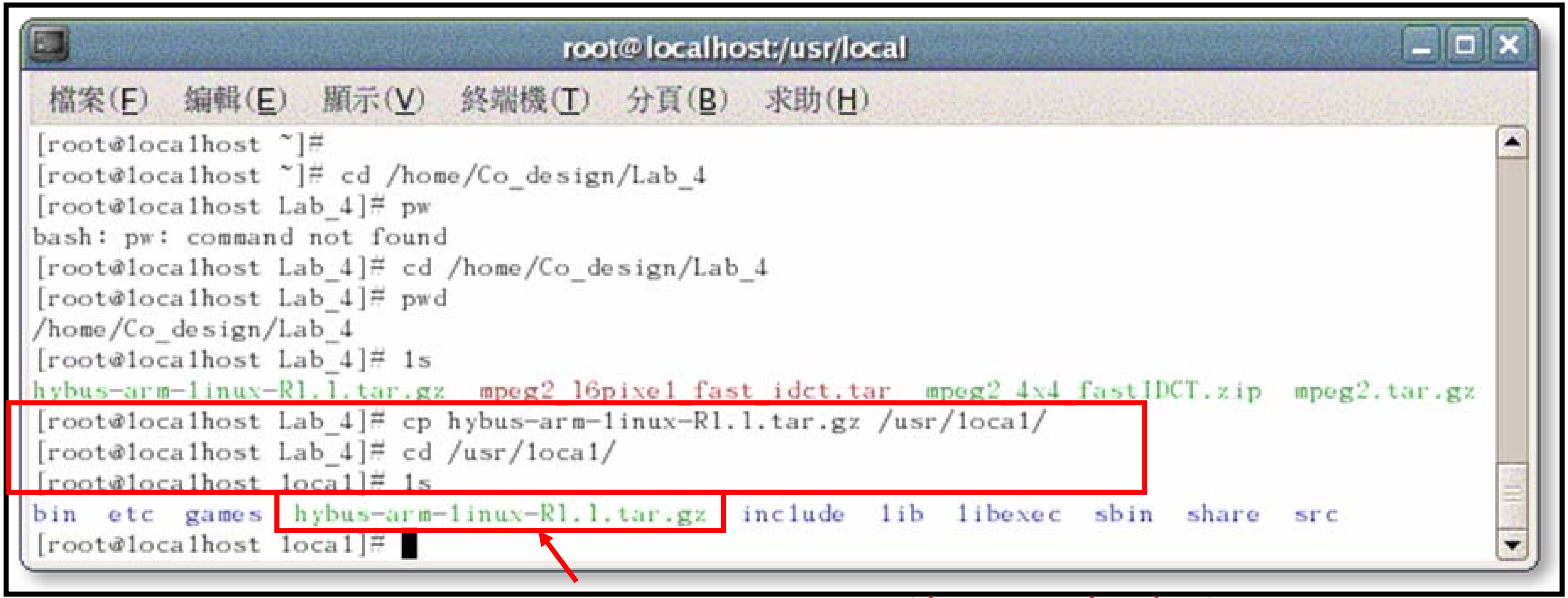
```
[root@localhost ~]# cd /home/Co_design/Lab_4
[root@localhost Lab_4]# pw
bash: pw: command not found
[root@localhost Lab_4]# cd /home/Co_design/Lab_4
[root@localhost Lab_4]# pwd
/home/Co_design/Lab_4
[root@localhost Lab_4]# ls
hybus-arm-linux-R1.1.tar.gz  mpeg2_16pixel_fast_idct.tar  mpeg2_4x4_fastIDCT.zip  mpeg2.tar.gz
[root@localhost Lab_4]#
```

Need “mpeg2.tar.gz” and “hybus-arm-linux-R1.1.tar.gz”

# 4.2 MPEG-2 Decoder Software

## Decompress Source Code Package

- Decompress the Cross Compiler Tool Chain
  - Lab_4>cp hybus-arm-linux-R1.1.tar.gz /usr/local
  - Lab_4>cp /usr/local/
  - local>ls



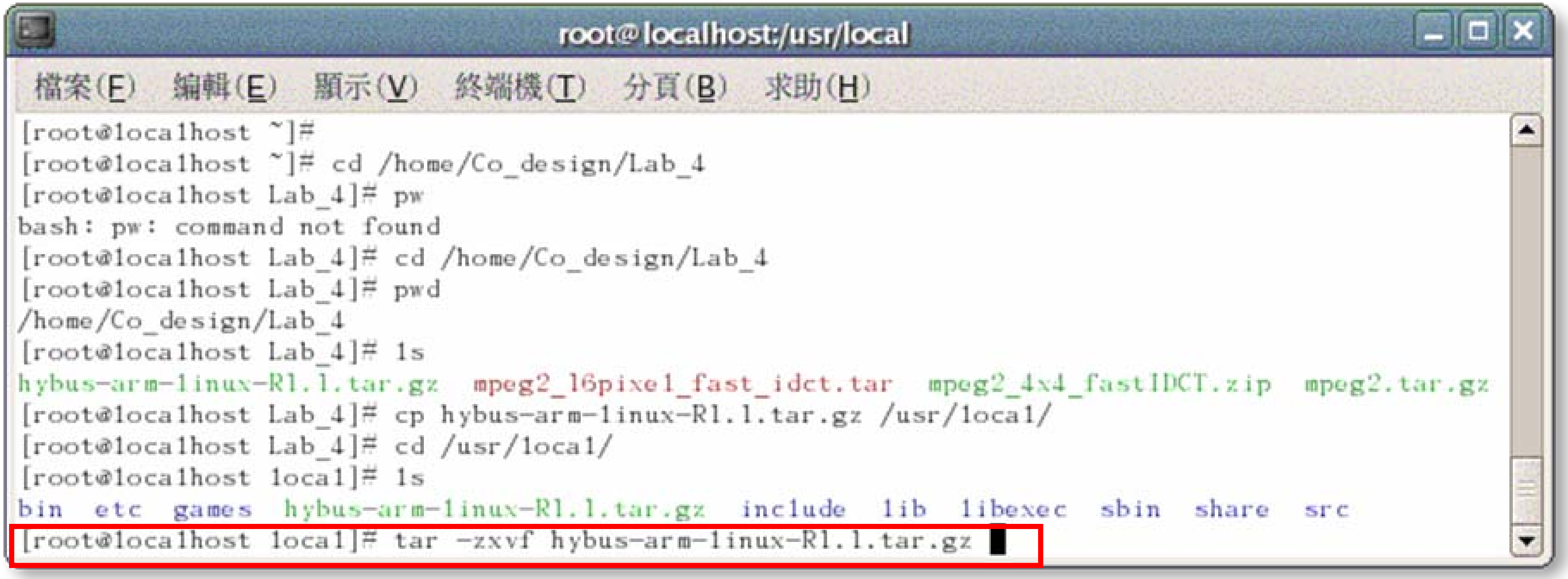
```
[root@localhost ~]#
[root@localhost ~]# cd /home/Co_design/Lab_4
[root@localhost Lab_4]# pw
bash: pw: command not found
[root@localhost Lab_4]# cd /home/Co_design/Lab_4
[root@localhost Lab_4]# pwd
/home/Co_design/Lab_4
[root@localhost Lab_4]# ls
hybus-arm-linux-R1.1.tar.gz  mpeg2_16pixel_fast_idct.tar  mpeg2_4x4_fastIDCT.zip  mpeg2.tar.gz
[root@localhost Lab_4]# cp hybus-arm-linux-R1.1.tar.gz /usr/local/
[root@localhost Lab_4]# cd /usr/local/
[root@localhost local]# ls
bin  etc  games  hybus-arm-linux-R1.1.tar.gz  include  lib  libexec  sbin  share  src
[root@localhost local]# 
```

Cross Compiler Tool Chain

# 4.2 MPEG-2 Decoder Software

## ■ Decompress Source Code Package

- Decompress the Cross Compiler Tool Chain
  - local>tar –zxvf hybus-arm-linux-R1.1.tar.gz

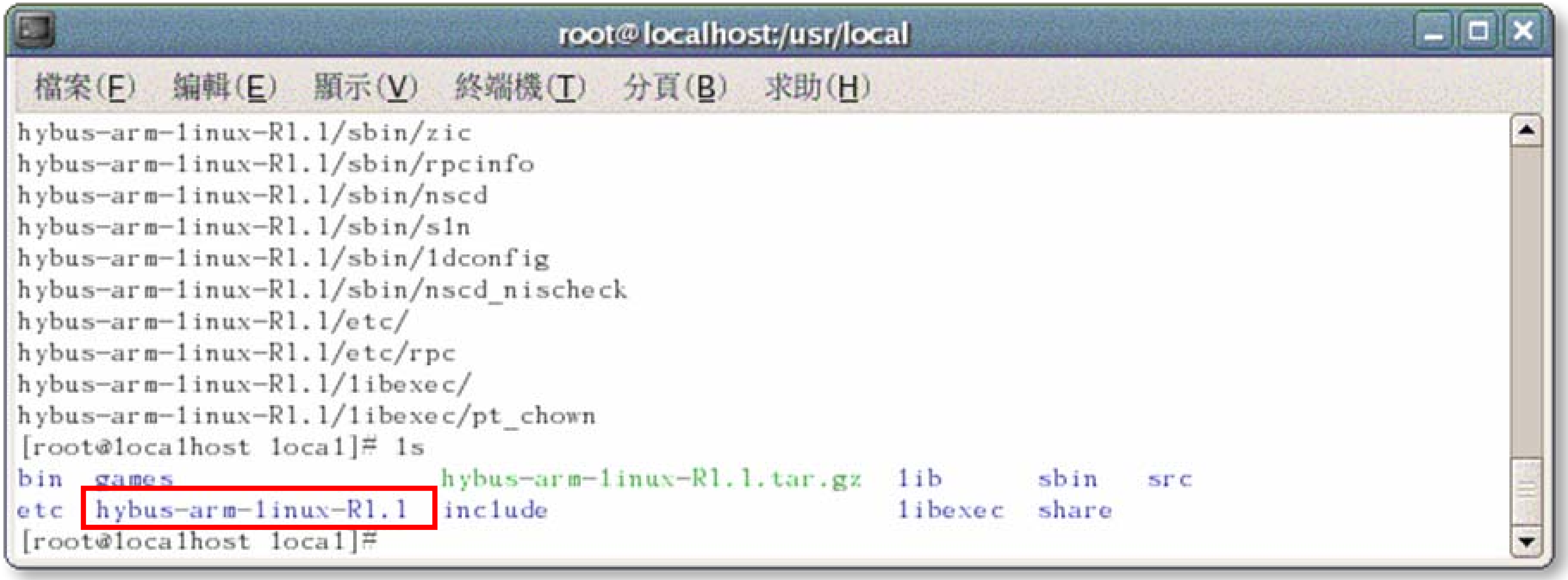


```
[root@localhost ~]#
[root@localhost ~]# cd /home/Co_design/Lab_4
[root@localhost Lab_4]# pw
bash: pw: command not found
[root@localhost Lab_4]# cd /home/Co_design/Lab_4
[root@localhost Lab_4]# pwd
/home/Co_design/Lab_4
[root@localhost Lab_4]# ls
hybus-arm-linux-R1.1.tar.gz  mpeg2_16pixel_fast_idct.tar  mpeg2_4x4_fastIDCT.zip  mpeg2.tar.gz
[root@localhost Lab_4]# cp hybus-arm-linux-R1.1.tar.gz /usr/local/
[root@localhost Lab_4]# cd /usr/local/
[root@localhost local]# ls
bin  etc  games  hybus-arm-linux-R1.1.tar.gz  include  lib  libexec  sbin  share  src
[root@localhost local]# tar -zxvf hybus-arm-linux-R1.1.tar.gz
```

# 4.2 MPEG-2 Decoder Software

## ■ Decompress Source Code Package

- Decompress the Cross Compiler Tool Chain
  - Get “hybus-arm-linux-R1.1” Folder

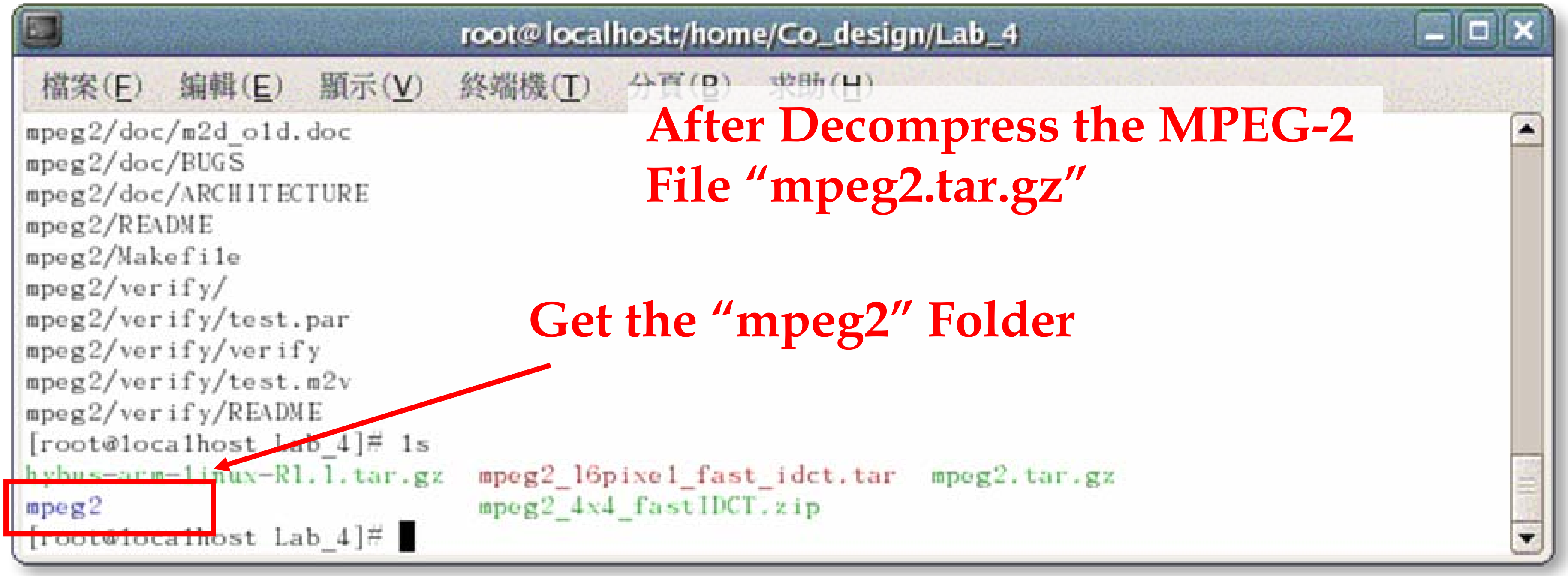


```
hybus-arm-linux-R1.1/sbin/zic
hybus-arm-linux-R1.1/sbin/rpcinfo
hybus-arm-linux-R1.1/sbin/nscd
hybus-arm-linux-R1.1/sbin/sln
hybus-arm-linux-R1.1/sbin/ldconfig
hybus-arm-linux-R1.1/sbin/nscd_nischeck
hybus-arm-linux-R1.1/etc/
hybus-arm-linux-R1.1/etc/rpc
hybus-arm-linux-R1.1/libexec/
hybus-arm-linux-R1.1/libexec/pt_chown
[root@localhost local]# ls
bin  games                 hybus-arm-linux-R1.1.tar.gz  lib      sbin   src
etc  hybus-arm-linux-R1.1  include                      libexec  share
[root@localhost local]#
```

# 4.2 MPEG-2 Decoder Software

## ■ Decompress Source Code Package

- Decompress the MPEG-2 Decoder File “mpeg2.tar.gz”
  - #>cd /home/Co_design/Lab_4
  - #Lab_4>tar –zxvf mpeg2.tar.gz


root@localhost:/home/Co_design/Lab_4
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
mpeg2/doc/m2d_old.doc
mpeg2/doc/BUGS
mpeg2/doc/ARCHITECTURE
mpeg2/README
mpeg2/Makefile
mpeg2/verify/
mpeg2/verify/test.par
mpeg2/verify/verify
mpeg2/verify/test.m2v
mpeg2/verify/README
[root@localhost Lab_4]# ls
hybus-arm-linux-R1.1.tar.gz  mpeg2_16pixel_fast_idct.tar  mpeg2.tar.gz
mpeg2                        mpeg2_4x4_fastIDCT.zip
[root@localhost Lab_4]# 
```

After Decompress the MPEG-2 File “mpeg2.tar.gz”

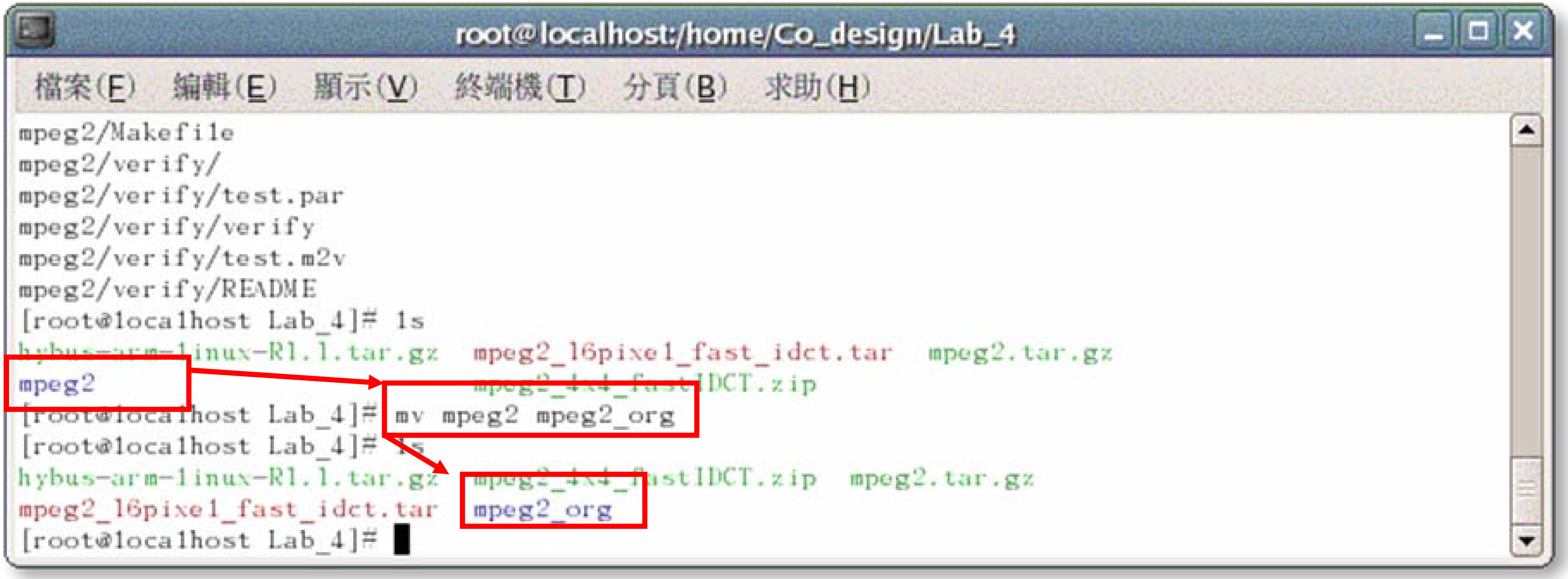
Get the “mpeg2” Folder

# 4.2 MPEG-2 Decoder Software

## ■ Decompress Source Code Package

- Modify “mpeg2” to “mpeg2_org”
  - Lab_4>ls
  - Lab_4>mv mpeg2 mpeg4_org

Avoid to Confuse with the other Source Code

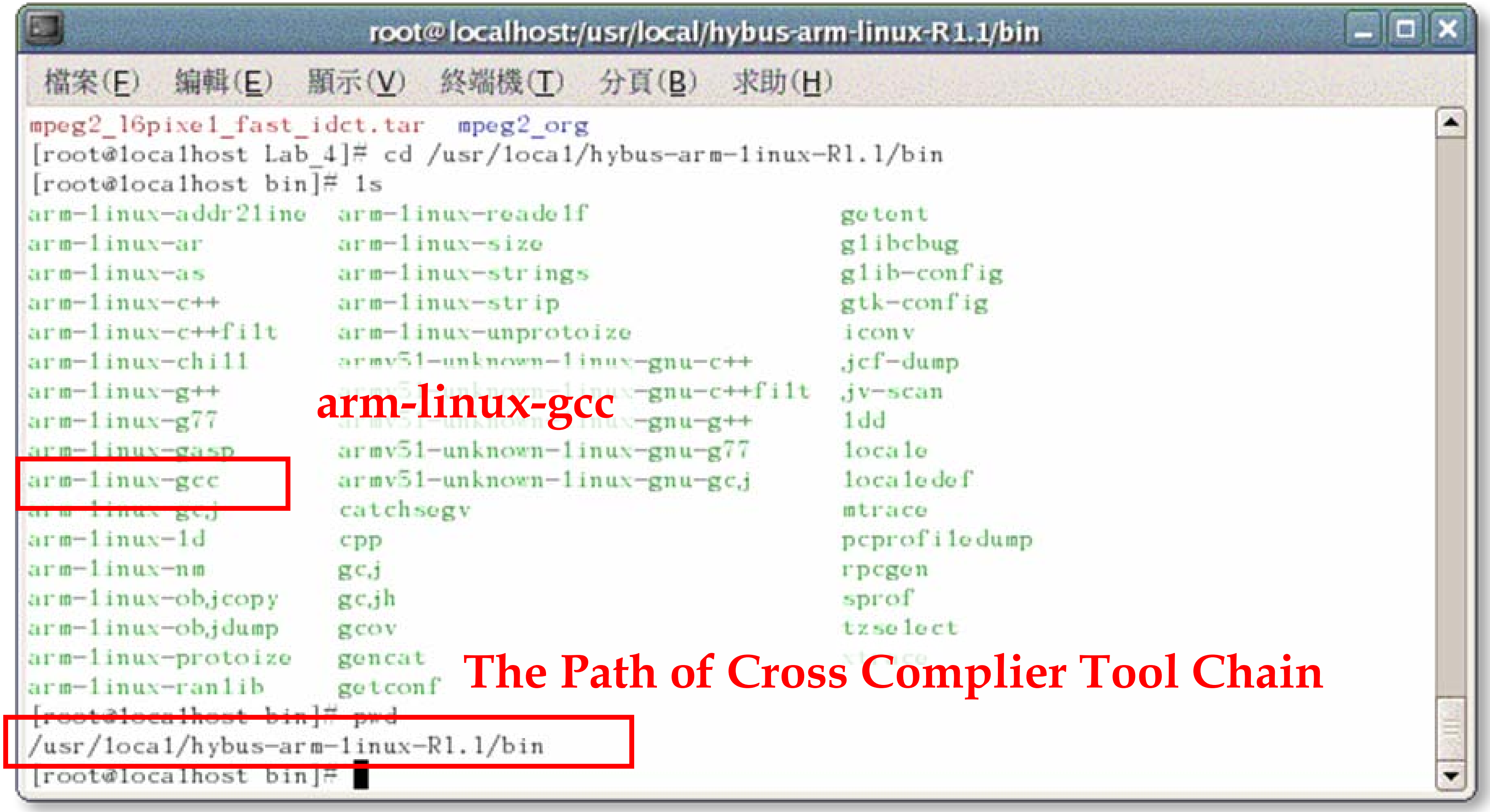

root@localhost:/home/Co_design/Lab_4
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
mpeg2/Makefile
mpeg2/verify/
mpeg2/verify/test.par
mpeg2/verify/verify
mpeg2/verify/test.m2v
mpeg2/verify/README
[root@localhost Lab_4]# ls
hybus-arm-linux-R1.1.tar.gz  mpeg2_16pixel_fast_idct.tar  mpeg2.tar.gz
mpeg2                        mpeg2_4x4_fastIDCT.zip
[root@localhost Lab_4]# mv mpeg2 mpeg2_org
[root@localhost Lab_4]# ls
hybus-arm-linux-R1.1.tar.gz  mpeg2_4x4_fastIDCT.zip  mpeg2.tar.gz
mpeg2_16pixel_fast_idct.tar  mpeg2_org
[root@localhost Lab_4]# █
```

# 4.2 MPEG-2 Decoder Software

## ■ Compile the MPEG2 Source Code

- ◆ Cross-compile Tool Path:
  - ■ /usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc


root@localhost:/usr/local/hybus-arm-linux-R1.1/bin
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
mpeg2_16pixel_fast_idct.tar  mpeg2_org
[root@localhost Lab_4]# cd /usr/local/hybus-arm-linux-R1.1/bin
[root@localhost bin]# ls
arm-linux-addr2line   arm-linux-readelf                    getent
arm-linux-ar          arm-linux-size                       glibcbug
arm-linux-as          arm-linux-strings                    glib-config
arm-linux-c++         arm-linux-strip                      gtk-config
arm-linux-c++filt     arm-linux-unprotoize                 iconv
arm-linux-chill       armv5l-unknown-linux-gnu-c++         jcf-dump
arm-linux-g++         armv5l-unknown-linux-gnu-c++filt     jv-scan
arm-linux-g77         armv5l-unknown-linux-gnu-g++         ldd
arm-linux-gasp        armv5l-unknown-linux-gnu-g77         locale
arm-linux-gcc         armv5l-unknown-linux-gnu-gcj         localedef
arm-linux-gcj         catchsegv                            mtrace
arm-linux-ld          cpp                                  pcprofiledump
arm-linux-nm          gcj                                  rpcgen
arm-linux-objcopy     gcjh                                 sprof
arm-linux-objdump     gcov                                 tzselect
arm-linux-protoize    gencat                               xtrace
arm-linux-ranlib      getconf
[root@localhost bin]# pwd
/usr/local/hybus-arm-linux-R1.1/bin
[root@localhost bin]# █
```

arm-linux-gcc

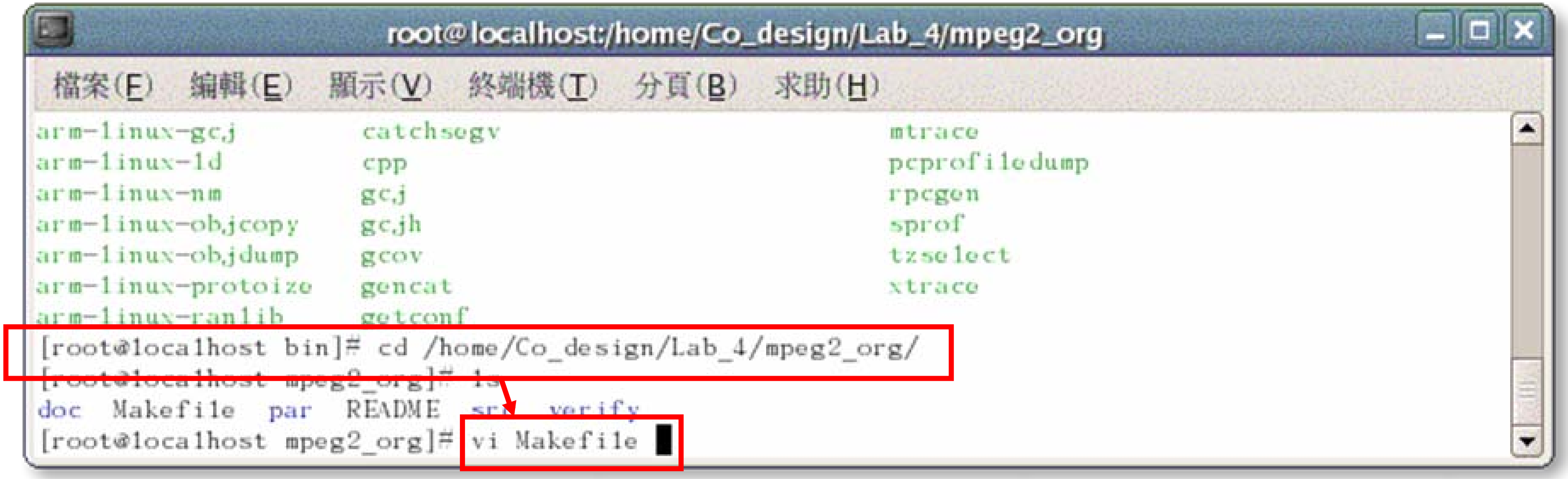
The Path of Cross Complier Tool Chain

# 4.2 MPEG-2 Decoder Software

## ■ Compile the MPEG2 Source Code

- **Modify the Makefile for MPEG-2 Source Code**
  - #>cd /home/Co_design/Lab_4/mpeg2_org/
  - mpeg2_org>ls
  - mpeg2_org>vi Makefile

**Open the Makefile to Modify**



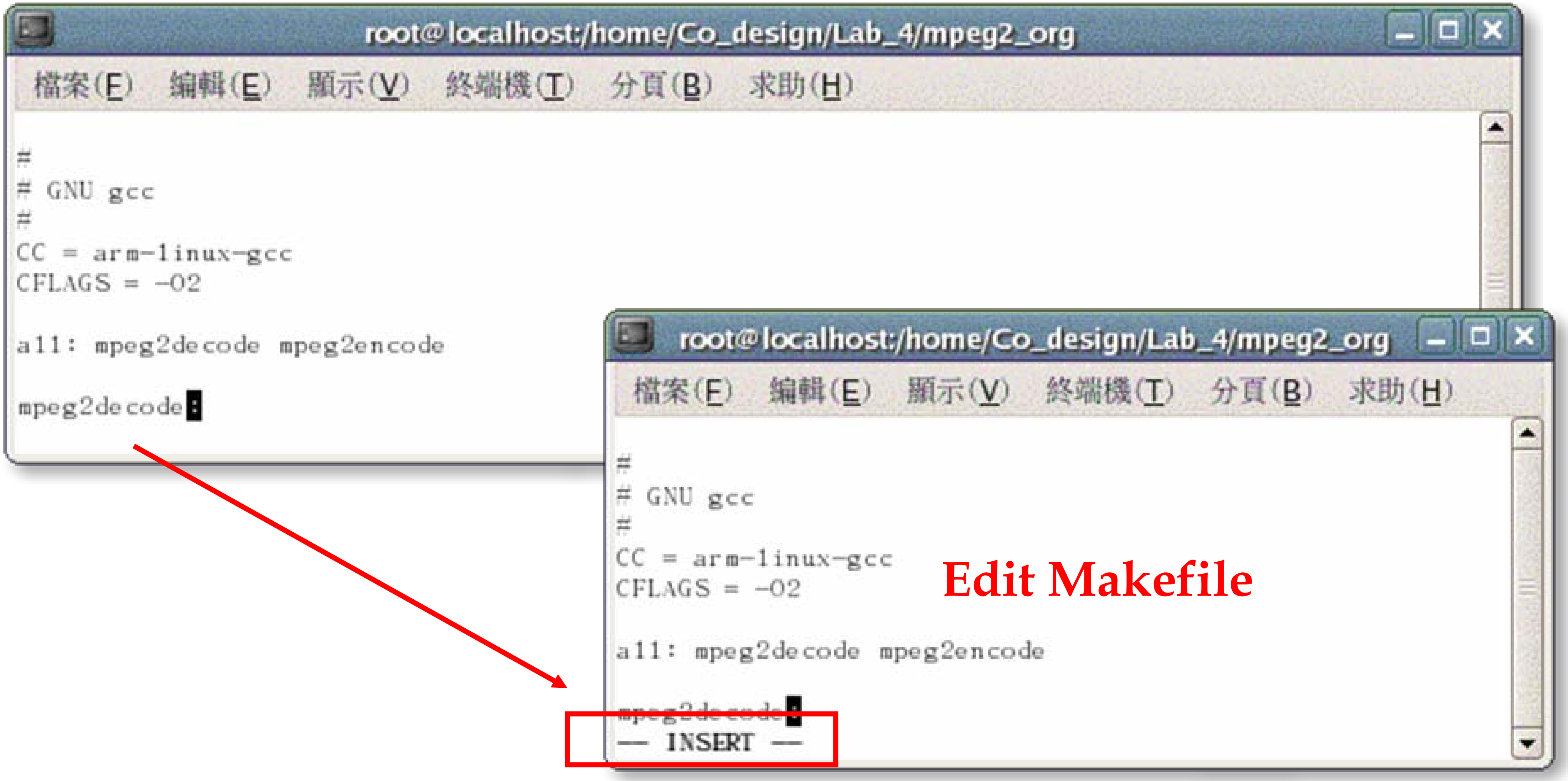
root@localhost:/home/Co_design/Lab_4/mpeg2_org
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
arm-linux-gcj            catchsegv                          mtrace
arm-linux-ld             cpp                                pcprofiledump
arm-linux-nm             gcj                                rpcgen
arm-linux-objcopy        gcjh                               sprof
arm-linux-objdump        gcov                               tzselect
arm-linux-protoize       gencat                             xtrace
arm-linux-ranlib         getconf
[root@localhost bin]# cd /home/Co_design/Lab_4/mpeg2_org/
[root@localhost mpeg2_org]# ls
doc  Makefile  par  README  src  verify
[root@localhost mpeg2_org]# vi Makefile
```

# 4.2 MPEG-2 Decoder Software

## ■ Compile the MPEG2 Source Code

- Modify Makefile
  - Type “i” to Edit Makefile



root@localhost:/home/Co_design/Lab_4/mpeg2_org
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

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

all: mpeg2decode mpeg2encode

mpeg2decode:
```

root@localhost:/home/Co_design/Lab_4/mpeg2_org
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

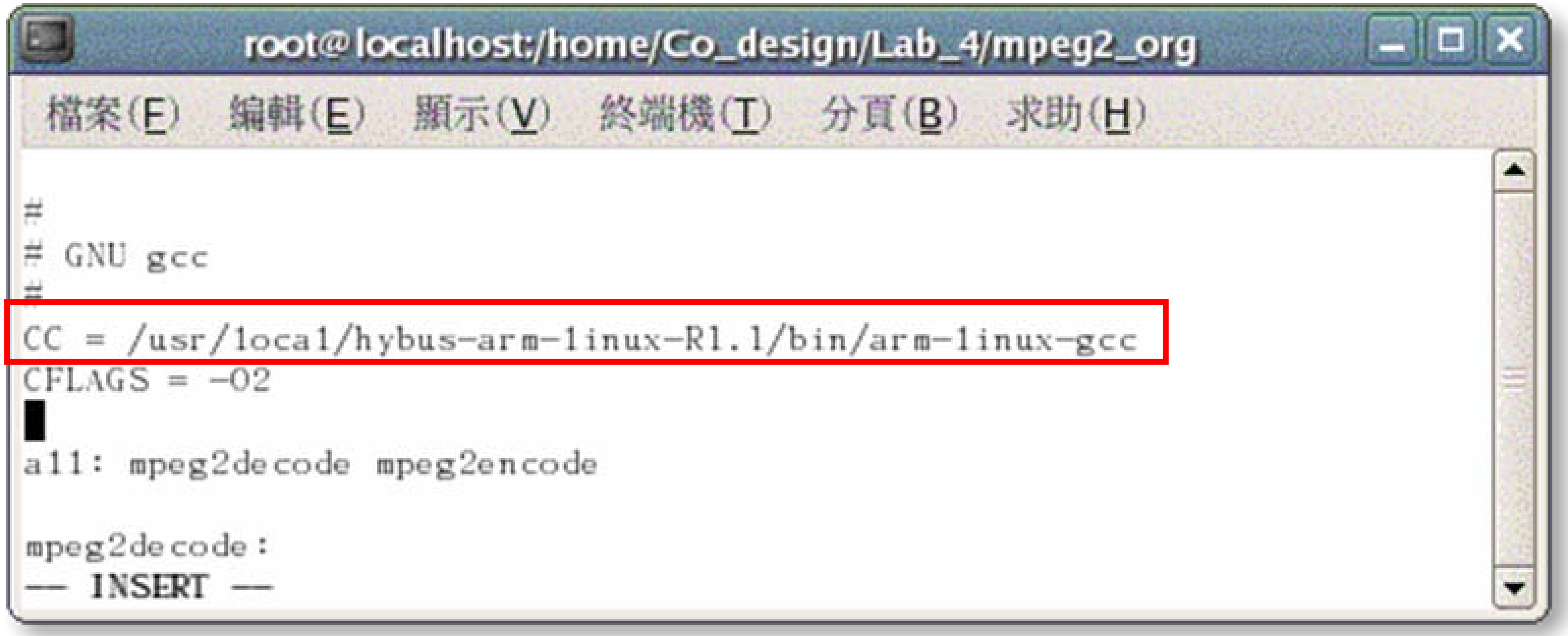
```
#
# GNU gcc
#
CC = arm-linux-gcc
CFLAGS = -O2

all: mpeg2decode mpeg2encode

mpeg2decode:
-- INSERT --
```

Edit Makefile

# 4.2 MPEG-2 Decoder Software

## Compile the MPEG2 Source Code

- Modify the Path of Cross-compiler Tool Chain
  - Modify CC= /usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc


root@localhost:/home/Co_design/Lab_4/mpeg2_org
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

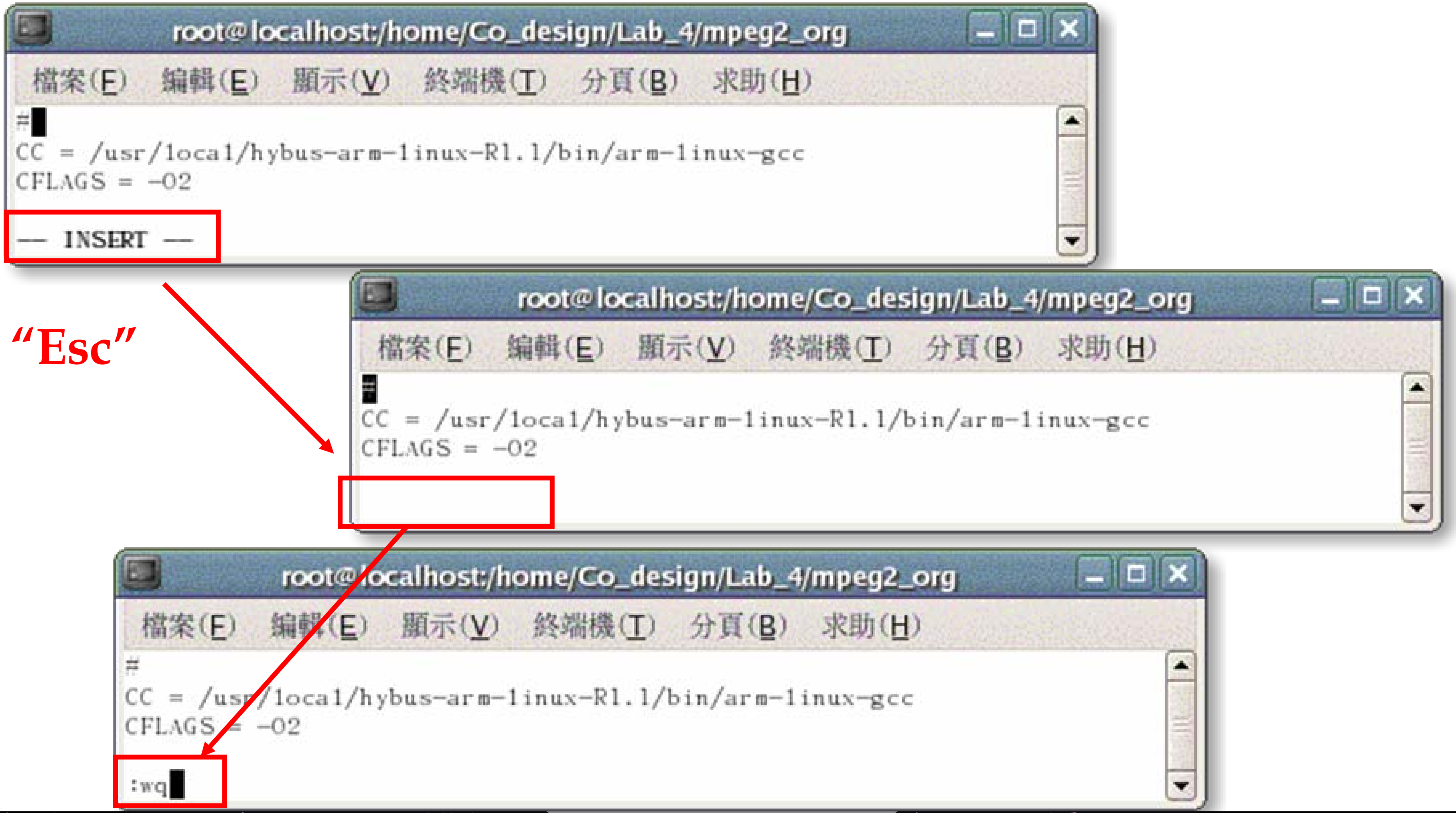
```
#
# GNU gcc
#
CC = /usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc
CFLAGS = -O2

all: mpeg2decode mpeg2encode

mpeg2decode:
-- INSERT --
```

# 4.2 MPEG-2 Decoder Software

## ■ Compile the MPEG2 Source Code

- ◆ **Push the Button “Esc” to exit Makefile**

**Type “:wq!” to Quit Makefile and Save**


root@localhost:/home/Co_design/Lab_4/mpeg2_org
檔案(F) 编輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

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

-- INSERT --
```

“Esc”

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

```

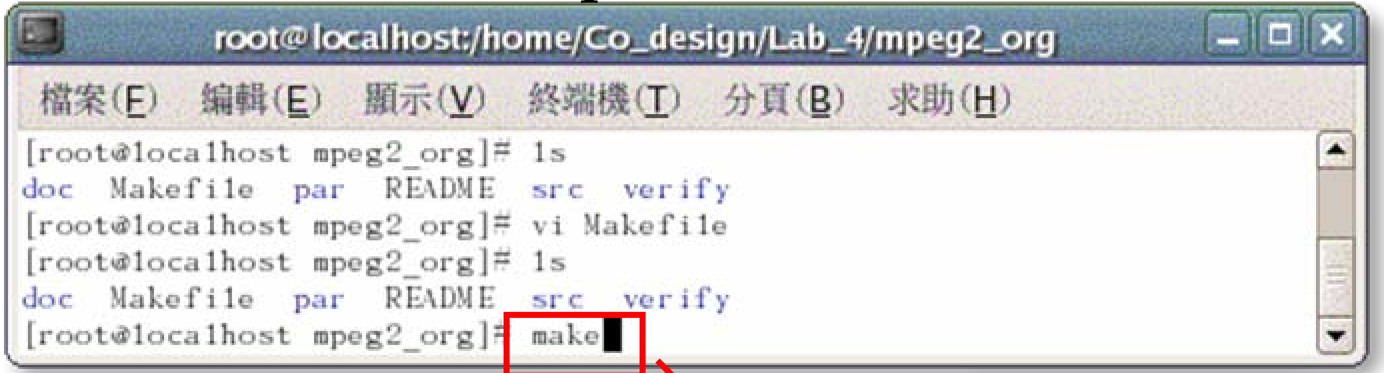
```
#
CC = /usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc
CFLAGS = -O2

:wq
```

# 4.2 MPEG-2 Decoder Software

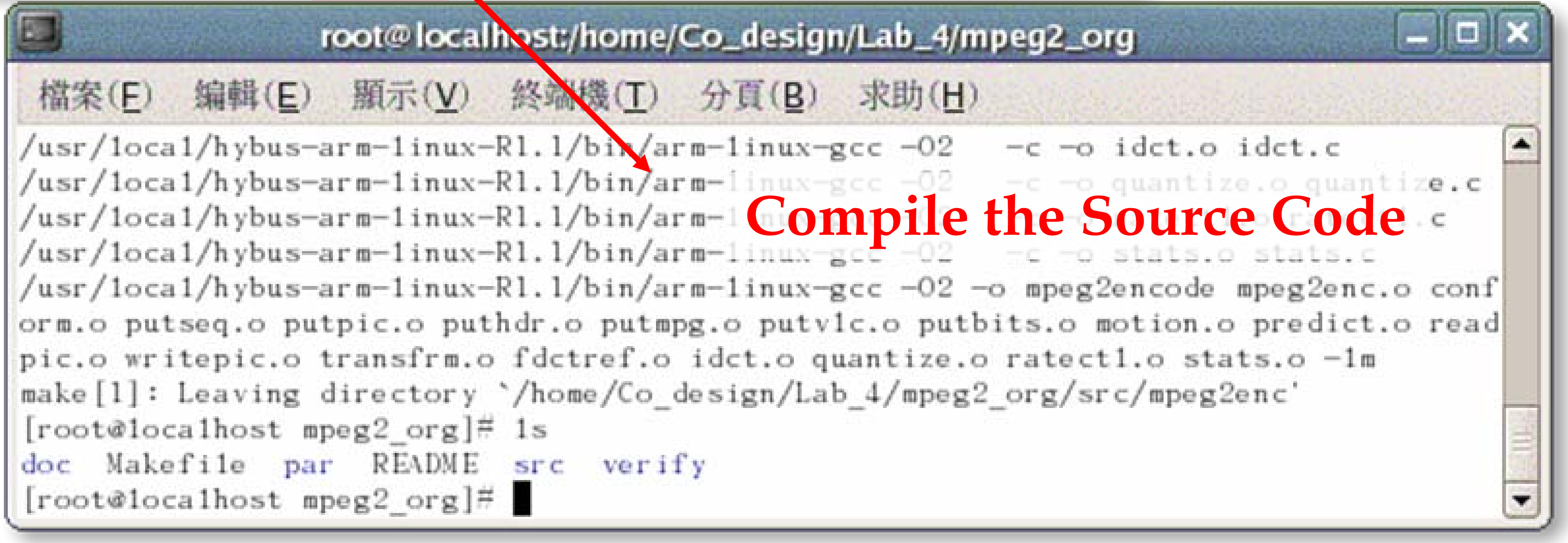
## ■ Compile the MPEG2 Source Code

- ◆ Compile the MPEG-2 Source Code
  - ■ mpeg2_org>make

Use “make” : Compile MPEG-2 Source Code


root@localhost:/home/Co_design/Lab_4/mpeg2_org
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
[root@localhost mpeg2_org]# ls
doc   Makefile   par   README   src   verify
[root@localhost mpeg2_org]# vi Makefile
[root@localhost mpeg2_org]# ls
doc   Makefile   par   README   src   verify
[root@localhost mpeg2_org]# make
```


root@localhost:/home/Co_design/Lab_4/mpeg2_org
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

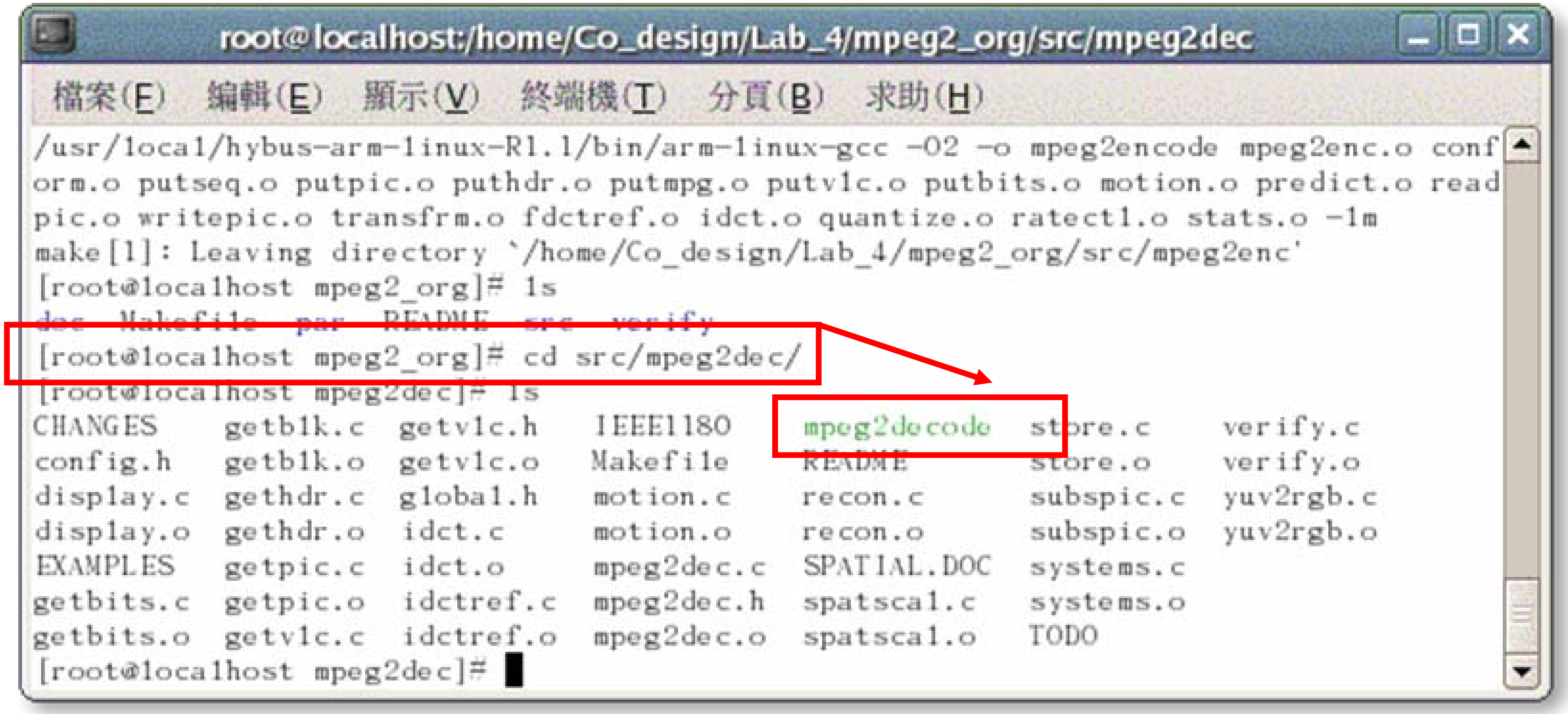
```
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o idct.o idct.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o quantize.o quantize.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-[illegible] .c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o stats.o stats.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2 -o mpeg2encode mpeg2enc.o conf
orm.o putseq.o putpic.o puthdr.o putmpg.o putvlc.o putbits.o motion.o predict.o read
pic.o writepic.o transfrm.o fdctref.o idct.o quantize.o ratectl.o stats.o -lm
make[1]: Leaving directory `/home/Co_design/Lab_4/mpeg2_org/src/mpeg2enc'
[root@localhost mpeg2_org]# ls
doc   Makefile   par   README   src   verify
[root@localhost mpeg2_org]# 
```

Compile the Source Code

# 4.2 MPEG-2 Decoder Software

## ■ Compile the MPEG2 Source Code

- MPEG2 Decoder Execute File
  - Path:/home/Co_design/mpeg2_org/src/mpeg2dec/
  - "mpeg2decode":Execute for ARM

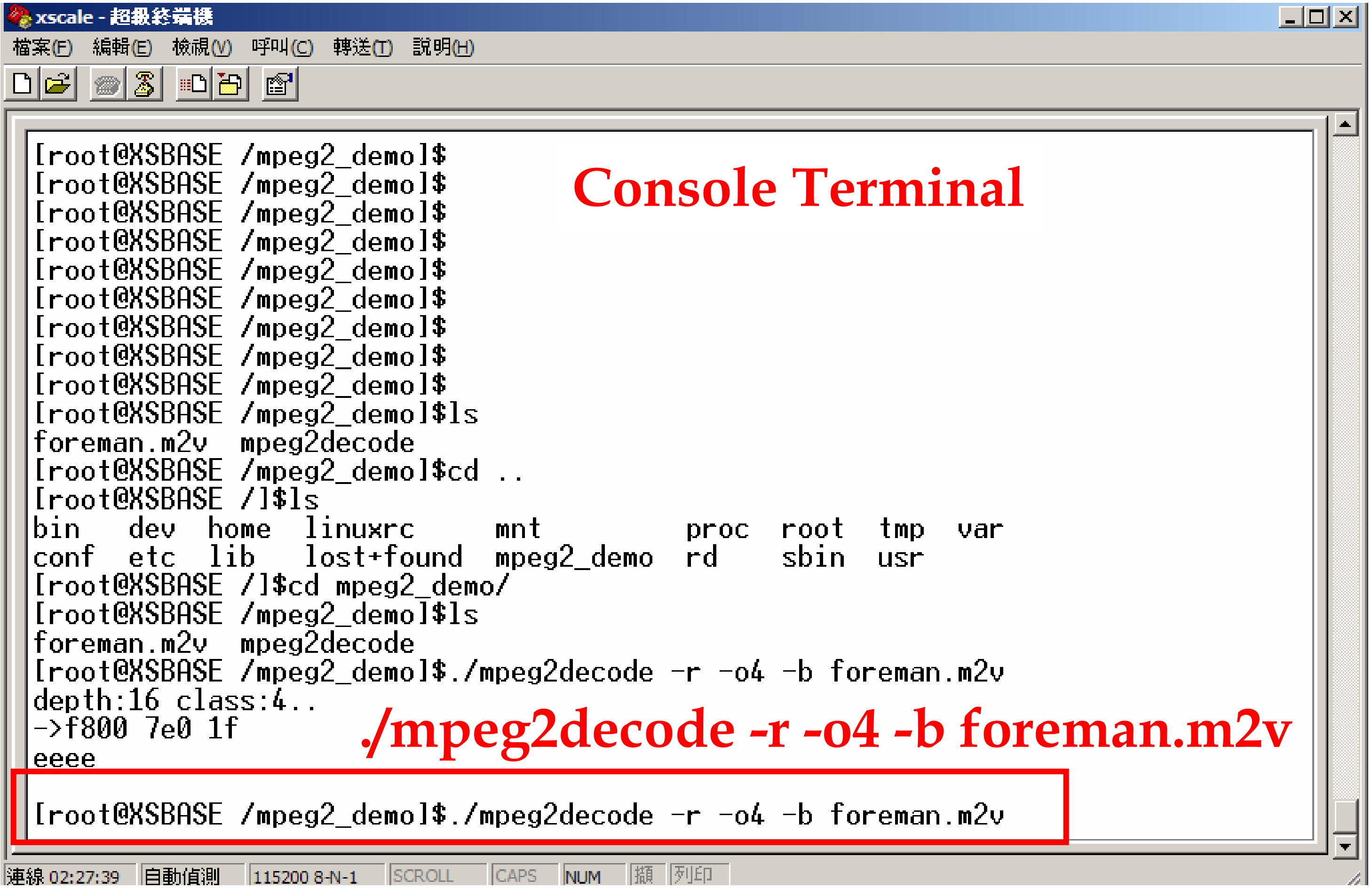

root@localhost:/home/Co_design/Lab_4/mpeg2_org/src/mpeg2dec
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2 -o mpeg2encode mpeg2enc.o conf
orm.o putseq.o putpic.o puthdr.o putmpg.o putvlc.o putbits.o motion.o predict.o read
pic.o writepic.o transfrm.o fdctref.o idct.o quantize.o ratectl.o stats.o -lm
make[1]: Leaving directory `/home/Co_design/Lab_4/mpeg2_org/src/mpeg2enc'
[root@localhost mpeg2_org]# ls
doc  Makefile  par  README  src  verify
[root@localhost mpeg2_org]# cd src/mpeg2dec/
[root@localhost mpeg2dec]# ls
CHANGES    getblk.c  getvlc.h   IEEE1180    mpeg2decode  store.c    verify.c
config.h   getblk.o  getvlc.o   Makefile    README       store.o    verify.o
display.c  gethdr.c  global.h   motion.c    recon.c      subspic.c  yuv2rgb.c
display.o  gethdr.o  idct.c     motion.o    recon.o      subspic.o  yuv2rgb.o
EXAMPLES   getpic.c  idct.o     mpeg2dec.c  SPATIAL.DOC  systems.c
getbits.c  getpic.o  idctref.c  mpeg2dec.h  spatscal.c   systems.o
getbits.o  getvlc.c  idctref.o  mpeg2dec.o  spatscal.o   TODO
[root@localhost mpeg2dec]# █
```

# 4.2 MPEG-2 Decoder Software

## ■ Execute MPEG2 Decoder

- Execute the MPEG-2 Decoder
  - [root@XSBASE /mpeg2_demo]$./mpeg2decode -r -o4 -b foreman.m2v



```
[root@XSBASE /mpeg2_demo]$
[root@XSBASE /mpeg2_demo]$
[root@XSBASE /mpeg2_demo]$
[root@XSBASE /mpeg2_demo]$
[root@XSBASE /mpeg2_demo]$
[root@XSBASE /mpeg2_demo]$
[root@XSBASE /mpeg2_demo]$
[root@XSBASE /mpeg2_demo]$
[root@XSBASE /mpeg2_demo]$
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v  mpeg2decode
[root@XSBASE /mpeg2_demo]$cd ..
[root@XSBASE /]$ls
bin   dev  home  linuxrc     mnt         proc  root  tmp  var
conf  etc  lib   lost+found  mpeg2_demo  rd    sbin  usr
[root@XSBASE /]$cd mpeg2_demo/
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v  mpeg2decode
[root@XSBASE /mpeg2_demo]$./mpeg2decode -r -o4 -b foreman.m2v
depth:16 class:4..
->f800 7e0 1f
eeee

[root@XSBASE /mpeg2_demo]$./mpeg2decode -r -o4 -b foreman.m2v
```

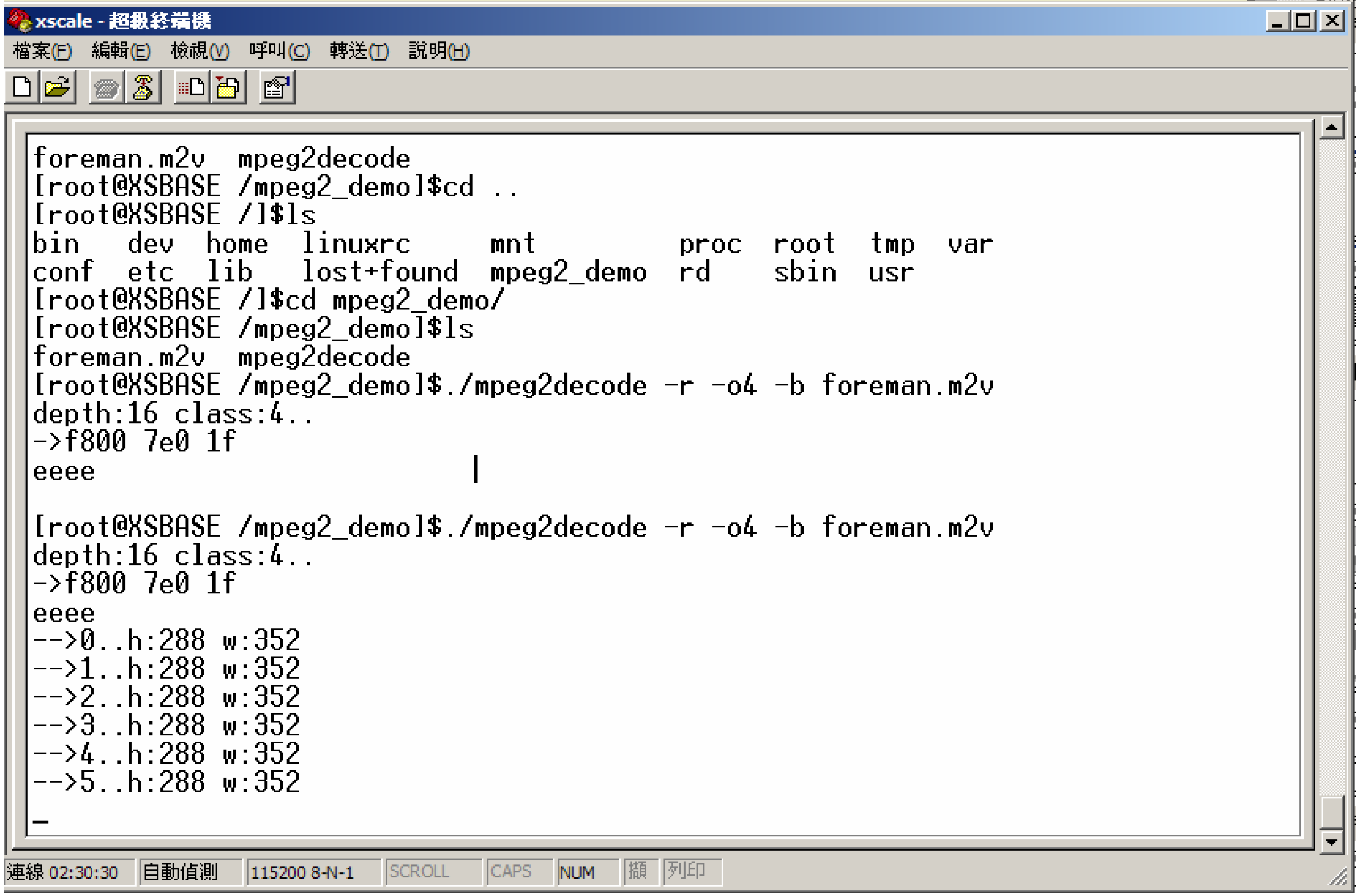
Console Terminal

./mpeg2decode -r -o4 -b foreman.m2v

# 4.2 MPEG-2 Decoder Software

## ■ Execute MPEG2 Decoder

- ◆ **MPEG-2 Decoder Execute**
  - ■ **Decoder the foreman.m2v file , and Display on TFT LCD**



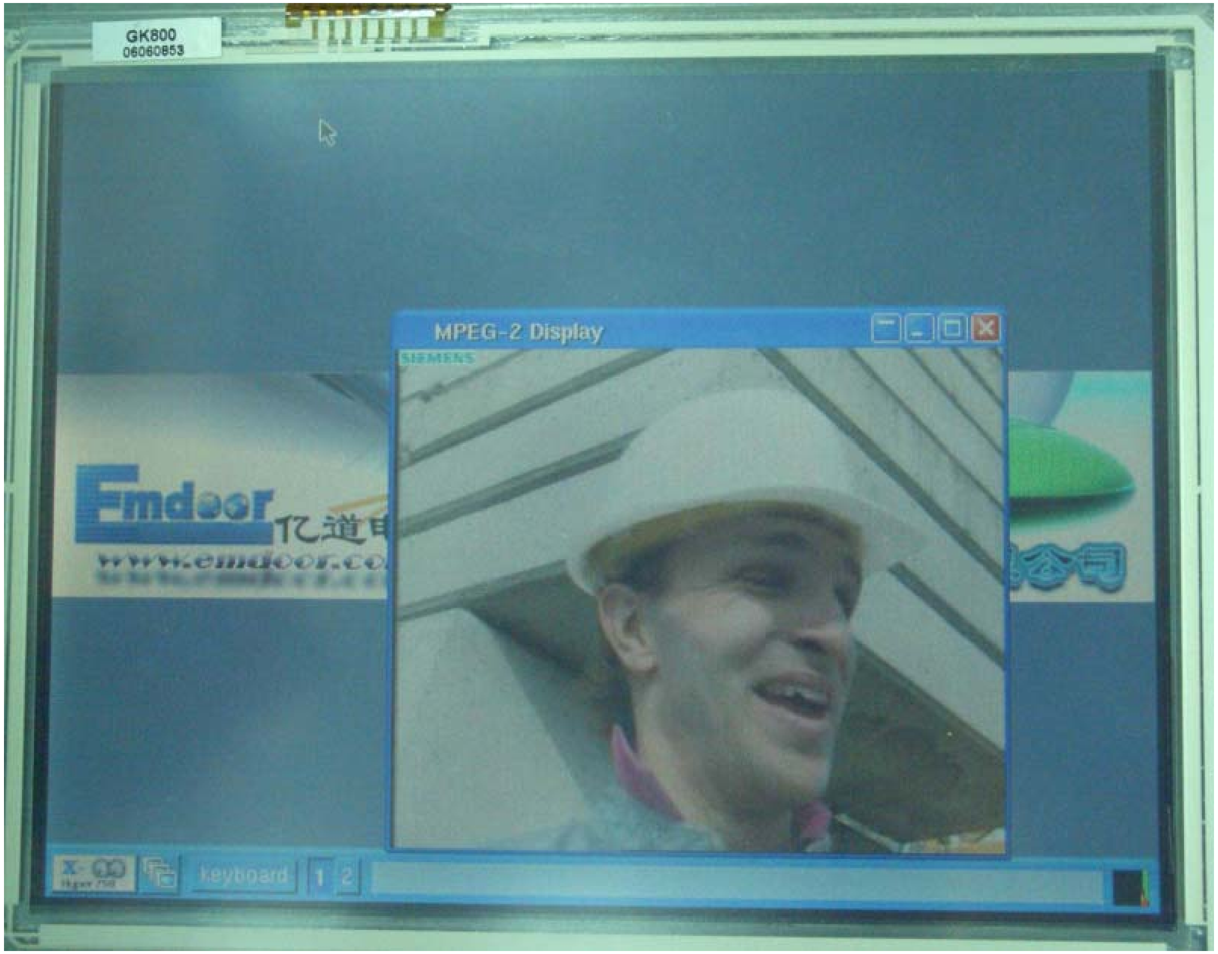
```
foreman.m2v  mpeg2decode
[root@XSBASE /mpeg2_demo]$cd ..
[root@XSBASE /]$ls
bin   dev  home  linuxrc     mnt         proc  root  tmp  var
conf  etc  lib   lost+found  mpeg2_demo  rd    sbin  usr
[root@XSBASE /]$cd mpeg2_demo/
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v  mpeg2decode
[root@XSBASE /mpeg2_demo]$./mpeg2decode -r -o4 -b foreman.m2v
depth:16 class:4..
->f800 7e0 1f
eeee

[root@XSBASE /mpeg2_demo]$./mpeg2decode -r -o4 -b foreman.m2v
depth:16 class:4..
->f800 7e0 1f
eeee
-->0..h:288 w:352
-->1..h:288 w:352
-->2..h:288 w:352
-->3..h:288 w:352
-->4..h:288 w:352
-->5..h:288 w:352

_
```

Console Terminal

# 4.2 MPEG-2 Decoder Software

## ■ Execute MPEG2 Decoder

- ◆ Display on TFT LCD



# 4.2 MPEG-2 Decoder Software

## Performance Report

- Total Execute Time
  - Execute Time = 814Seconds
  - Frame Rate = 0.36fps
    - fps: Frame Per Second

# 4.3 IDCT Source Code

4.1 Preparing

4.2 MPEG-2 Decoder Software

4.3 IDCT Source Code

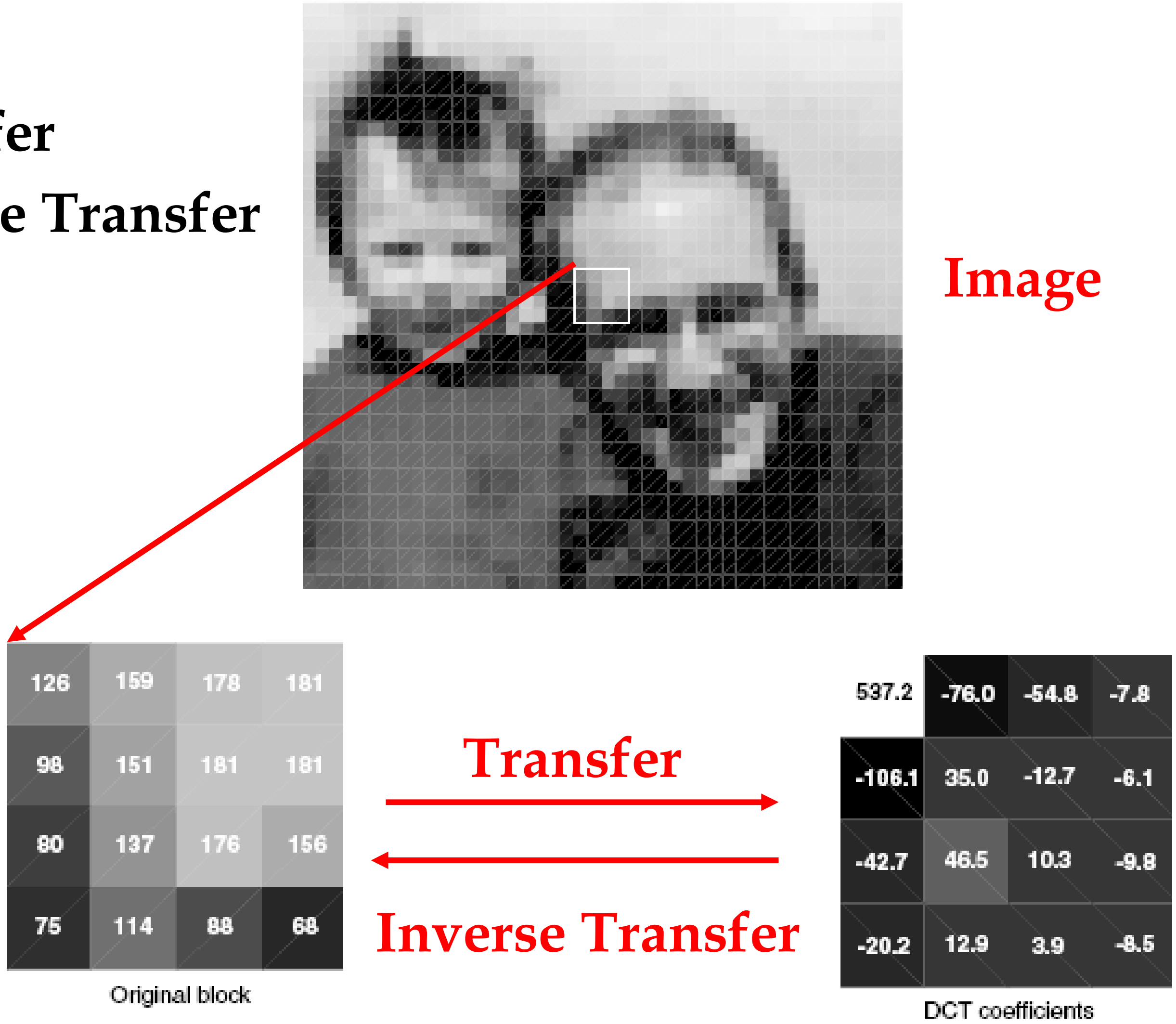
4.4 IDCT Fast Algorithm - 1

4.5 IDCT Fast Algorithm – 2

4.6 Insert Assembly Code in C Language

# 4.3 IDCT Source Code

- Transfer
  - Transfer
  - Inverse Transfer



Image

| | | | |
|---|---|---|---|
| 126 | 159 | 178 | 181 |
| 98 | 151 | 181 | 181 |
| 80 | 137 | 176 | 156 |
| 75 | 114 | 88 | 68 |

Original block

Transfer

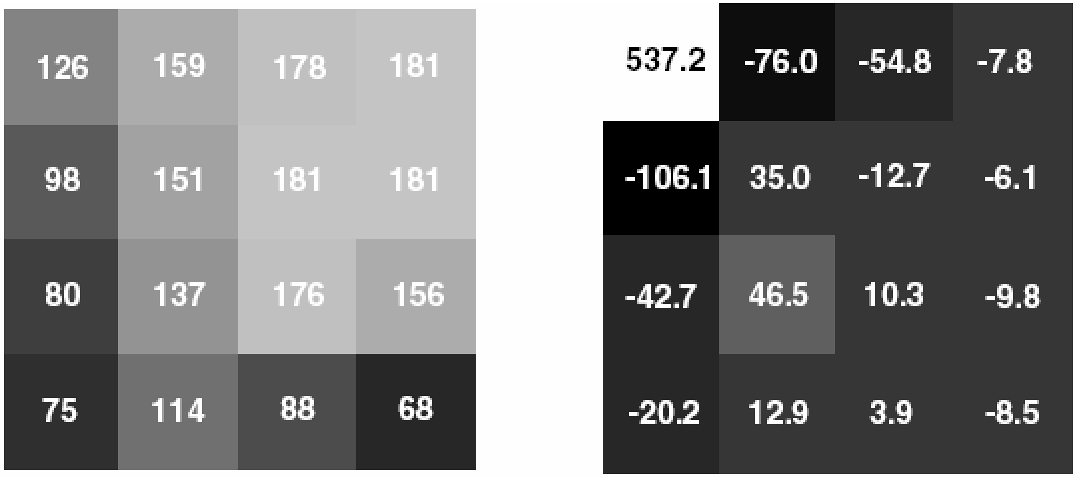
Inverse Transfer

| | | | |
|---|---|---|---|
| 537.2 | -76.0 | -54.8 | -7.8 |
| -106.1 | 35.0 | -12.7 | -6.1 |
| -42.7 | 46.5 | 10.3 | -9.8 |
| -20.2 | 12.9 | 3.9 | -8.5 |

DCT coefficients

# 4.3 IDCT Source Code

- DCT / IDCT
  - DCT : Discrete Cosine Transform
  - IDCT: Inverse Discrete Cosine Transform

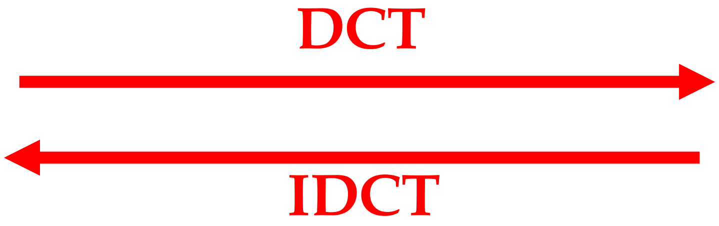


| | | | |
|---|---|---|---|
| 126 | 159 | 178 | 181 |
| 98 | 151 | 181 | 181 |
| 80 | 137 | 176 | 156 |
| 75 | 114 | 88 | 68 |

Original block

| | | | |
|---|---|---|---|
| 537.2 | -76.0 | -54.8 | -7.8 |
| -106.1 | 35.0 | -12.7 | -6.1 |
| -42.7 | 46.5 | 10.3 | -9.8 |
| -20.2 | 12.9 | 3.9 | -8.5 |

DCT coefficients



# 4.3 IDCT Source Code

- **Forward DCT (FDCT) of an N x N sample block**

$$Y = AXA^T$$

$$Y_{xy} = C_x C_y \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{ij} \cos\frac{(2j+1)y\pi}{2N} \cos\frac{(2i+1)x\pi}{2N} \qquad A^T$$

- **Inverse DCT (IDCT)**

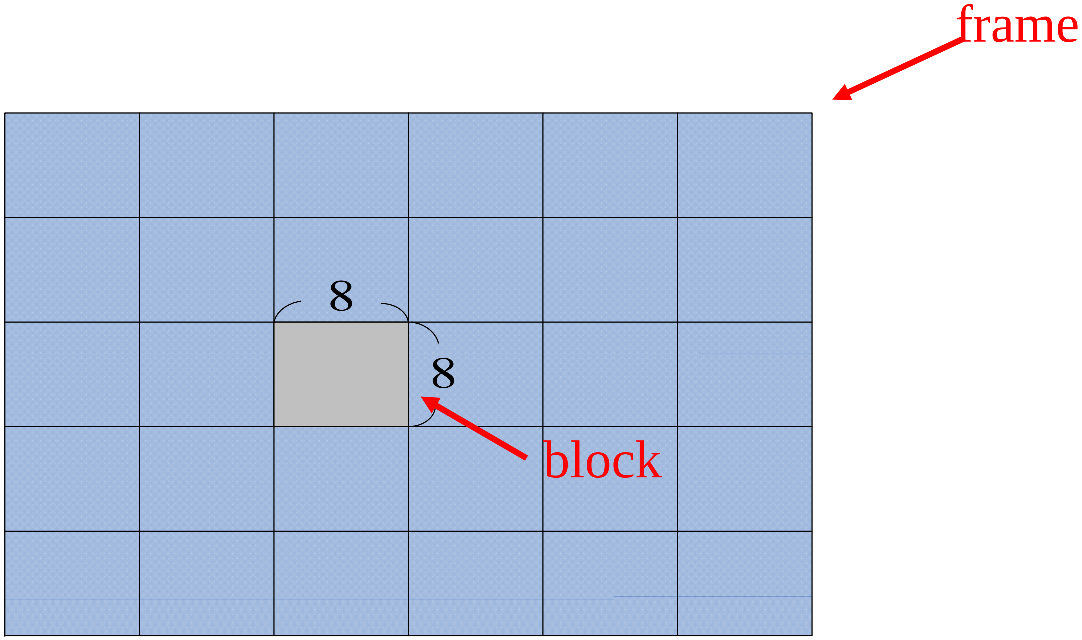
$$X = A^T Y A$$

where X is a matrix of samples ,Y is a matrix of coefficients and A is an N×N transform matrix.

$$X_{ij} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C_x C_y Y_{xy} \cos\frac{(2j+1)y\pi}{2N} \cos\frac{(2i+1)x\pi}{2N}$$

# 4.3 IDCT Source Code

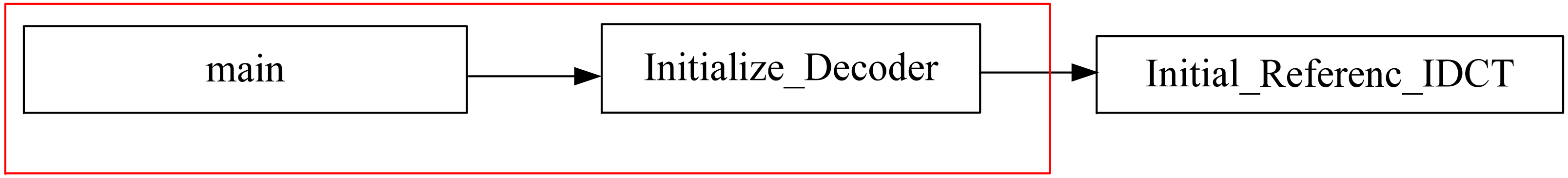
- Initialize_Reference_IDCT(): Initialize the “Cosine” Coefficient
- Reference_IDCT(block): the Function of 2D-IDCT



# 4.3 IDCT Source Code

- IDCT Function
  - The Initial Function: Initial the IDCT Coefficient Table

Initial IDCT coefficient table



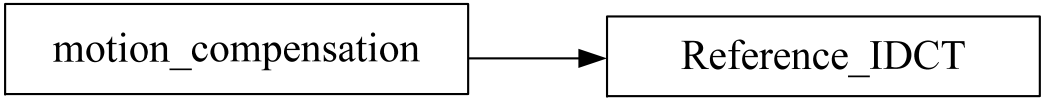
main → Initialize_Decoder → Initial_Referenc_IDCT

mpeg2dec.c

idctref.c

  - Call IDCT Function

IDCT fuction



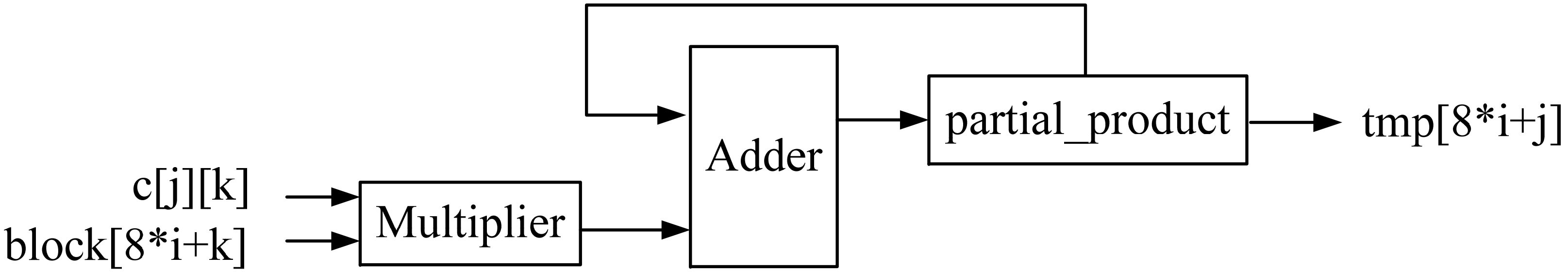
motion_compensation → Reference_IDCT

get_pic.c

idct.c

# 4.3 IDCT Source Code

- **Reference_IDCT()**



$$\begin{bmatrix} c[0][0] & c[0][1] & c[0][2] & c[0][3] & c[0][4] & c[0][5] & c[0][6] & c[0][7] \\ c[1][0] & c[1][1] & c[1][2] & c[1][3] & c[1][4] & c[1][5] & c[1][6] & c[1][7] \\ c[2][0] & c[2][1] & c[2][2] & c[2][3] & c[2][4] & c[2][5] & c[2][6] & c[2][7] \\ c[3][0] & c[3][1] & c[3][2] & c[3][3] & c[3][4] & c[3][5] & c[3][6] & c[3][7] \\ c[4][0] & c[4][1] & c[4][2] & c[4][3] & c[4][4] & c[4][5] & c[4][6] & c[4][7] \\ c[5][0] & c[5][1] & c[5][2] & c[5][3] & c[5][4] & c[5][5] & c[5][6] & c[5][7] \\ c[6][0] & c[6][1] & c[6][2] & c[6][3] & c[6][4] & c[6][5] & c[6][6] & c[6][7] \\ c[7][0] & c[7][1] & c[7][2] & c[7][3] & c[7][4] & c[7][5] & c[7][6] & c[7][7] \end{bmatrix} \begin{bmatrix} block[8*i+0] \\ block[8*i+1] \\ block[8*i+2] \\ block[8*i+3] \\ block[8*i+4] \\ block[8*i+5] \\ block[8*i+6] \\ block[8*i+7] \end{bmatrix} = \begin{bmatrix} tmp[8*i+0] \\ tmp[8*i+1] \\ tmp[8*i+2] \\ tmp[8*i+3] \\ tmp[8*i+4] \\ tmp[8*i+5] \\ tmp[8*i+6] \\ tmp[8*i+7] \end{bmatrix}$$

# 4.3 IDCT Source Code

- **File Name : idctref.c**

```
---------------------------------------------------------------------------
 static double c[8][8];

 void Initialize_Reference_IDCT()
 {                         //initial c[8][8]
   int freq, time;
   double scale;
   for (freq=0; freq < 8; freq++)  {
   scale = (freq == 0) ? sqrt(0.125) : 0.5;
   for (time=0; time<8; time++)
    c[freq][time] = scale*cos((PI/8.0)*freq*(time + 0.5));
   }
 }
---------------------------------------------------------------------------
```

# 4.3 IDCT Source Code

- **File Name : idctref.c**

```
---------------------------------------------------------------------------
void Reference_IDCT(block)
 short *block;
 {
   int i, j, k, v;
   double partial_product;
   double tmp[64];
   for (i=0; i<8; i++)
      for (j=0; j<8; j++)    {                         //1D-IDCT
          partial_product = 0.0;

          for (k=0; k<8; k++)
               partial_product+= c[k][j]*block[8*i+k];

            tmp[8*i+j] = partial_product;
       }
    ……
---------------------------------------------------------------------------
```

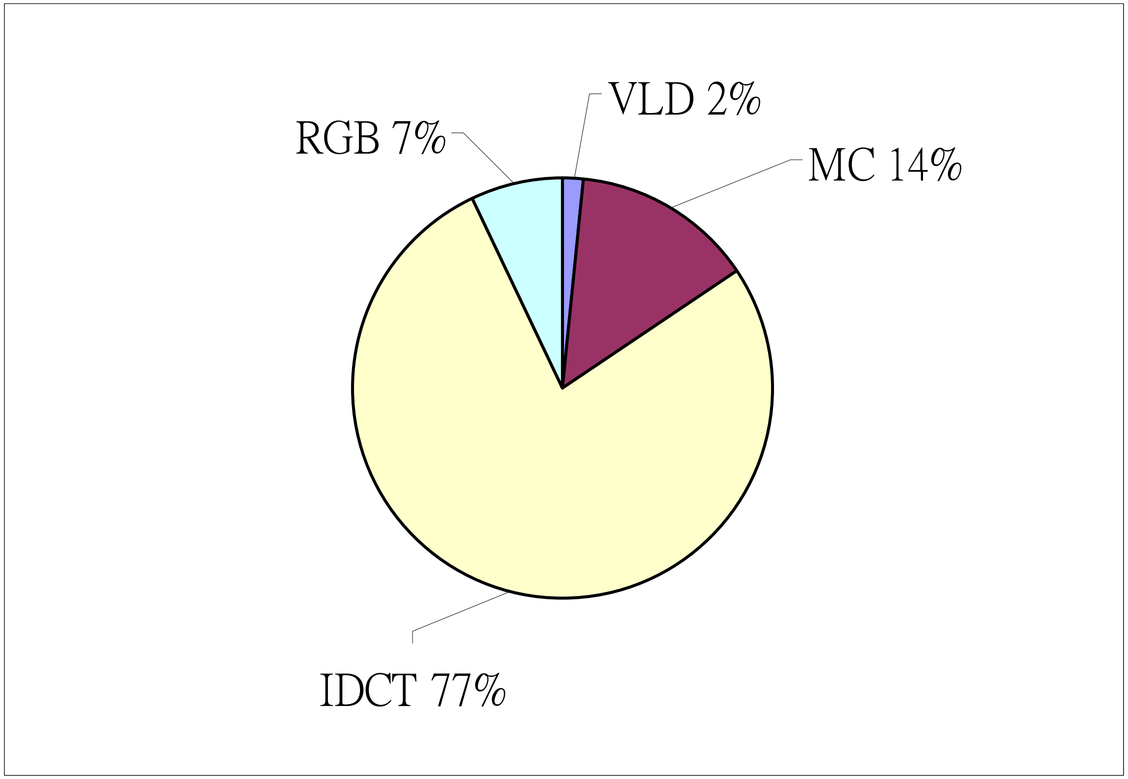
# 4.3 IDCT Source Code

- **File Name : idctref.c**

```
------------------------------------------------------------------------------
 ……
  for (j=0; j<8; j++)
     for (i=0; i<8; i++)
   {
      partial_product = 0.0; //2D-IDCT
      for (k=0; k<8; k++)
         partial_product+= c[k][i]*tmp[8*k+j];

       v = (int) floor(partial_product+0.5);
        block[8*i+j] = (v<-256) ? -256 : ((v>255) ? 255 : v); //Clip
   }
 }
------------------------------------------------------------------------------
```

# 4.3 IDCT Source Code

## ■ Performance Analysis



# 4.3 IDCT Source Code

## ■ Performance Report

- ◆ Total Execute Time
  - ■ Execute Time = 814 Seconds
  - ■ Frame Rate = 0.061fps
    - ● fps : Frame Per Second
- ◆ The Possible Bottleneck
  - ■ The Complexity of the Operation
    - ● Solution : The Improvement of Algorithm Level
  - ■ C Program
    - ● Program Coding
    - ● The Type of Data

      The Operation of Float-Point is Slow.

# 4.4 IDCT Fast Algorithm - 1

4.1 Preparing

4.2 MPEG-2 Decoder Software

4.3 IDCT Source Code

4.4 IDCT Fast Algorithm - 1

4.5 IDCT Fast Algorithm – 2

4.6 Insert Assembly Code in C Language

# 4.4 IDCT Fast Algorithm - 1

- IDCT Fast Algorithm -1
  - Modify the Operation of mathematics
  - Use Two 4x4 Matrix to Complete IDCT
  - Lossless Operation
  - MPEG-2 Source Code after Modify
    - mpeg2_4x4_fastIDCT.zip

# 4.4 IDCT Fast Algorithm - 1

- **8 x 8 IDCT**

**$X = A^TYA$ Let $B = A^TY$ ,$X = BA$**

$$\begin{bmatrix} B_0 & B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 \\ B_8 & B_9 & B_{10} & B_{11} & B_{12} & B_{13} & B_{14} & B_{15} \\ B_{16} & B_{17} & B_{18} & B_{19} & B_{20} & B_{21} & B_{22} & B_{23} \\ B_{24} & B_{25} & B_{26} & B_{27} & B_{28} & B_{29} & B_{30} & B_{31} \\ B_{32} & B_{33} & B_{34} & B_{35} & B_{36} & B_{37} & B_{38} & B_{39} \\ B_{40} & B_{41} & B_{42} & B_{43} & B_{44} & B_{45} & B_{46} & B_{47} \\ B_{48} & B_{49} & B_{50} & B_{51} & B_{52} & B_{53} & B_{54} & B_{55} \\ B_{56} & B_{57} & B_{58} & B_{59} & B_{60} & B_{61} & B_{62} & B_{63} \end{bmatrix} = \begin{bmatrix} d & a & b & c & d & e & f & g \\ d & c & f & -g & -d & -a & -b & -e \\ d & e & -f & -a & -d & g & b & c \\ d & g & -b & -e & -d & c & -f & -a \\ d & -g & -b & e & d & -c & -f & a \\ d & -e & -f & a & -d & -g & b & -c \\ d & -c & f & g & -d & a & -b & f \\ d & -a & b & -c & d & -e & f & -g \end{bmatrix} \begin{bmatrix} Y_0 & Y_1 & Y_2 & Y_3 & Y_4 & Y_5 & Y_6 & Y_7 \\ Y_8 & Y_9 & Y_{10} & Y_{11} & Y_{12} & Y_{13} & Y_{14} & Y_{15} \\ Y_{16} & Y_{17} & Y_{18} & Y_{19} & Y_{20} & Y_{21} & Y_{22} & Y_{23} \\ Y_{24} & Y_{25} & Y_{26} & Y_{27} & Y_{28} & Y_{29} & Y_{30} & Y_{31} \\ Y_{32} & Y_{33} & Y_{34} & Y_{35} & Y_{36} & Y_{37} & Y_{38} & Y_{39} \\ Y_{40} & Y_{41} & Y_{42} & Y_{43} & Y_{44} & Y_{45} & Y_{46} & Y_{47} \\ Y_{48} & Y_{49} & Y_{50} & Y_{51} & Y_{52} & Y_{53} & Y_{54} & Y_{55} \\ Y_{56} & Y_{57} & Y_{58} & Y_{59} & Y_{60} & Y_{61} & Y_{62} & Y_{63} \end{bmatrix}$$

$$B \qquad = \qquad A^T \qquad\qquad Y$$

$$\begin{bmatrix} X_0 & X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 \\ X_8 & X_9 & X_{10} & X_{11} & X_{12} & X_{13} & X_{14} & X_{15} \\ X_{16} & X_{17} & X_{18} & X_{19} & X_{20} & X_{21} & X_{22} & X_{23} \\ X_{24} & X_{25} & X_{26} & X_{27} & X_{28} & X_{29} & X_{30} & X_{31} \\ X_{32} & X_{33} & X_{34} & X_{35} & X_{36} & X_{37} & X_{38} & X_{39} \\ X_{40} & X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} & X_{47} \\ X_{48} & X_{49} & X_{50} & X_{51} & X_{52} & X_{53} & X_{54} & X_{55} \\ X_{56} & X_{57} & X_{58} & X_{59} & X_{60} & X_{61} & X_{62} & X_{63} \end{bmatrix} = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 \\ B_8 & B_9 & B_{10} & B_{11} & B_{12} & B_{13} & B_{14} & B_{15} \\ B_{16} & B_{17} & B_{18} & B_{19} & B_{20} & B_{21} & B_{22} & B_{23} \\ B_{24} & B_{25} & B_{26} & B_{27} & B_{28} & B_{29} & B_{30} & B_{31} \\ B_{32} & B_{33} & B_{34} & B_{35} & B_{36} & B_{37} & B_{38} & B_{39} \\ B_{40} & B_{41} & B_{42} & B_{43} & B_{44} & B_{45} & B_{46} & B_{47} \\ B_{48} & B_{49} & B_{50} & B_{51} & B_{52} & B_{53} & B_{54} & B_{55} \\ B_{56} & B_{57} & B_{58} & B_{59} & B_{60} & B_{61} & B_{62} & B_{63} \end{bmatrix} \begin{bmatrix} d & d & d & d & d & d & d & d \\ a & c & e & g & -g & -e & -c & -a \\ b & f & -f & -b & -b & -f & f & b \\ c & -g & -a & -e & e & a & g & -c \\ d & -d & -d & d & d & -d & -d & d \\ e & -a & g & c & -c & -g & a & -e \\ f & -b & b & -f & -f & b & -b & f \\ g & -e & c & -a & a & -c & e & -g \end{bmatrix}$$

$$X \qquad = \qquad B \qquad\qquad A$$

# 4.4 IDCT Fast Algorithm - 1

- **1-D IDCT ($B = A^T Y$)**

$$
\begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \end{bmatrix}
=
\left[\begin{array}{cccc|cccc}
d & a & b & c & d & e & f & g \\
d & c & f & -g & -d & -a & -b & -e \\
d & e & -f & -a & -d & g & b & c \\
d & g & -b & -e & -d & c & -f & -a \\
\hline
d & -g & -b & e & d & -c & -f & a \\
d & -e & -f & a & -d & -g & b & -c \\
d & -c & f & g & -d & a & -b & f \\
d & -a & b & -c & d & -e & f & -g
\end{array}\right]
\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \\ Y_7 \end{bmatrix}
$$

$$B \quad = \quad A^T \quad Y$$

$$B_0 = dY_0 + aY_1 + bY_2 + cY_3 + dY_4 + eY_5 + fY_6 + gY_7$$

$$B_1 = dY_0 + cY_1 + fY_2 - gY_3 - dY_4 - aY_5 - bY_6 - eY_7$$

$$B_2 = dY_0 + eY_1 - fY_2 - aY_3 - dY_4 + gY_5 + bY_6 + cY_7$$

$$B_3 = dY_0 + gY_1 - bY_2 - eY_3 - dY_4 + cY_5 - fY_6 - aY_7$$

$$B_4 = dY_0 - gY_1 - bY_2 + eY_3 + dY_4 - cY_5 - fY_6 + aY_7$$

$$B_5 = dY_0 - eY_1 - fY_2 + aY_3 - dY_4 - gY_5 + bY_6 - cY_7$$

$$B_6 = dY_0 - cY_1 + fY_2 + gY_3 - dY_4 + aY_5 - bY_6 + eY_7$$

$$B_7 = dY_0 - aY_1 + bY_2 - cY_3 + dY_4 - eY_5 + fY_6 - gY_7$$

# 4.4 IDCT Fast Algorithm - 1

- **Add**

$$B_0 + B_7 = (dY_0 + bY_2 + dY_4 + fY_6) \times 2$$
$$B_1 + B_6 = (dY_0 + fY_2 - dY_4 - bY_6) \times 2$$
$$B_2 + B_5 = (dY_0 - fX_2 - dY_4 + bY_6) \times 2$$
$$B_3 + B_4 = (dY_0 - bY_2 + dY_4 - fY_6) \times 2$$

- **Sub**

$$B_0 - B_7 = (aY_1 + cY_3 + eY_5 + gY_7) \times 2$$
$$B_1 - B_6 = (cY_1 - gY_3 - aY_5 - eY_7) \times 2$$
$$B_2 - B_5 = (eY_1 - aY_3 + gY_5 + cY_7) \times 2$$
$$B_3 - B_4 = (gY_1 - eY_3 + cY_5 - aY_7) \times 2$$

$$\frac{1}{2}\begin{bmatrix} B_0 + B_7 \\ B_1 + B_6 \\ B_2 + B_5 \\ B_3 + B_4 \end{bmatrix} = \begin{bmatrix} \mathrm{d} & \mathrm{b} & \mathrm{d} & \mathrm{f} \\ \mathrm{d} & \mathrm{f} & \mathrm{-d} & \mathrm{-b} \\ \mathrm{d} & \mathrm{-f} & \mathrm{-d} & \mathrm{b} \\ \mathrm{d} & \mathrm{-b} & \mathrm{d} & \mathrm{-f} \end{bmatrix} \begin{bmatrix} Y_0 \\ Y_2 \\ Y_4 \\ Y_6 \end{bmatrix}$$

$$\frac{1}{2}\begin{bmatrix} B_0 - B_7 \\ B_1 - B_6 \\ B_2 - B_5 \\ B_3 - B_4 \end{bmatrix} = \begin{bmatrix} \mathrm{a} & \mathrm{c} & \mathrm{e} & \mathrm{g} \\ \mathrm{c} & \mathrm{-g} & \mathrm{-a} & \mathrm{-e} \\ \mathrm{e} & \mathrm{-a} & \mathrm{g} & \mathrm{c} \\ \mathrm{g} & \mathrm{-e} & \mathrm{c} & \mathrm{-a} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_3 \\ Y_5 \\ Y_7 \end{bmatrix}$$

# 4.4 IDCT Fast Algorithm - 1

- **1-D IDCT (B = A$^T$Y)**

$$
\begin{aligned}
B_0 &= dY_0 + aY_1 + bY_2 + cY_3 + dY_4 + eY_5 + fY_6 + gY_7 \\
B_1 &= dY_0 + cY_1 + fY_2 - gY_3 - dY_4 - aY_5 - bY_6 - eY_7 \\
B_2 &= dY_0 + eY_1 - fY_2 - aY_3 - dY_4 + gY_5 + bY_6 + cY_7 \\
B_3 &= dY_0 + gY_1 - bY_2 - eY_3 - dY_4 + cY_5 - fY_6 - aY_7 \\
B_4 &= dY_0 - gY_1 - bY_2 + eY_3 + dY_4 - cY_5 - fY_6 + aY_7 \\
B_5 &= dY_0 - eY_1 - fY_2 + aY_3 - dY_4 - gY_5 + bY_6 - cY_7 \\
B_6 &= dY_0 - cY_1 + fY_2 + gY_3 - dY_4 + aY_5 - bY_6 + eY_7 \\
B_7 &= dY_0 - aY_1 + bY_2 - cY_3 + dY_4 - eY_5 + fY_6 - gY_7
\end{aligned}
$$

$$
\begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix} = {}^{1}\!/_{2} \begin{bmatrix} B_0 + B_7 \\ B_1 + B_6 \\ B_2 + B_5 \\ B_3 + B_4 \end{bmatrix} + {}^{1}\!/_{2} \begin{bmatrix} B_0 - B_7 \\ B_1 - B_6 \\ B_2 - B_5 \\ B_3 - B_4 \end{bmatrix} \Rightarrow \begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix} = {}^{1}\!/_{2} \begin{bmatrix} Ve0 \\ Ve1 \\ Ve2 \\ Ve3 \end{bmatrix} + {}^{1}\!/_{2} \begin{bmatrix} Vo0 \\ Vo1 \\ Vo2 \\ Vo3 \end{bmatrix} = {}^{1}\!/_{2} \begin{bmatrix} Ve0 + Vo0 \\ Ve1 + Vo1 \\ Ve2 + Vo2 \\ Ve3 + Vo2 \end{bmatrix}
$$

$$
\begin{bmatrix} B_7 \\ B_6 \\ B_5 \\ B_4 \end{bmatrix} = {}^{1}\!/_{2} \begin{bmatrix} B_0 + B_7 \\ B_1 + B_6 \\ B_2 + B_5 \\ B_3 + B_4 \end{bmatrix} - {}^{1}\!/_{2} \begin{bmatrix} B_0 - B_7 \\ B_1 - B_6 \\ B_2 - B_5 \\ B_3 - B_4 \end{bmatrix} \Rightarrow \begin{bmatrix} B_7 \\ B_6 \\ B_5 \\ B_4 \end{bmatrix} = {}^{1}\!/_{2} \begin{bmatrix} Ve0 \\ Ve1 \\ Ve2 \\ Ve3 \end{bmatrix} - {}^{1}\!/_{2} \begin{bmatrix} Vo0 \\ Vo1 \\ Vo2 \\ Vo3 \end{bmatrix} = {}^{1}\!/_{2} \begin{bmatrix} Ve0 - Vo0 \\ Ve1 - Vo1 \\ Ve2 - Vo2 \\ Ve3 - Vo2 \end{bmatrix}
$$

# 4.4 IDCT Fast Algorithm - 1

- **1-D IDCT (B = A$^T$Y)**
  - **The 1D-IDCT can Become the 4x4 Matrix**

$$\begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} \text{d} & \text{b} & \text{d} & \text{f} \\ \text{d} & \text{f} & \text{-d} & \text{-b} \\ \text{d} & \text{-f} & \text{-d} & \text{b} \\ \text{d} & \text{-b} & \text{d} & \text{-f} \end{bmatrix} \begin{bmatrix} Y_0 \\ Y_2 \\ Y_4 \\ Y_6 \end{bmatrix} + \begin{bmatrix} \text{a} & \text{c} & \text{e} & \text{g} \\ \text{c} & \text{-g} & \text{-a} & \text{-e} \\ \text{e} & \text{-a} & \text{g} & \text{c} \\ \text{g} & \text{-e} & \text{c} & \text{-a} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_3 \\ Y_5 \\ Y_7 \end{bmatrix} = [Ce] \begin{bmatrix} Y_0 \\ Y_2 \\ Y_4 \\ Y_6 \end{bmatrix} + [Co] \begin{bmatrix} Y_1 \\ Y_3 \\ Y_5 \\ Y_7 \end{bmatrix} = {}^1\!/_2 \begin{bmatrix} Ve0 + Vo0 \\ Ve1 + Vo1 \\ Ve2 + Vo2 \\ Ve3 + Vo2 \end{bmatrix}$$

$$\begin{bmatrix} B_7 \\ B_6 \\ B_5 \\ B_4 \end{bmatrix} = \begin{bmatrix} \text{d} & \text{b} & \text{d} & \text{f} \\ \text{d} & \text{f} & \text{-d} & \text{-b} \\ \text{d} & \text{-f} & \text{-d} & \text{b} \\ \text{d} & \text{-b} & \text{d} & \text{-f} \end{bmatrix} \begin{bmatrix} Y_0 \\ Y_2 \\ Y_4 \\ Y_6 \end{bmatrix} - \begin{bmatrix} \text{a} & \text{c} & \text{e} & \text{g} \\ \text{c} & \text{-g} & \text{-a} & \text{-e} \\ \text{e} & \text{-a} & \text{g} & \text{c} \\ \text{g} & \text{-e} & \text{c} & \text{-a} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_3 \\ Y_5 \\ Y_7 \end{bmatrix} = [Ce] \begin{bmatrix} Y_0 \\ Y_2 \\ Y_4 \\ Y_6 \end{bmatrix} - [Co] \begin{bmatrix} Y_1 \\ Y_3 \\ Y_5 \\ Y_7 \end{bmatrix} = {}^1\!/_2 \begin{bmatrix} Ve0 - Vo0 \\ Ve1 - Vo1 \\ Ve2 - Vo2 \\ Ve3 - Vo2 \end{bmatrix}$$

# 4.4 IDCT Fast Algorithm - 1

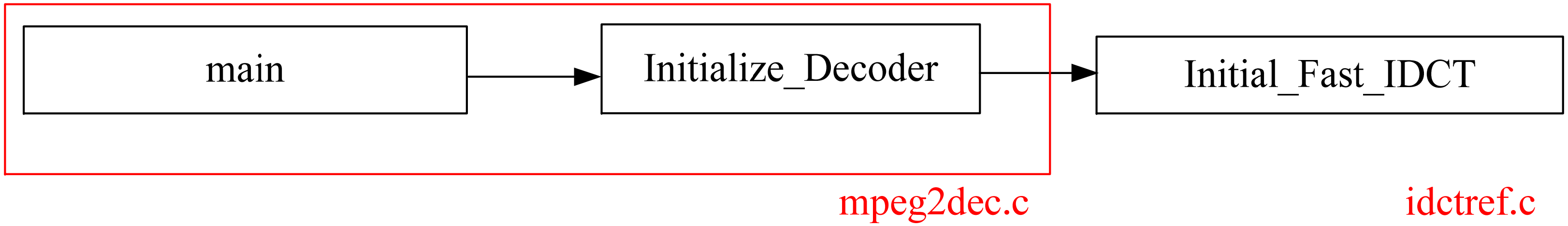
- The Coefficient of Cosine

| Cos value | Coefficient | x1024 | Label |
|---|---|---|---|
| $\sqrt{1/2}\ \cos(\pi/16)$ | 0.490393 | 1004 | a |
| $\sqrt{1/2}\ \cos(2\pi/16)$ | 0.461940 | 946 | b |
| $\sqrt{1/2}\ \cos(3\pi/16)$ | 0.415735 | 851 | c |
| $\sqrt{1/2}\ \cos(4\pi/16)$ | 0.353553 | 724 | d |
| $\sqrt{1/2}\ \cos(5\pi/16)$ | 0.277785 | 569 | e |
| $\sqrt{1/2}\ \cos(6\pi/16)$ | 0.191342 | 392 | f |
| $\sqrt{1/2}\ \cos(7\pi/16)$ | 0.097545 | 200 | g |

# 4.4 IDCT Fast Algorithm - 1

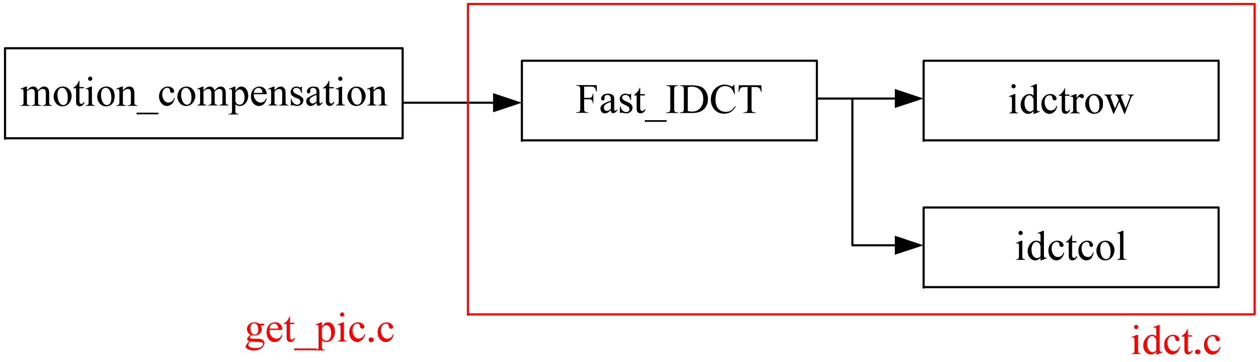
- IDCT Fast Algorithm
  - IDCT Coefficient Table

Initial IDCT table



  - Call Fast IDCT Function

IDCT fuction



# 4.4 IDCT Fast Algorithm - 1

## Modify the Fast IDCT(1/8)

- Source Code Path: mpeg2/src/mpeg2dec/idct.c

```
#define  a 1004    //Define the Coefficient of Cosine
#define  b 946
#define  c 851
#define  d 724
#define  e 569
#define  f 392
#define  g 200
```

# 4.4 IDCT Fast Algorithm - 1

## Modify the Fast IDCT(2/8)

### Odd 4x4 Matrix

- a0 = d * x0;
- a1 = b * x2;
- a2 = f * x2;
- a3 = d * x4;
- a4 = f * x6;
- a5 = b * x6;

- o0=a0+a1+a3+a4;
- o1=a0+a2-a3-a5;
- o2=a0-a2-a3+a5;
- o3=a0-a1+a3-a4;

$$\begin{bmatrix} d & b & d & f \\ d & f & -d & -b \\ d & -f & -d & b \\ d & -b & d & -f \end{bmatrix} \begin{bmatrix} Y0 \\ Y2 \\ Y4 \\ Y6 \end{bmatrix}$$

# 4.4 IDCT Fast Algorithm - 1

## Modify the Fast IDCT(3/8)

- **Even 4x4 Matrix**

```
e0=(a*x1)+(c*x3)+(e*x5)+(g*x7);
e1=(c*x1)-(g*x3)-(a*x5)-(e*x7);
e2=(e*x1)-(a*x3)+(g*x5)+(c*x7);
e3=(g*x1)-(e*x3)+(c*x5)-(a*x7);
```

$$\begin{bmatrix} a & c & e & g \\ c & -g & -a & -e \\ e & -a & g & c \\ g & -e & c & -a \end{bmatrix} \begin{bmatrix} Y1 \\ Y3 \\ Y5 \\ Y7 \end{bmatrix}$$

# 4.4 IDCT Fast Algorithm - 1

## Modify the Fast IDCT(4/8)

- Add / Sub

```
blk[0] = iclp[(o0+e0)>>14];
blk[1] = iclp[(o1+e1)>>14];
blk[2] = iclp[(o2+e2)>>14];
blk[3] = iclp[(o3+e3)>>14];
blk[4] = iclp[(o3-e3)>>14];
blk[5] = iclp[(o2-e2)>>14];
blk[6] = iclp[(o1-e1)>>14];
blk[7] = iclp[(o0-e0)>>14];
```

$$\begin{bmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} d & b & d & f \\ d & f & -d & -b \\ d & -f & -d & b \\ d & -b & d & -f \end{bmatrix} \begin{bmatrix} Y_0 \\ Y_2 \\ Y_4 \\ Y_6 \end{bmatrix} + \begin{bmatrix} a & c & e & g \\ c & -g & -a & -e \\ e & -a & g & c \\ g & -e & c & -a \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_3 \\ Y_5 \\ Y_7 \end{bmatrix}$$

$$\begin{bmatrix} B_7 \\ B_6 \\ B_5 \\ B_4 \end{bmatrix} = \begin{bmatrix} d & b & d & f \\ d & f & -d & -b \\ d & -f & -d & b \\ d & -b & d & -f \end{bmatrix} \begin{bmatrix} Y_0 \\ Y_2 \\ Y_4 \\ Y_6 \end{bmatrix} - \begin{bmatrix} a & c & e & g \\ c & -g & -a & -e \\ e & -a & g & c \\ g & -e & c & -a \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_3 \\ Y_5 \\ Y_7 \end{bmatrix}$$

# 4.4 IDCT Fast Algorithm - 1

## ■ Modify the Fast IDCT(5/8)

- Initialize the Clip Table

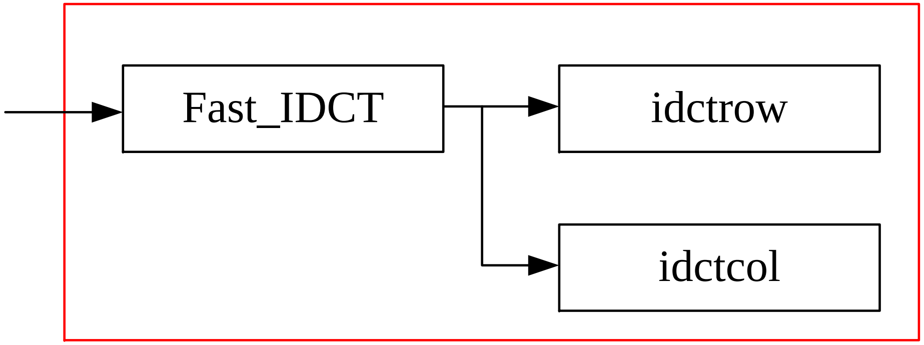
```
void Initialize_Fast_IDCT()
{
  int i;
  iclp = iclip+512;
    for (i= -512; i<512; i++)
         iclp[i] = (i<-256) ? -256 : ((i>255) ? 255 : i);
}

//Set the Clip Table
```

# 4.4 IDCT Fast Algorithm - 1

## Modify the Fast IDCT(6/8)

### Fast IDCT Algorithm

```
void Fast_IDCT(block)
short *block;
{
  int i;
     for (i=0; i<8; i++)
         idctrow(block+8*i);
     for (i=0; i<8; i++)
        idctcol(block+i);
}
```



idct.c

# 4.4 IDCT Fast Algorithm - 1

## ■ Modify the Fast IDCT(7/8)

- **File Name : idct.c**

```
 ---------------------------------------------------------------------------
Path: mpeg2/src/mpeg2dec/idct.c
static void idctrow(blk)
short *blk;
{
 int x0, x1, x2, x3, x4, x5, x6, x7, x8;
 int a0,a1,a2,a3,a4,a5;
 int o0,o1,o2,o3,int e0,e1,e2,e3;
 x0=blk[0];  x1=blk[1];  x2=blk[2];  x3=blk[3];
 x4=blk[4];  x5=blk[5];  x6=blk[6];  x7=blk[7];
 a0=d*x0;
 a1=b*x2;
 a2=f*x2;
 a3=d*x4;
 a4=f*x6;
 a5=b*x6;
 o0=a0+a1+a3+a4;
 o1=a0+a2-a3-a5;
 o2=a0-a2-a3+a5;
 o3=a0-a1+a3-a4;
 ---------------------------------------------------------------------------
```

# 4.4 IDCT Fast Algorithm - 1

## ■ Modify the Fast IDCT(7/8)

- File Name : idct.c

```
 ---------------------------------------------------------------------------
e0=(a*x1)+(c*x3)+(e*x5)+(g*x7);
e1=(c*x1)-(g*x3)-(a*x5)-(e*x7);
e2=(e*x1)-(a*x3)+(g*x5)+(c*x7);
e3=(g*x1)-(e*x3)+(c*x5)-(a*x7);

blk[0] = (o0+e0)>>8;
blk[1] = (o1+e1)>>8;
blk[2] = (o2+e2)>>8;
blk[3] = (o3+e3)>>8;
blk[4] = (o3-e3)>>8;
blk[5] = (o2-e2)>>8;
blk[6] = (o1-e1)>>8;
blk[7] = (o0-e0)>>8;
}
 ---------------------------------------------------------------------------
```

1D-IDCT

# 4.4 IDCT Fast Algorithm - 1

## ■ Modify the Fast IDCT(7/8)

- ◆ **File Name : idct.c**

2D-IDCT

```
 ---------------------------------------------------------------------------
static void idctrow(blk)
static void idctcol(blk)
short *blk;
{
  int x0, x1, x2, x3, x4, x5, x6, x7, x8;
  int a0,a1,a2,a3,a4,a5;
  int o0,o1,o2,o3;
  int e0,e1,e2,e3;
  x0=blk[8*0];  x1=blk[8*1];
  x2=blk[8*2];  x3=blk[8*3];
  x4=blk[8*4]; x5=blk[8*5];
  x6=blk[8*6]; x7=blk[8*7];

  a0=d*x0;
  a1=b*x2;
  a2=f*x2;
  a3=d*x4;
  a4=f*x6;
  a5=b*x6;
---------------------------------------------------------------------------
```

# 4.4 IDCT Fast Algorithm - 1

## ■ Modify the Fast IDCT(7/8)

- File Name : idct.c

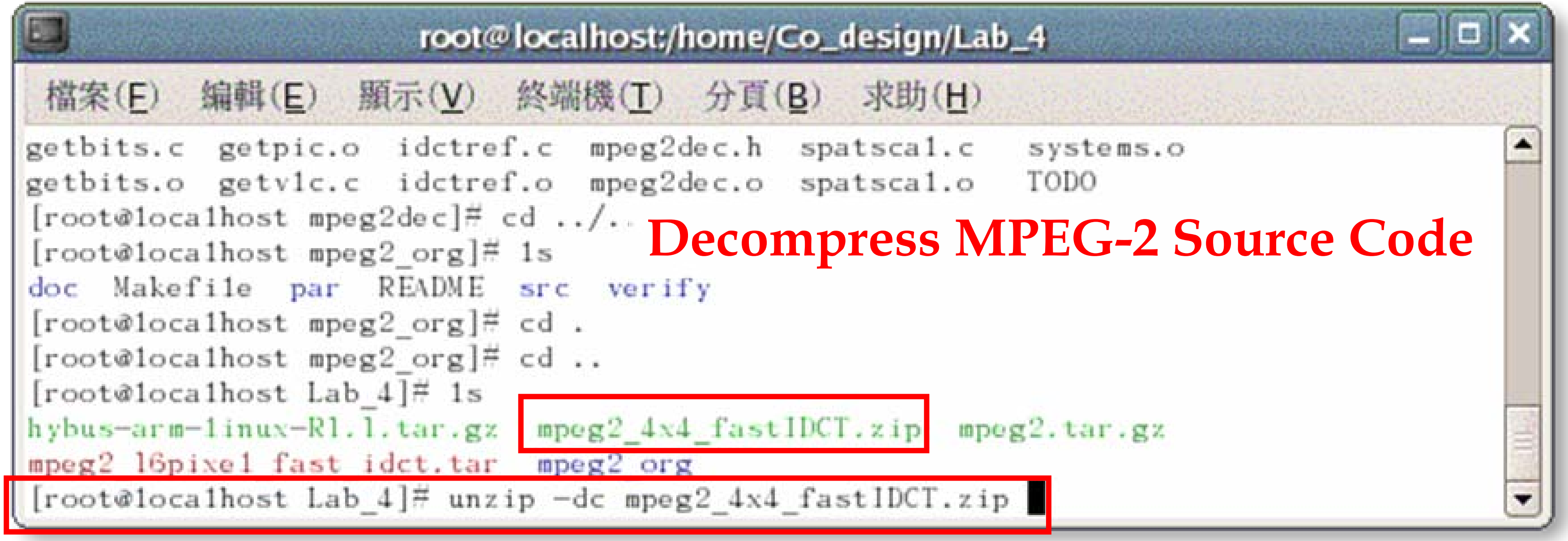
2D-IDCT

```
--------------------------------------------------------------------------
 e0=(a*x1)+(c*x3)+(e*x5)+(g*x7);
 e1=(c*x1)-(g*x3)-(a*x5)-(e*x7);
 e2=(e*x1)-(a*x3)+(g*x5)+(c*x7);
 e3=(g*x1)-(e*x3)+(c*x5)-(a*x7);

  blk[8*0] = iclp[(o0+e0)>>14];
  blk[8*1] = iclp[(o1+e1)>>14];
  blk[8*2] = iclp[(o2+e2)>>14];
  blk[8*3] = iclp[(o3+e3)>>14];
  blk[8*4] = iclp[(o3-e3)>>14];
  blk[8*5] = iclp[(o2-e2)>>14];
  blk[8*6] = iclp[(o1-e1)>>14];
  blk[8*7] = iclp[(o0-e0)>>14];
}

--------------------------------------------------------------------------
```

# 4.4 IDCT Fast Algorithm - 1

## Decompress MPEG-2 Source Code

- IDCT Fast Algorithm -1
  - MPEG-2 Source Code after Modify
    - mpeg2_4x4_fastIDCT.zip
  - #>cd /home/Co_design/Lab_4/
  - LAB4>unzip –dc mpeg2_4x4_fastIDCT.zip



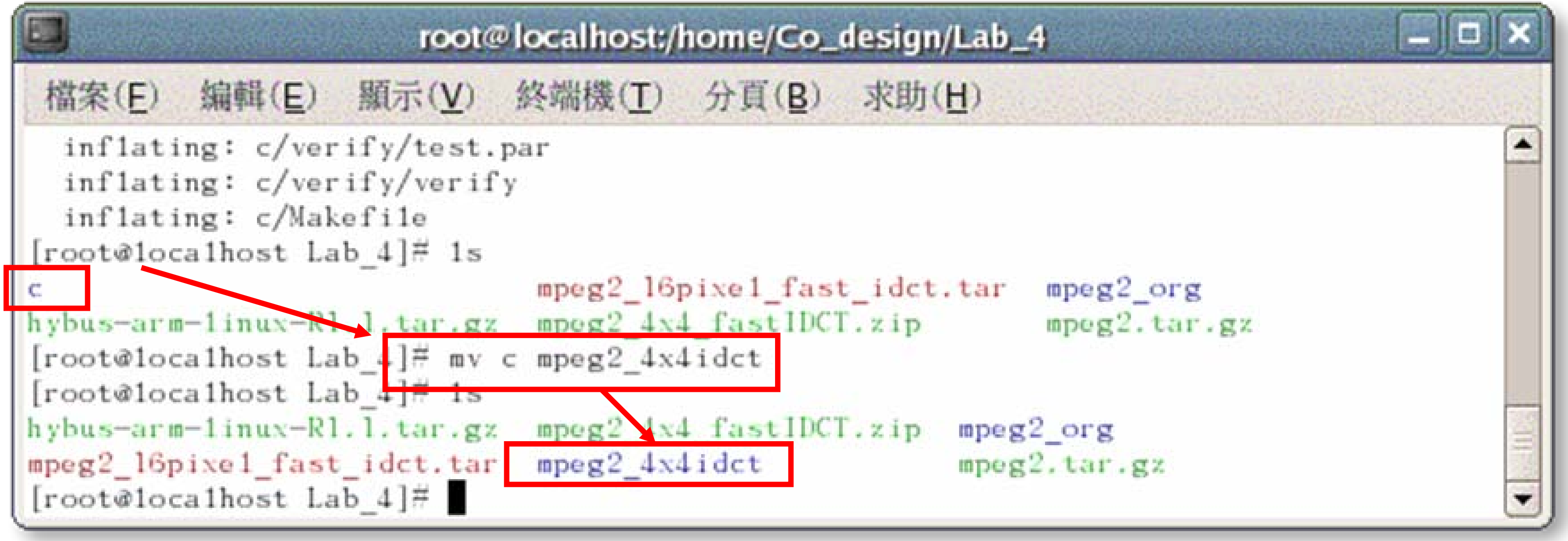
```
getbits.c   getpic.o   idctref.c   mpeg2dec.h   spatscal.c   systems.o
getbits.o   getvlc.c   idctref.o   mpeg2dec.o   spatscal.o   TODO
[root@localhost mpeg2dec]# cd ../..
[root@localhost mpeg2_org]# ls
doc  Makefile  par  README  src  verify
[root@localhost mpeg2_org]# cd .
[root@localhost mpeg2_org]# cd ..
[root@localhost Lab_4]# ls
hybus-arm-linux-R1.1.tar.gz  mpeg2_4x4_fastIDCT.zip  mpeg2.tar.gz
mpeg2_16pixel_fast_idct.tar  mpeg2_org
[root@localhost Lab_4]# unzip -dc mpeg2_4x4_fastIDCT.zip
```

Decompress MPEG-2 Source Code

# 4.4 IDCT Fast Algorithm - 1

## ■ Decompress MPEG-2 Source Code

- ◆ Get “c” Folder
  - ■ Lab_4>mv c mpeg2_4x4idct
  - ■ Change the Name of the Folder “c” into “mpeg2_4x4idct”

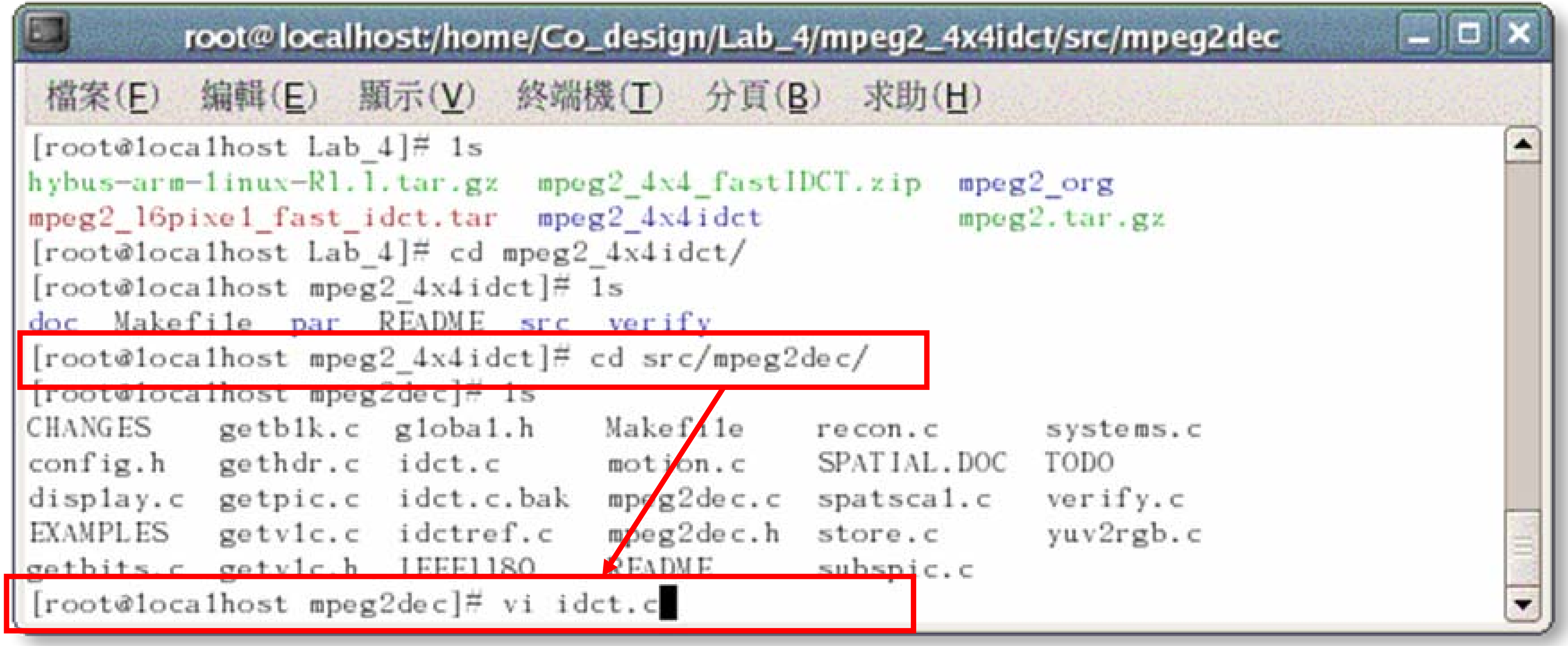

root@localhost:/home/Co_design/Lab_4
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
  inflating: c/verify/test.par
  inflating: c/verify/verify
  inflating: c/Makefile
[root@localhost Lab_4]# ls
c                            mpeg2_16pixel_fast_idct.tar  mpeg2_org
hybus-arm-linux-R1.1.tar.gz  mpeg2_4x4_fastIDCT.zip       mpeg2.tar.gz
[root@localhost Lab_4]# mv c mpeg2_4x4idct
[root@localhost Lab_4]# ls
hybus-arm-linux-R1.1.tar.gz  mpeg2_4x4_fastIDCT.zip  mpeg2_org
mpeg2_16pixel_fast_idct.tar  mpeg2_4x4idct           mpeg2.tar.gz
[root@localhost Lab_4]# █
```

Change the Name of the Folder

# 4.4 IDCT Fast Algorithm - 1

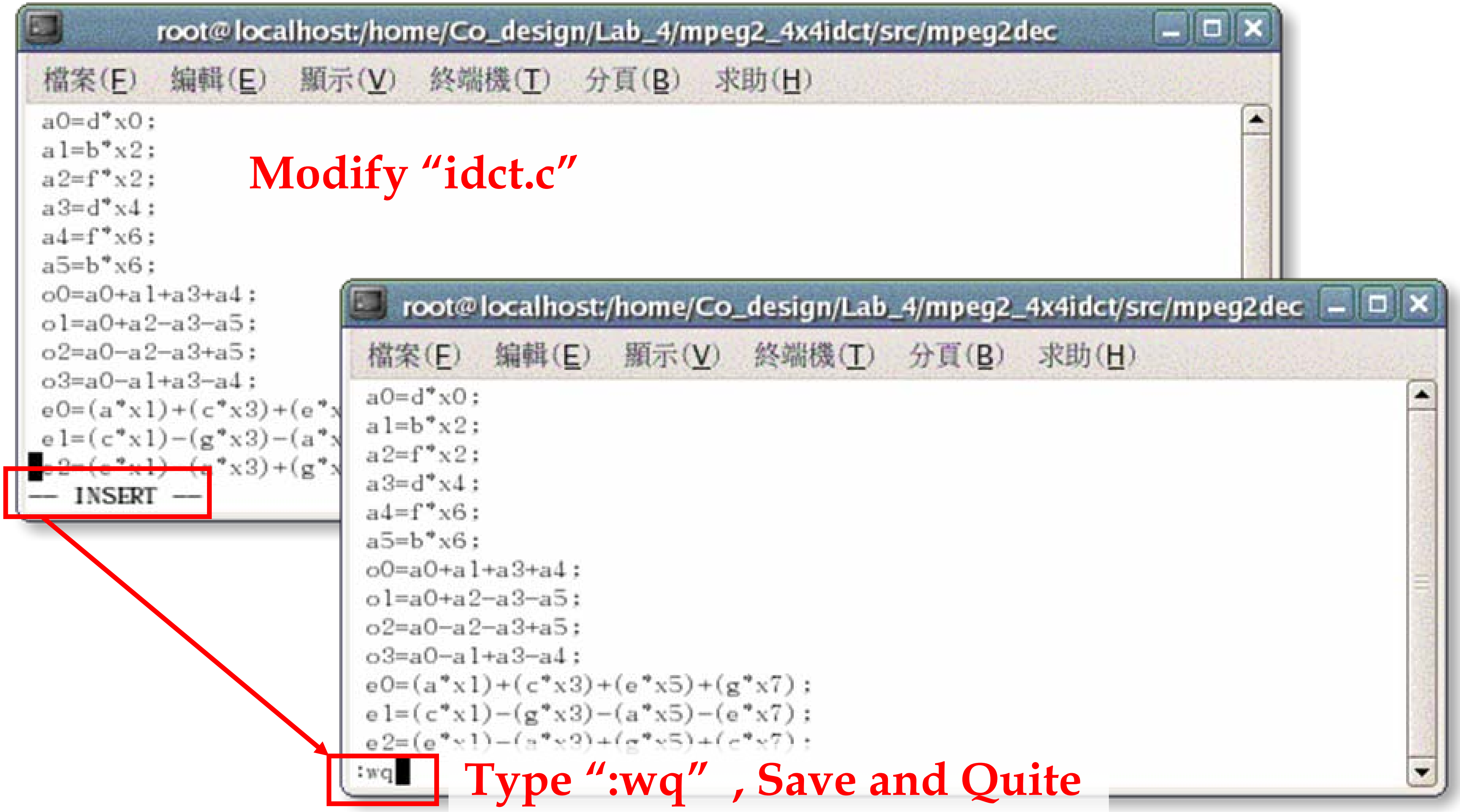
- Modify Fast IDCT Algorithm in Software
  - Modify the File : /home/Co_design/Lab_4/mpeg2_4x4idct/src/mpeg2dec/idct.c
  - #Lab_4> cd mpeg2_4x4idct/src/mpeg2dec/
  - #mpeg2dec>vi idct.c


root@localhost:/home/Co_design/Lab_4/mpeg2_4x4idct/src/mpeg2dec
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
[root@localhost Lab_4]# ls
hybus-arm-linux-R1.1.tar.gz  mpeg2_4x4_fastIDCT.zip  mpeg2_org
mpeg2_16pixel_fast_idct.tar  mpeg2_4x4idct           mpeg2.tar.gz
[root@localhost Lab_4]# cd mpeg2_4x4idct/
[root@localhost mpeg2_4x4idct]# ls
doc  Makefile  par  README  src  verify
[root@localhost mpeg2_4x4idct]# cd src/mpeg2dec/
[root@localhost mpeg2dec]# ls
CHANGES    getblk.c  global.h    Makefile    recon.c     systems.c
config.h   gethdr.c  idct.c      motion.c    SPATIAL.DOC TODO
display.c  getpic.c  idct.c.bak  mpeg2dec.c  spatscal.c  verify.c
EXAMPLES   getvlc.c  idctref.c   mpeg2dec.h  store.c     yuv2rgb.c
getbits.c  getvlc.h  IEEE1180    README      subspic.c
[root@localhost mpeg2dec]# vi idct.c
```

# 4.4 IDCT Fast Algorithm - 1

- Modify the MPEG-2 Source Code
  - Modify the Function : idctrow(), idctcol(), Fast_IDCT()


root@localhost:/home/Co_design/Lab_4/mpeg2_4x4idct/src/mpeg2dec
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
a0=d*x0;
a1=b*x2;
a2=f*x2;
a3=d*x4;
a4=f*x6;
a5=b*x6;
o0=a0+a1+a3+a4;
o1=a0+a2-a3-a5;
o2=a0-a2-a3+a5;
o3=a0-a1+a3-a4;
e0=(a*x1)+(c*x3)+(e*x
e1=(c*x1)-(g*x3)-(a*x
e2=(e*x1)-(a*x3)+(g*x
-- INSERT --
```

Modify “idct.c”

root@localhost:/home/Co_design/Lab_4/mpeg2_4x4idct/src/mpeg2dec
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
a0=d*x0;
a1=b*x2;
a2=f*x2;
a3=d*x4;
a4=f*x6;
a5=b*x6;
o0=a0+a1+a3+a4;
o1=a0+a2-a3-a5;
o2=a0-a2-a3+a5;
o3=a0-a1+a3-a4;
e0=(a*x1)+(c*x3)+(e*x5)+(g*x7);
e1=(c*x1)-(g*x3)-(a*x5)-(e*x7);
e2=(e*x1)-(a*x3)+(g*x5)+(c*x7);
:wq
```

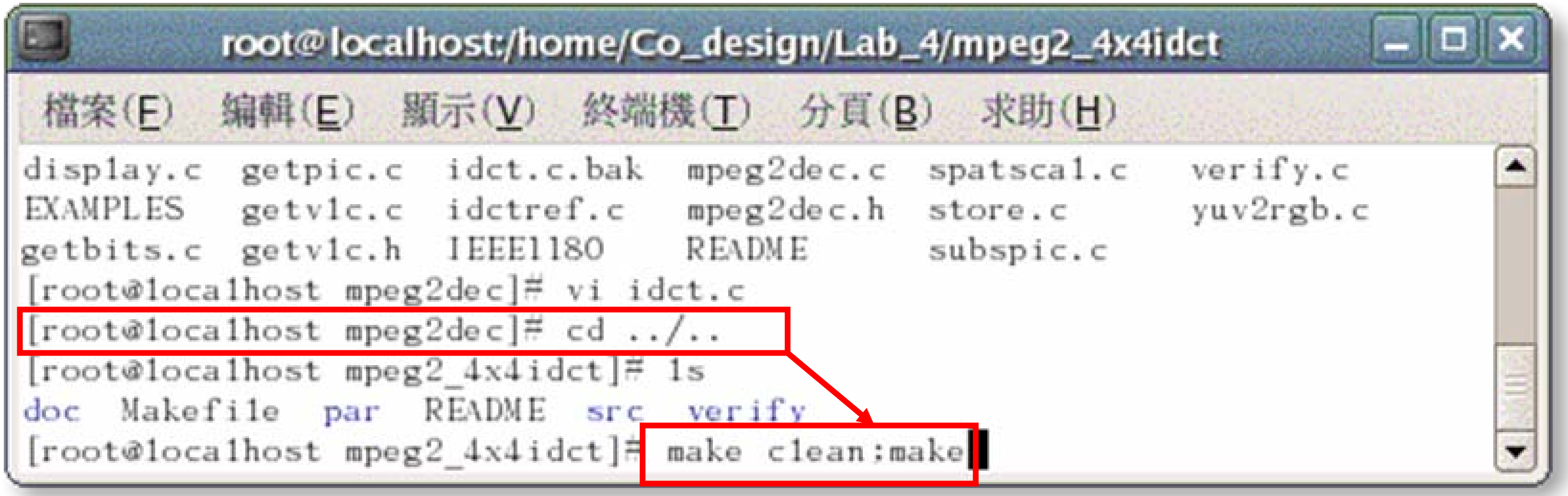
Type “:wq” , Save and Quite

# 4.4 IDCT Fast Algorithm - 1

- IDCT Fast Algorithm -1
  - Path: /home/Co_design/Lab_4/mpeg2_4x4idct/
  - #>cd /home/Co_design/Lab_4/mpeg2_4x4idct/
- Compile the MPEG-2 Source Code
  - #mpeg2_4x4idct>make clean;make

    make clean : Remove the Object Files

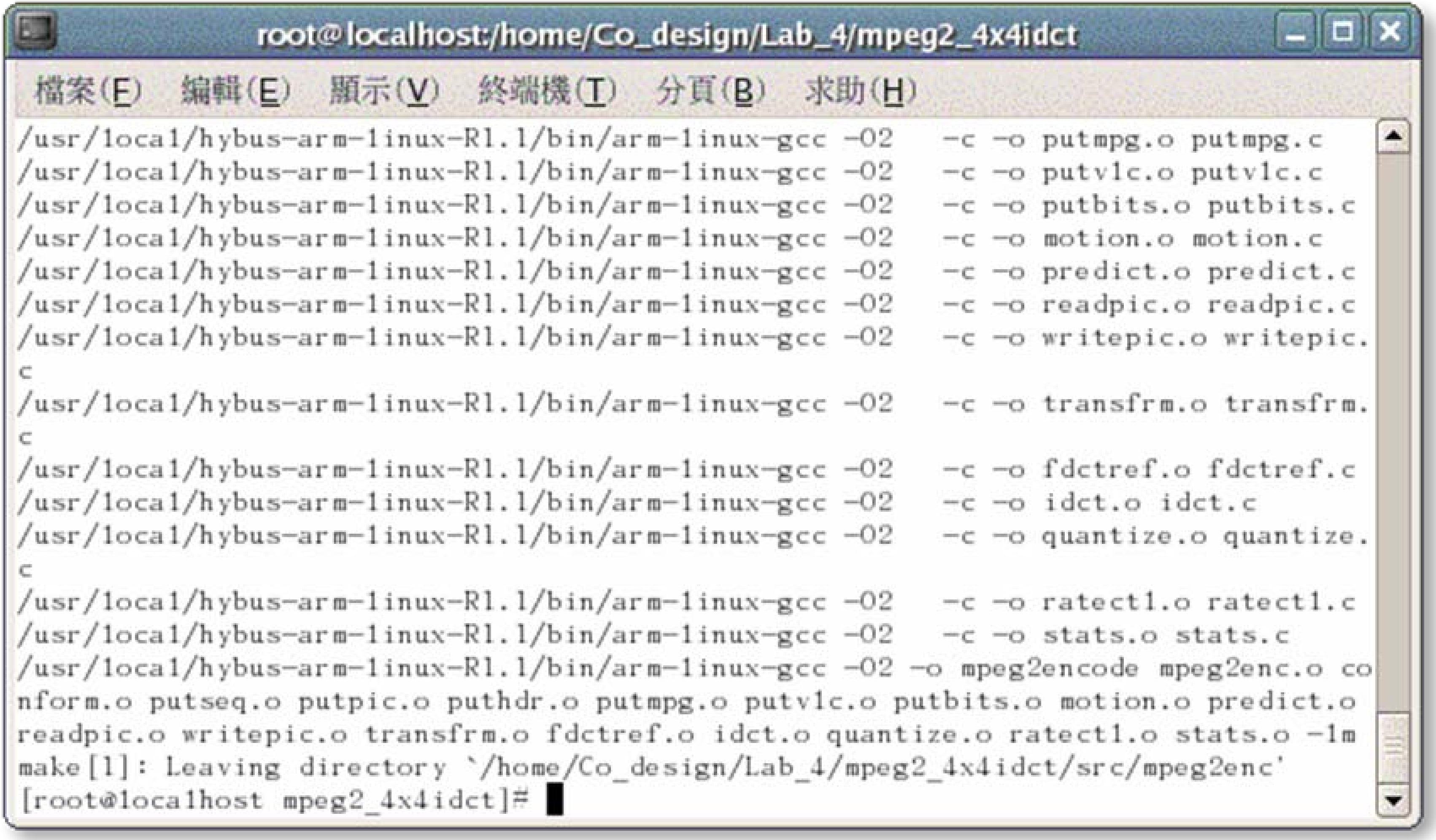
    make : Compile the MPEG-2 Source Code



```
display.c   getpic.c   idct.c.bak   mpeg2dec.c   spatscal.c   verify.c
EXAMPLES    getvlc.c   idctref.c    mpeg2dec.h   store.c      yuv2rgb.c
getbits.c   getvlc.h   IEEE1180     README       subspic.c
[root@localhost mpeg2dec]# vi idct.c
[root@localhost mpeg2dec]# cd ../..
[root@localhost mpeg2_4x4idct]# ls
doc  Makefile  par  README  src  verify
[root@localhost mpeg2_4x4idct]# make clean:make
```

# 4.4 IDCT Fast Algorithm - 1

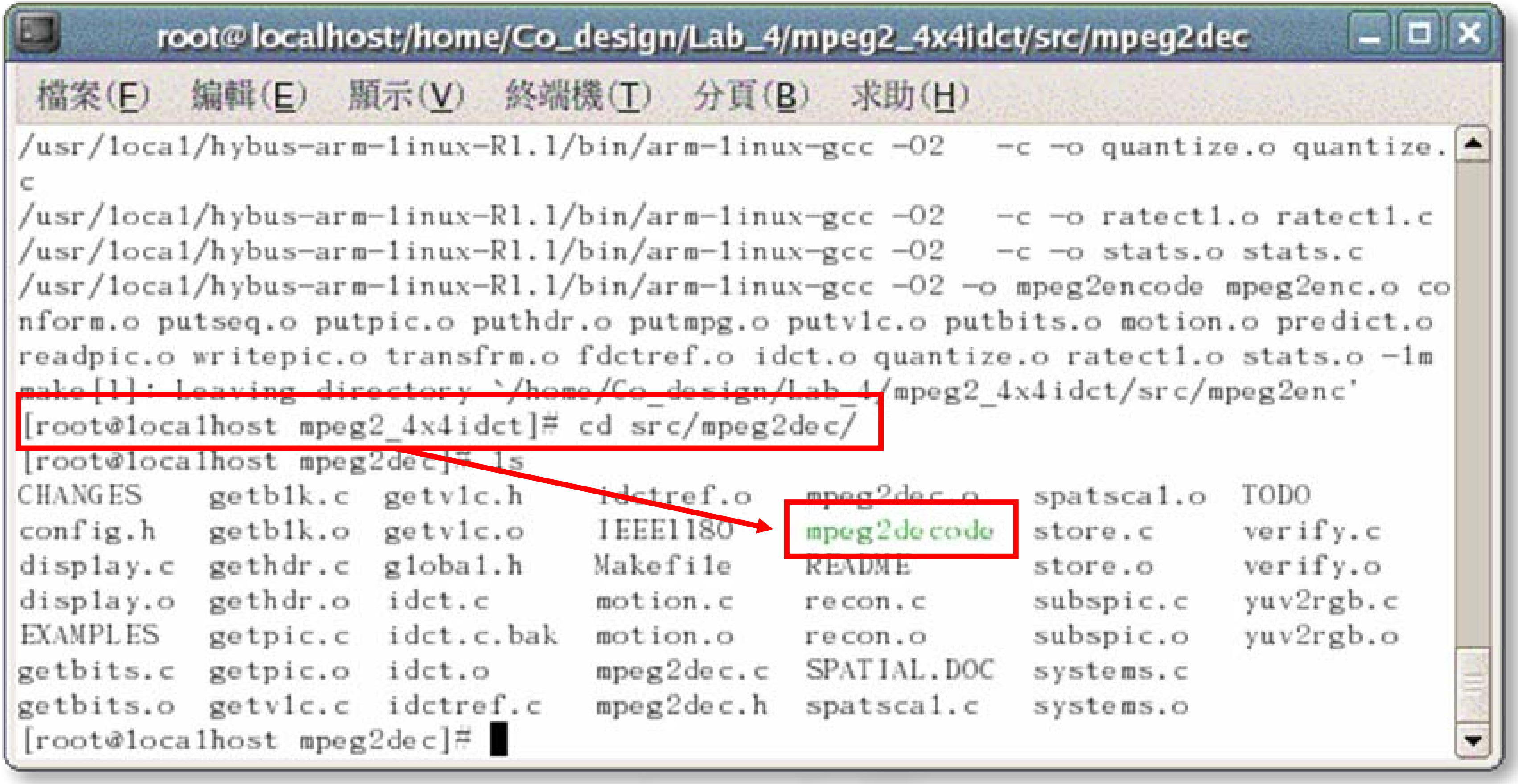
- Compile Successfully


root@localhost:/home/Co_design/Lab_4/mpeg2_4x4idct
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o putmpg.o putmpg.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o putvlc.o putvlc.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o putbits.o putbits.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o motion.o motion.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o predict.o predict.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o readpic.o readpic.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o writepic.o writepic.
c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o transfrm.o transfrm.
c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o fdctref.o fdctref.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o idct.o idct.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o quantize.o quantize.
c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o ratectl.o ratectl.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o stats.o stats.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2 -o mpeg2encode mpeg2enc.o co
nform.o putseq.o putpic.o puthdr.o putmpg.o putvlc.o putbits.o motion.o predict.o
readpic.o writepic.o transfrm.o fdctref.o idct.o quantize.o ratectl.o stats.o -lm
make[1]: Leaving directory `/home/Co_design/Lab_4/mpeg2_4x4idct/src/mpeg2enc'
[root@localhost mpeg2_4x4idct]# █
```

# 4.4 IDCT Fast Algorithm - 1

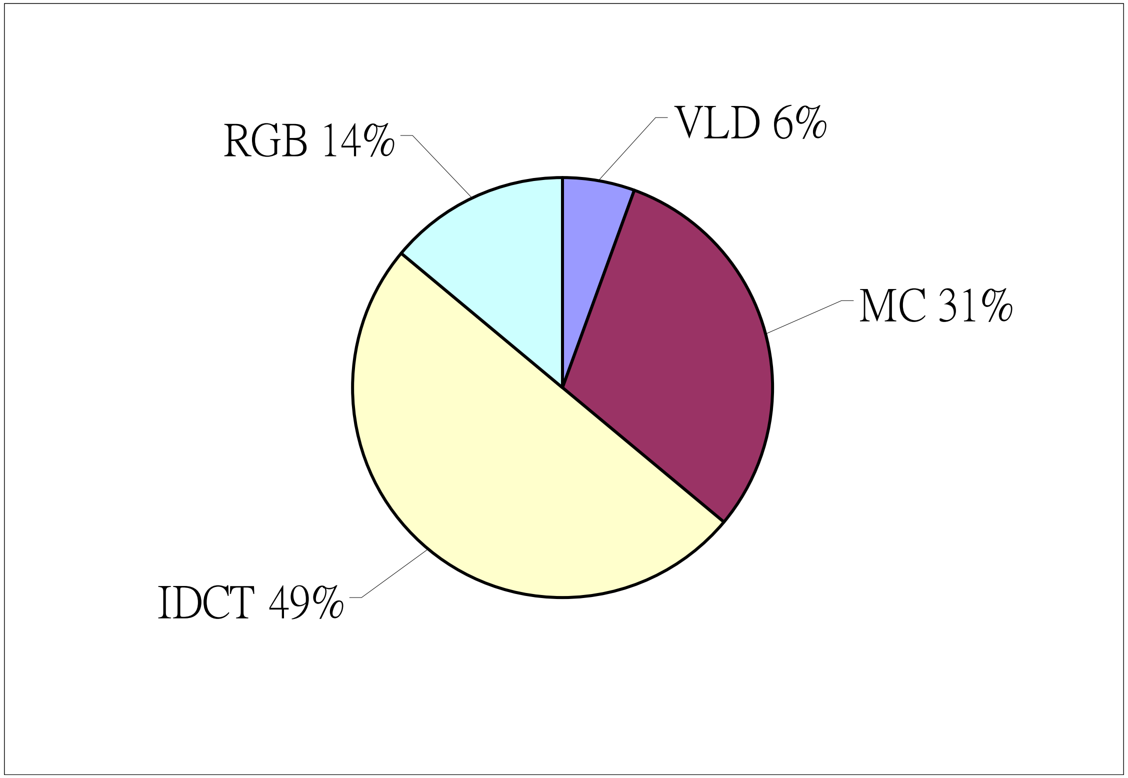
- Execute the File for ARM CPU
  - "mpeg2decode"
  - Path : /home/Co_design/Lab_4/mpeg2_4x4idct /src/mpeg2dec/


root@localhost:/home/Co_design/Lab_4/mpeg2_4x4idct/src/mpeg2dec
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o quantize.o quantize.
c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o ratectl.o ratectl.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o stats.o stats.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2 -o mpeg2encode mpeg2enc.o co
nform.o putseq.o putpic.o puthdr.o putmpg.o putvlc.o putbits.o motion.o predict.o
readpic.o writepic.o transfrm.o fdctref.o idct.o quantize.o ratectl.o stats.o -lm
make[1]: Leaving directory `/home/Co_design/Lab_4/mpeg2_4x4idct/src/mpeg2enc'
[root@localhost mpeg2_4x4idct]# cd src/mpeg2dec/
[root@localhost mpeg2dec]# ls
CHANGES     getblk.c   getvlc.h    idctref.o   mpeg2dec.o   spatscal.o  TODO
config.h    getblk.o   getvlc.o    IEEE1180    mpeg2decode  store.c     verify.c
display.c   gethdr.c   global.h    Makefile    README       store.o     verify.o
display.o   gethdr.o   idct.c      motion.c    recon.c      subspic.c   yuv2rgb.c
EXAMPLES    getpic.c   idct.c.bak  motion.o    recon.o      subspic.o   yuv2rgb.o
getbits.c   getpic.o   idct.o      mpeg2dec.c  SPATIAL.DOC  systems.c
getbits.o   getvlc.c   idctref.c   mpeg2dec.h  spatscal.c   systems.o
[root@localhost mpeg2dec]# █
```

# 4.4 IDCT Fast Algorithm - 1

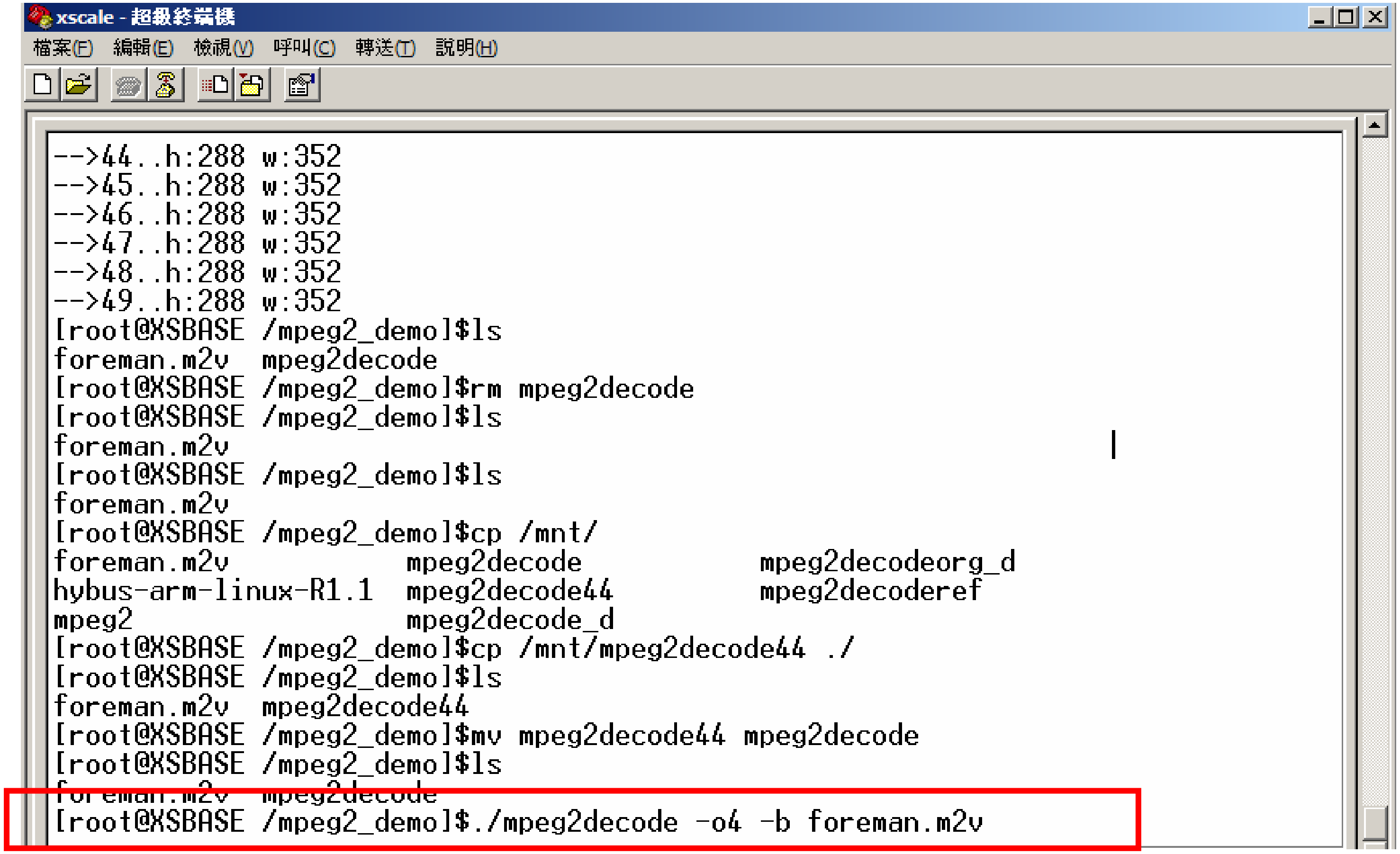
## ■ Performance Analysis



**IDCT : 77% -> 49%**

# 4.4 IDCT Fast Algorithm - 1

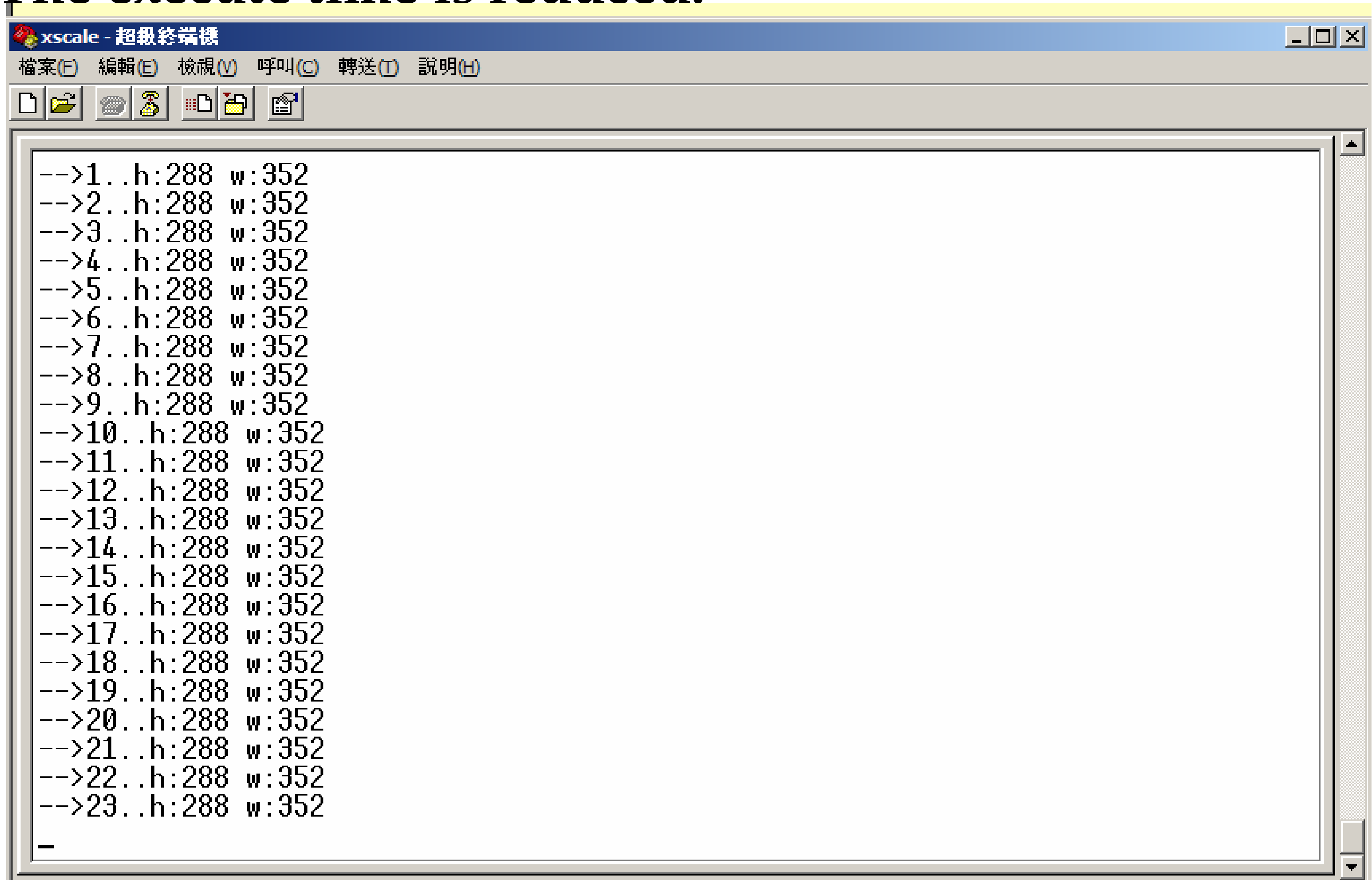
- Execute the MPEG-2 Decoder File
  - Execute MPEG-2 decoder with the fast IDCT algorithm
  - #mpeg2_demo>./mpeg2decode -o4 -b foreman.m2v



```
-->44..h:288 w:352
-->45..h:288 w:352
-->46..h:288 w:352
-->47..h:288 w:352
-->48..h:288 w:352
-->49..h:288 w:352
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v  mpeg2decode
[root@XSBASE /mpeg2_demo]$rm mpeg2decode
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v
[root@XSBASE /mpeg2_demo]$cp /mnt/
foreman.m2v           mpeg2decode           mpeg2decodeorg_d
hybus-arm-linux-R1.1  mpeg2decode44         mpeg2decoderef
mpeg2                 mpeg2decode_d
[root@XSBASE /mpeg2_demo]$cp /mnt/mpeg2decode44 ./
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v  mpeg2decode44
[root@XSBASE /mpeg2_demo]$mv mpeg2decode44 mpeg2decode
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v  mpeg2decode
[root@XSBASE /mpeg2_demo]$./mpeg2decode -o4 -b foreman.m2v
```

# 4.4 IDCT Fast Algorithm - 1

- Execute MPEG-2 Decoder
  - Display on TFT LCD
  - The execute time is reduced.



```
-->1..h:288 w:352
-->2..h:288 w:352
-->3..h:288 w:352
-->4..h:288 w:352
-->5..h:288 w:352
-->6..h:288 w:352
-->7..h:288 w:352
-->8..h:288 w:352
-->9..h:288 w:352
-->10..h:288 w:352
-->11..h:288 w:352
-->12..h:288 w:352
-->13..h:288 w:352
-->14..h:288 w:352
-->15..h:288 w:352
-->16..h:288 w:352
-->17..h:288 w:352
-->18..h:288 w:352
-->19..h:288 w:352
-->20..h:288 w:352
-->21..h:288 w:352
-->22..h:288 w:352
-->23..h:288 w:352
_
```

# 4.4 IDCT Fast Algorithm - 1

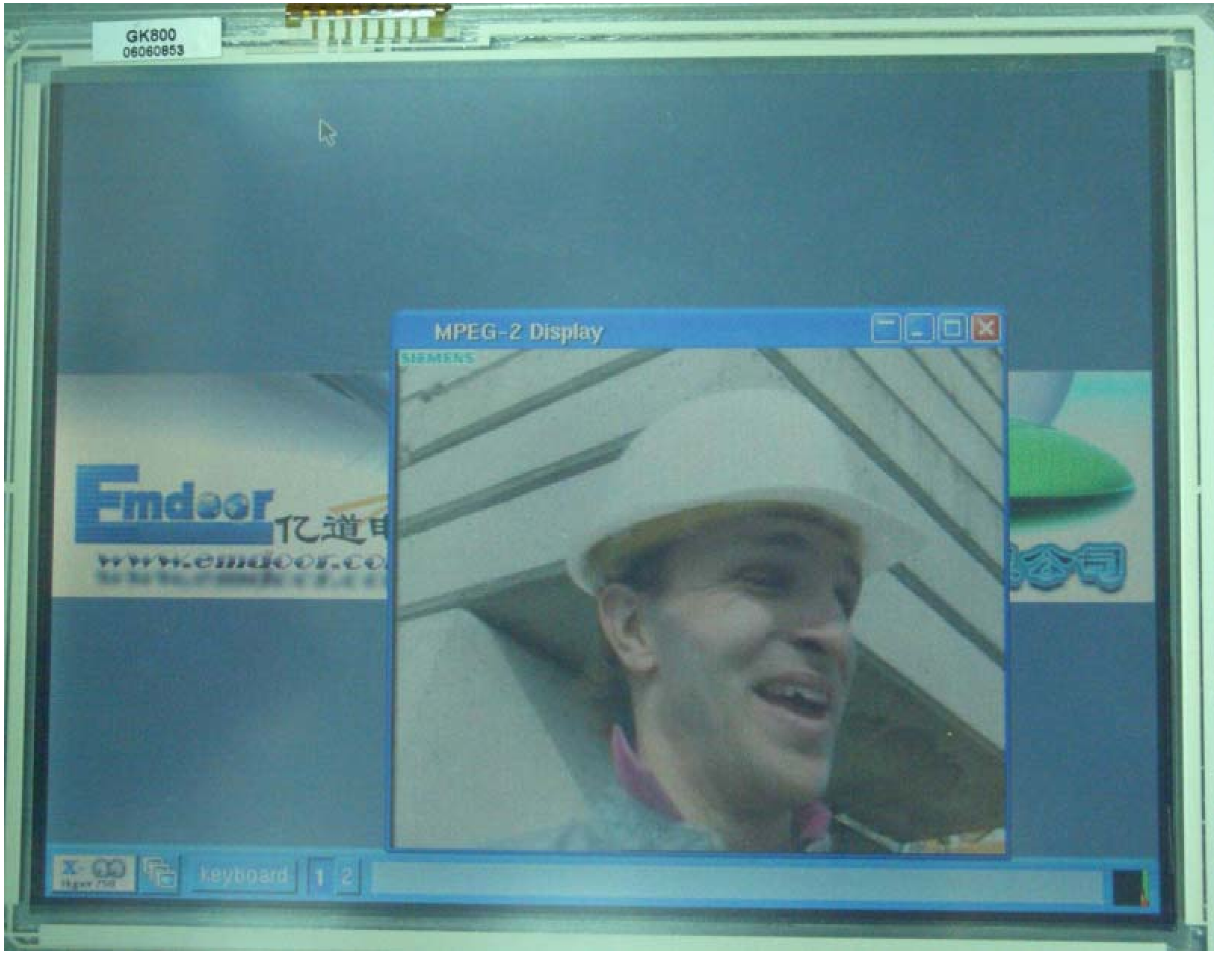
## Performance Report

- The Method to Improve the Performance
  - Fast Algorithm
    - IDCT Fast Algorithm – 1
  - C Program
    - The Type of the Data

      -Integer
- Total Execute Time
  - Execute Time = 6.52 Seconds
  - Frame Rate = 7.66 fps
    - fps : Frame Per Second

# 4.4 IDCT Fast Algorithm - 1

- Display on TFT LCD



# 4.5 IDCT Fast Algorithm - 2

4.1 Preparing

4.2 MPEG-2 Decoder Software

4.3 IDCT Source Code

4.4 IDCT Fast Algorithm - 1

4.5 IDCT Fast Algorithm – 2

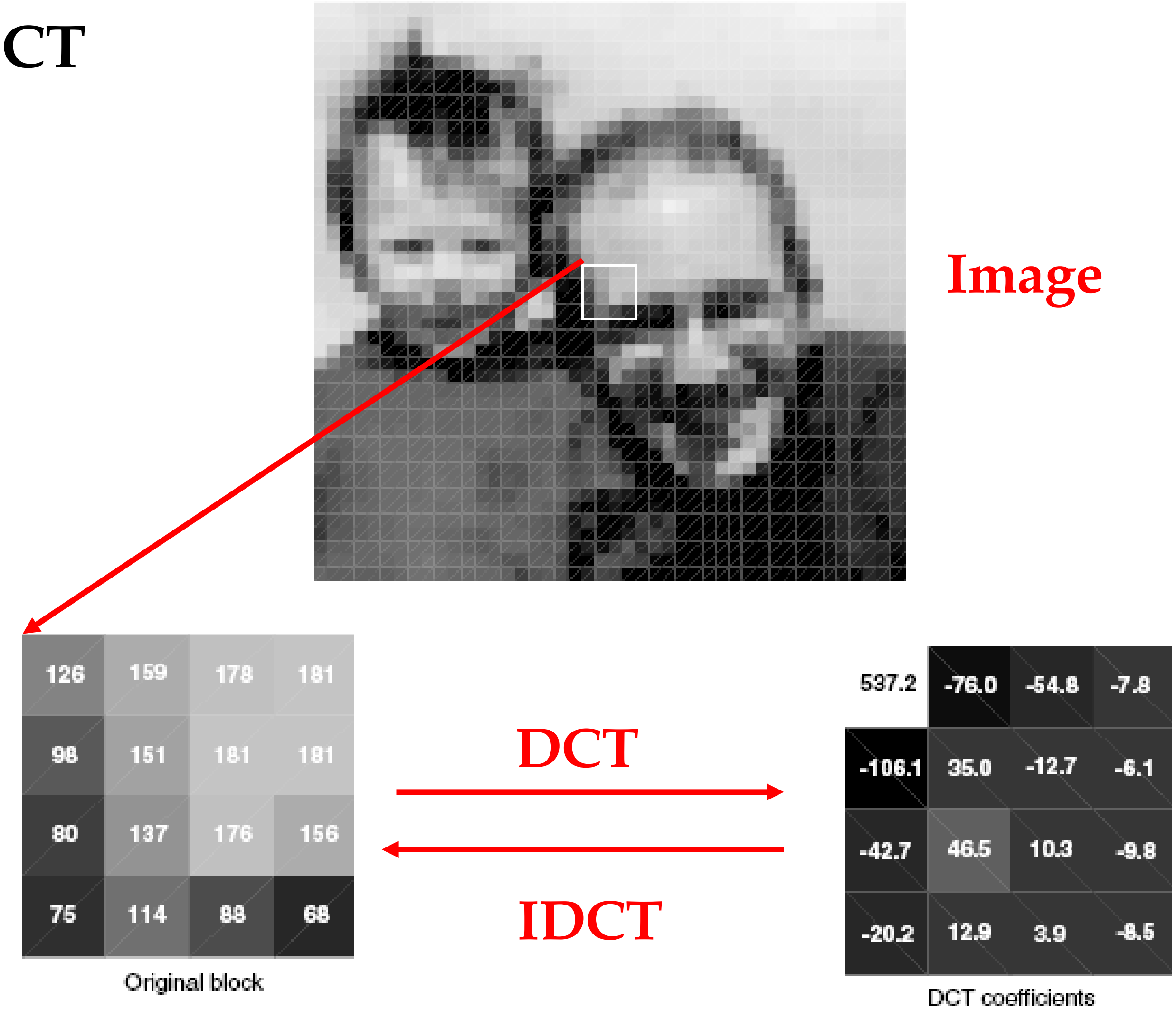
4.6 Insert Assembly Code in C Language

# 4.5 IDCT Fast Algorithm - 2

- IDCT Fast Algorithm – 2
  - Use Two 4x4 Matrix to Complete IDCT
  - Lossy Operation
  - MPEG-2 Source Code after Modify
    - mpeg2_16pixel_fast_idct.tar

# 4.5 IDCT Fast Algorithm - 2

- DCT/IDCT



Image

| | | | |
|---|---|---|---|
| 126 | 159 | 178 | 181 |
| 98 | 151 | 181 | 181 |
| 80 | 137 | 176 | 156 |
| 75 | 114 | 88 | 68 |

Original block

DCT

IDCT

| | | | |
|---|---|---|---|
| 537.2 | -76.0 | -54.8 | -7.8 |
| -106.1 | 35.0 | -12.7 | -6.1 |
| -42.7 | 46.5 | 10.3 | -9.8 |
| -20.2 | 12.9 | 3.9 | -8.5 |

DCT coefficients

# 4.5 IDCT Fast Algorithm - 2

- **General 8 x 8 IDCT**

  **$X = A^TYA$ Let $B = A^TY$ ,$X = BA$**

$$
\begin{bmatrix}
B_0 & B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 \\
B_8 & B_9 & B_{10} & B_{11} & B_{12} & B_{13} & B_{14} & B_{15} \\
B_{16} & B_{17} & B_{18} & B_{19} & B_{20} & B_{21} & B_{22} & B_{23} \\
B_{24} & B_{25} & B_{26} & B_{27} & B_{28} & B_{29} & B_{30} & B_{31} \\
B_{32} & B_{33} & B_{34} & B_{35} & B_{36} & B_{37} & B_{38} & B_{39} \\
B_{40} & B_{41} & B_{42} & B_{43} & B_{44} & B_{45} & B_{46} & B_{47} \\
B_{48} & B_{49} & B_{50} & B_{51} & B_{52} & B_{53} & B_{54} & B_{55} \\
B_{56} & B_{57} & B_{58} & B_{59} & B_{60} & B_{61} & B_{62} & B_{63}
\end{bmatrix}
=
\begin{bmatrix}
d & a & b & c & d & e & f & g \\
d & c & f & -g & -d & -a & -b & -e \\
d & e & -f & -a & -d & g & b & c \\
d & g & -b & -e & -d & c & -f & -a \\
d & -g & -b & e & d & -c & -f & a \\
d & -e & -f & a & -d & -g & b & -c \\
d & -c & f & g & -d & a & -b & f \\
d & -a & b & -c & d & -e & f & -g
\end{bmatrix}
\begin{bmatrix}
Y_0 & Y_1 & Y_2 & Y_3 & Y_4 & Y_5 & Y_6 & Y_7 \\
Y_8 & Y_9 & Y_{10} & Y_{11} & Y_{12} & Y_{13} & Y_{14} & Y_{15} \\
Y_{16} & Y_{17} & Y_{18} & Y_{19} & Y_{20} & Y_{21} & Y_{22} & Y_{23} \\
Y_{24} & Y_{25} & Y_{26} & Y_{27} & Y_{28} & Y_{29} & Y_{30} & Y_{31} \\
Y_{32} & Y_{33} & Y_{34} & Y_{35} & Y_{36} & Y_{37} & Y_{38} & Y_{39} \\
Y_{40} & Y_{41} & Y_{42} & Y_{43} & Y_{44} & Y_{45} & Y_{46} & Y_{47} \\
Y_{48} & Y_{49} & Y_{50} & Y_{51} & Y_{52} & Y_{53} & Y_{54} & Y_{55} \\
Y_{56} & Y_{57} & Y_{58} & Y_{59} & Y_{60} & Y_{61} & Y_{62} & Y_{63}
\end{bmatrix}
$$

$$B \qquad = \qquad A^T \qquad\qquad Y$$

$$
\begin{bmatrix}
X_0 & X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 \\
X_8 & X_9 & X_{10} & X_{11} & X_{12} & X_{13} & X_{14} & X_{15} \\
X_{16} & X_{17} & X_{18} & X_{19} & X_{20} & X_{21} & X_{22} & X_{23} \\
X_{24} & X_{25} & X_{26} & X_{27} & X_{28} & X_{29} & X_{30} & X_{31} \\
X_{32} & X_{33} & X_{34} & X_{35} & X_{36} & X_{37} & X_{38} & X_{39} \\
X_{40} & X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} & X_{47} \\
X_{48} & X_{49} & X_{50} & X_{51} & X_{52} & X_{53} & X_{54} & X_{55} \\
X_{56} & X_{57} & X_{58} & X_{59} & X_{60} & X_{61} & X_{62} & X_{63}
\end{bmatrix}
=
\begin{bmatrix}
B_0 & B_1 & B_2 & B_3 & B_4 & B_5 & B_6 & B_7 \\
B_8 & B_9 & B_{10} & B_{11} & B_{12} & B_{13} & B_{14} & B_{15} \\
B_{16} & B_{17} & B_{18} & B_{19} & B_{20} & B_{21} & B_{22} & B_{23} \\
B_{24} & B_{25} & B_{26} & B_{27} & B_{28} & B_{29} & B_{30} & B_{31} \\
B_{32} & B_{33} & B_{34} & B_{35} & B_{36} & B_{37} & B_{38} & B_{39} \\
B_{40} & B_{41} & B_{42} & B_{43} & B_{44} & B_{45} & B_{46} & B_{47} \\
B_{48} & B_{49} & B_{50} & B_{51} & B_{52} & B_{53} & B_{54} & B_{55} \\
B_{56} & B_{57} & B_{58} & B_{59} & B_{60} & B_{61} & B_{62} & B_{63}
\end{bmatrix}
\begin{bmatrix}
d & d & d & d & d & d & d & d \\
a & c & e & g & -g & -e & -c & -a \\
b & f & -f & -b & -b & -f & f & b \\
c & -g & -a & -e & e & a & g & -c \\
d & -d & -d & d & d & -d & -d & d \\
e & -a & g & c & -c & -g & a & -e \\
f & -b & b & -f & -f & b & -b & f \\
g & -e & c & -a & a & -c & e & -g
\end{bmatrix}
$$

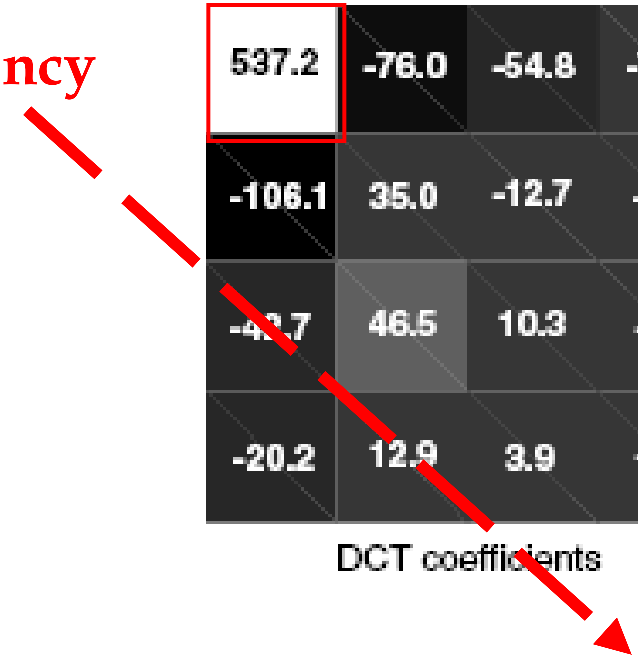
$$X \qquad = \qquad B \qquad\qquad A$$

# 4.5 IDCT Fast Algorithm - 2

- DCT Coefficients
  - DC : Coefficient (0,0)

DC

Low Frequency



| | | | |
|---|---|---|---|
| 537.2 | -76.0 | -54.8 | -7.8 |
| -106.1 | 35.0 | -12.7 | -6.1 |
| -42.7 | 46.5 | 10.3 | -9.8 |
| -20.2 | 12.9 | 3.9 | -8.5 |

DCT coefficients

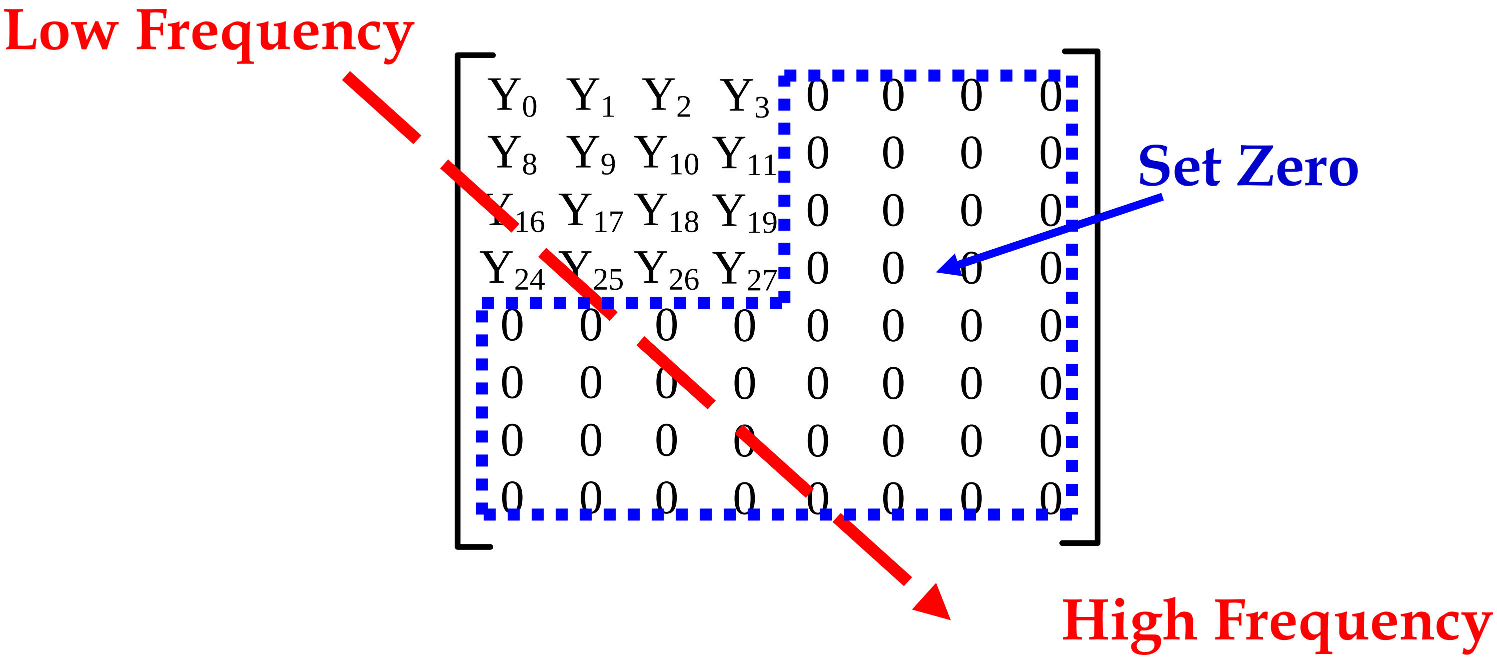
High Frequency

# 4.5 IDCT Fast Algorithm – 2

- The Eyes of Human is Sensitive with the Low Frequency of IDCT Coefficient.

# 4.5 IDCT Fast Algorithm – 2

- Preserve the Low Frequency of the IDCT Coefficient
  - Y0,Y1,Y2,Y3,Y8,Y9,Y10,Y11,Y16,Y17,Y18,Y19,Y24,Y25,Y26,Y27



Low Frequency

Set Zero

High Frequency

$$
\begin{bmatrix}
Y_0 & Y_1 & Y_2 & Y_3 & 0 & 0 & 0 & 0 \\
Y_8 & Y_9 & Y_{10} & Y_{11} & 0 & 0 & 0 & 0 \\
Y_{16} & Y_{17} & Y_{18} & Y_{19} & 0 & 0 & 0 & 0 \\
Y_{24} & Y_{25} & Y_{26} & Y_{27} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

**IDCT Coefficient**

# 4.5 IDCT Fast Algorithm - 2

- **IDCT Fast Algorithm**

$$\underset{Y}{\begin{bmatrix} Y_0 & Y_1 & Y_2 & Y_3 & 0 & 0 & 0 & 0 \\ Y_8 & Y_9 & Y_{10} & Y_{11} & 0 & 0 & 0 & 0 \\ Y_{16} & Y_{17} & Y_{18} & Y_{19} & 0 & 0 & 0 & 0 \\ Y_{24} & Y_{25} & Y_{26} & Y_{27} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}} \xrightarrow{X = A^TYA} \underset{X}{\begin{bmatrix} X_0 & X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 \\ X_8 & X_9 & X_{10} & X_{11} & X_{12} & X_{13} & X_{14} & X_{15} \\ X_{16} & X_{17} & X_{18} & X_{19} & X_{20} & X_{21} & X_{22} & X_{23} \\ X_{24} & X_{25} & X_{26} & X_{27} & X_{28} & X_{29} & X_{30} & X_{31} \\ X_{32} & X_{33} & X_{34} & X_{35} & X_{36} & X_{37} & X_{38} & X_{39} \\ X_{40} & X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} & X_{47} \\ X_{48} & X_{49} & X_{50} & X_{51} & X_{52} & X_{53} & X_{54} & X_{55} \\ X_{56} & X_{57} & X_{58} & X_{59} & X_{60} & X_{61} & X_{62} & X_{63} \end{bmatrix}}$$

# 4.5 IDCT Fast Algorithm - 2

- **Fast 16 Pixels IDCT**

**$X = A^TYA$ Let $B = A^TY$ ,$X = BA$**

$$\begin{bmatrix} B_0 & B_1 & B_2 & B_3 & 0 & 0 & 0 & 0 \\ B_8 & B_9 & B_{10} & B_{11} & 0 & 0 & 0 & 0 \\ B_{16} & B_{17} & B_{18} & B_{19} & 0 & 0 & 0 & 0 \\ B_{24} & B_{25} & B_{26} & B_{27} & 0 & 0 & 0 & 0 \\ B_{32} & B_{33} & B_{34} & B_{35} & 0 & 0 & 0 & 0 \\ B_{40} & B_{41} & B_{42} & B_{43} & 0 & 0 & 0 & 0 \\ B_{48} & B_{49} & B_{50} & B_{51} & 0 & 0 & 0 & 0 \\ B_{56} & B_{57} & B_{58} & B_{59} & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} d & a & b & c & d & e & f & g \\ d & c & f & -g & -d & -a & -b & -e \\ d & e & -f & -a & -d & g & b & c \\ d & g & -b & -e & -d & c & -f & -a \\ d & -g & -b & e & d & -c & -f & a \\ d & -e & -f & a & -d & -g & b & -c \\ d & -c & f & g & -d & a & -b & f \\ d & -a & b & -c & d & -e & f & -g \end{bmatrix} \begin{bmatrix} Y_0 & Y_1 & Y_2 & Y_3 & 0 & 0 & 0 & 0 \\ Y_8 & Y_9 & Y_{10} & Y_{11} & 0 & 0 & 0 & 0 \\ Y_{16} & Y_{17} & Y_{18} & Y_{19} & 0 & 0 & 0 & 0 \\ Y_{24} & Y_{25} & Y_{26} & Y_{27} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B \qquad = \qquad A^T \qquad\qquad Y$$

$$\begin{bmatrix} X_0 & X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 \\ X_8 & X_9 & X_{10} & X_{11} & X_{12} & X_{13} & X_{14} & X_{15} \\ X_{16} & X_{17} & X_{18} & X_{19} & X_{20} & X_{21} & X_{22} & X_{23} \\ X_{24} & X_{25} & X_{26} & X_{27} & X_{28} & X_{29} & X_{30} & X_{31} \\ X_{32} & X_{33} & X_{34} & X_{35} & X_{36} & X_{37} & X_{38} & X_{39} \\ X_{40} & X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} & X_{47} \\ X_{48} & X_{49} & X_{50} & X_{51} & X_{52} & X_{53} & X_{54} & X_{55} \\ X_{56} & X_{57} & X_{58} & X_{59} & X_{60} & X_{61} & X_{62} & X_{63} \end{bmatrix} = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & 0 & 0 & 0 & 0 \\ B_8 & B_9 & B_{10} & B_{11} & 0 & 0 & 0 & 0 \\ B_{16} & B_{17} & B_{18} & B_{19} & 0 & 0 & 0 & 0 \\ B_{24} & B_{25} & B_{26} & B_{27} & 0 & 0 & 0 & 0 \\ B_{32} & B_{33} & B_{34} & B_{35} & 0 & 0 & 0 & 0 \\ B_{40} & B_{41} & B_{42} & B_{43} & 0 & 0 & 0 & 0 \\ B_{48} & B_{49} & B_{50} & B_{51} & 0 & 0 & 0 & 0 \\ B_{56} & B_{57} & B_{58} & B_{59} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} d & d & d & d & d & d & d & d \\ a & c & e & g & -g & -e & -c & -a \\ b & f & -f & -b & -b & -f & f & b \\ c & -g & -a & -e & e & a & g & -c \\ d & -d & -d & d & d & -d & -d & d \\ e & -a & g & c & -c & -g & a & -e \\ f & -b & b & -f & -f & b & -b & f \\ g & -e & c & -a & a & -c & e & -g \end{bmatrix}$$

$$X \qquad = \qquad B \qquad\qquad A$$

# 4.5 IDCT Fast Algorithm – 2

## Fast 16 Pixels IDCT(1/8)

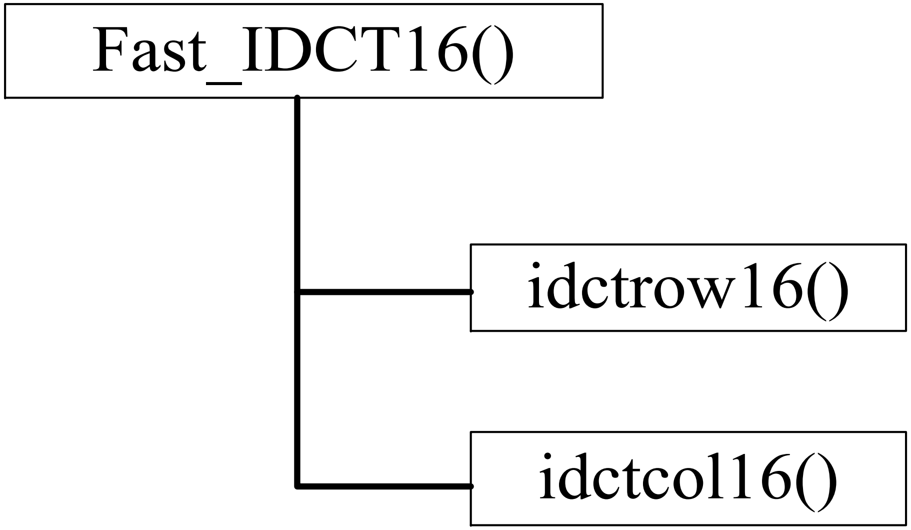
- Modify Source Code
  - Modify Files : getpic.c , idct.c
  - Modify Function
    - getpic.c :

      Add  Fast_IDCT16()
    - idct.c  :

      Fast_IDCT16()

      idctrow16()

      idctcol16();



# 4.5 IDCT Fast Algorithm – 2

## ■ Fast 16 Pixels IDCT(2/8)

◆ void idctrow16()

$$\begin{bmatrix} B_0 & B_1 & B_2 & B_3 & 0 & 0 & 0 & 0 \\ B_8 & B_9 & B_{10} & B_{11} & 0 & 0 & 0 & 0 \\ B_{16} & B_{17} & B_{18} & B_{19} & 0 & 0 & 0 & 0 \\ B_{24} & B_{25} & B_{26} & B_{27} & 0 & 0 & 0 & 0 \\ B_{32} & B_{33} & B_{34} & B_{35} & 0 & 0 & 0 & 0 \\ B_{40} & B_{41} & B_{42} & B_{43} & 0 & 0 & 0 & 0 \\ B_{48} & B_{49} & B_{50} & B_{51} & 0 & 0 & 0 & 0 \\ B_{56} & B_{57} & B_{58} & B_{59} & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} d & a & b & c & d & e & f & g \\ d & c & f & -g & -d & -a & -b & -e \\ d & e & -f & -a & -d & g & b & c \\ d & g & -b & -e & -d & c & -f & -a \\ d & -g & -b & e & d & -c & -f & a \\ d & -e & -f & a & -d & -g & b & -c \\ d & -c & f & g & -d & a & -b & f \\ d & -a & b & -c & d & -e & f & -g \end{bmatrix} \begin{bmatrix} Y_0 & Y_1 & Y_2 & Y_3 & 0 & 0 & 0 & 0 \\ Y_8 & Y_9 & Y_{10} & Y_{11} & 0 & 0 & 0 & 0 \\ Y_{16} & Y_{17} & Y_{18} & Y_{19} & 0 & 0 & 0 & 0 \\ Y_{24} & Y_{25} & Y_{26} & Y_{27} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# 4.5 IDCT Fast Algorithm - 2

## ■ Fast 16 Pixels IDCT(3/8)

- File Name :idct.c

```
------------------------------------
static void idctrow16(blk)
short *blk;
{
  int x0, x1, x2, x3, x4;
  int x5, x6, x7, x8;
  int a0,a1,a2,a3,a4,a5;
  int o0,o1,o2,o3;
  int e0,e1,e2,e3;
  x0=blk[0];
  x1=blk[1];
  x2=blk[2];
  x3=blk[3];

  a0=d*x0;
  a1=b*x2;
  a2=f*x2;
------------------------------------
```

```
------------------------------------
  o0=a0+a1;
  o1=a0+a2;
  o2=a0-a2;
  o3=a0-a1;
  e0=(a*x1)+(c*x3);
  e1=(c*x1)-(g*x3);
  e2=(e*x1)-(a*x3);
  e3=(g*x1)-(e*x3);


  blk[0] = (o0+e0)>>8;
  blk[1] = (o1+e1)>>8;
  blk[2] = (o2+e2)>>8;
  blk[3] = (o3+e3)>>8;
  blk[4] = (o3-e3)>>8;
  blk[5] = (o2-e2)>>8;
  blk[6] = (o1-e1)>>8;
  blk[7] = (o0-e0)>>8;
  }
------------------------------------
```

# 4.5 IDCT Fast Algorithm - 2

## ■ Fast 16 Pixels IDCT(4/8)

- void idctcol16()

$$\begin{bmatrix} X_0 & X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 \\ X_8 & X_9 & X_{10} & X_{11} & X_{12} & X_{13} & X_{14} & X_{15} \\ X_{16} & X_{17} & X_{18} & X_{19} & X_{20} & X_{21} & X_{22} & X_{23} \\ X_{24} & X_{25} & X_{26} & X_{27} & X_{28} & X_{29} & X_{30} & X_{31} \\ X_{32} & X_{33} & X_{34} & X_{35} & X_{36} & X_{37} & X_{38} & X_{39} \\ X_{40} & X_{41} & X_{42} & X_{43} & X_{44} & X_{45} & X_{46} & X_{47} \\ X_{48} & X_{49} & X_{50} & X_{51} & X_{52} & X_{53} & X_{54} & X_{55} \\ X_{56} & X_{57} & X_{58} & X_{59} & X_{60} & X_{61} & X_{62} & X_{63} \end{bmatrix} = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & 0 & 0 & 0 & 0 \\ B_8 & B_9 & B_{10} & B_{11} & 0 & 0 & 0 & 0 \\ B_{16} & B_{17} & B_{18} & B_{19} & 0 & 0 & 0 & 0 \\ B_{24} & B_{25} & B_{26} & B_{27} & 0 & 0 & 0 & 0 \\ B_{32} & B_{33} & B_{34} & B_{35} & 0 & 0 & 0 & 0 \\ B_{40} & B_{41} & B_{42} & B_{43} & 0 & 0 & 0 & 0 \\ B_{48} & B_{49} & B_{50} & B_{51} & 0 & 0 & 0 & 0 \\ B_{56} & B_{57} & B_{58} & B_{59} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} d & d & d & d & d & d & d & d \\ a & c & e & g & -g & -e & -c & -a \\ b & f & -f & -b & -b & -f & f & b \\ c & -g & -a & -e & e & a & g & -c \\ d & -d & -d & d & d & -d & -d & d \\ e & -a & g & c & -c & -g & a & -e \\ f & -b & b & -f & -f & b & -b & f \\ g & -e & c & -a & a & -c & e & -g \end{bmatrix}$$

$$X = B \qquad A$$

# 4.5 IDCT Fast Algorithm - 2

## Fast 16 Pixels IDCT(5/8)

- **File Name :idct.c**

```
--------------------------------------------------------------------------
static void idctcol16(blk)
short *blk;
{  int x0, x1, x2, x3;
   int  x4 x5, x6, x7, x8;
   int a0,a1,a2,a3,a4,a5;
   int o0,o1,o2,o3;
   int e0,e1,e2,e3;
   x0=blk[8*0];
   x1=blk[8*1];
   x2=blk[8*2];
   x3=blk[8*3];


   a0=d*x0;
   a1=b*x2;
   a2=f*x2;
--------------------------------------------------------------------------
```

# 4.5 IDCT Fast Algorithm - 2

## ■ Fast 16 Pixels IDCT(6/8)

- ◆ File Name :idct.c

```
---------------------------------------------------------------------------
  o0=a0+a1;
  o1=a0+a2;
  o2=a0-a2;
  o3=a0-a1;
  e0=(a*x1)+(c*x3);
  e1=(c*x1)-(g*x3);
  e2=(e*x1)-(a*x3);
  e3=(g*x1)-(e*x3);
  /* fourth stage */
  blk[8*0] = iclp[(o0+e0)>>14];
  blk[8*1] = iclp[(o1+e1)>>14];
  blk[8*2] = iclp[(o2+e2)>>14];
  blk[8*3] = iclp[(o3+e3)>>14];
  blk[8*4] = iclp[(o3-e3)>>14];
  blk[8*5] = iclp[(o2-e2)>>14];
  blk[8*6] = iclp[(o1-e1)>>14];
  blk[8*7] = iclp[(o0-e0)>>14];
  }
---------------------------------------------------------------------------
```

# 4.5 IDCT Fast Algorithm - 2

## ■ Fast 16 Pixels IDCT(7/8)

- File Name :idct.c

```
-----------------------------------------------------------------------
void Fast_IDCT16(block)
short *block;
{
 int i;
 for (i=0; i<8; i++){
   if(i<4)  {
   idctrow16(block+8*i);
  }else
  {
    block[8*i]=0;
    block[8*i+1]=0;
    block[8*i+2]=0;
    block[8*i+3]=0;
    block[8*i+4]=0;
    block[8*i+5]=0;
    block[8*i+6]=0;
    block[8*i+7]=0;
 }
}
  for (i=0; i<8; i++)
  {    idctcol16(block+i);  }
}
--------------------------------------------------------------------
```

# 4.5 IDCT Fast Algorithm - 2

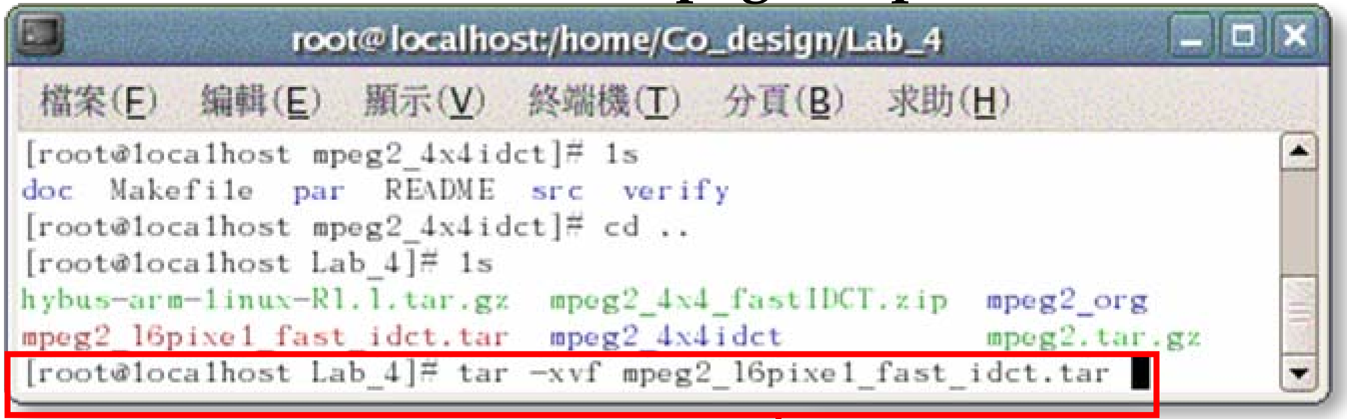
## ■ Fast 16 Pixels IDCT(8/8)

- **File Name :getpic.c**

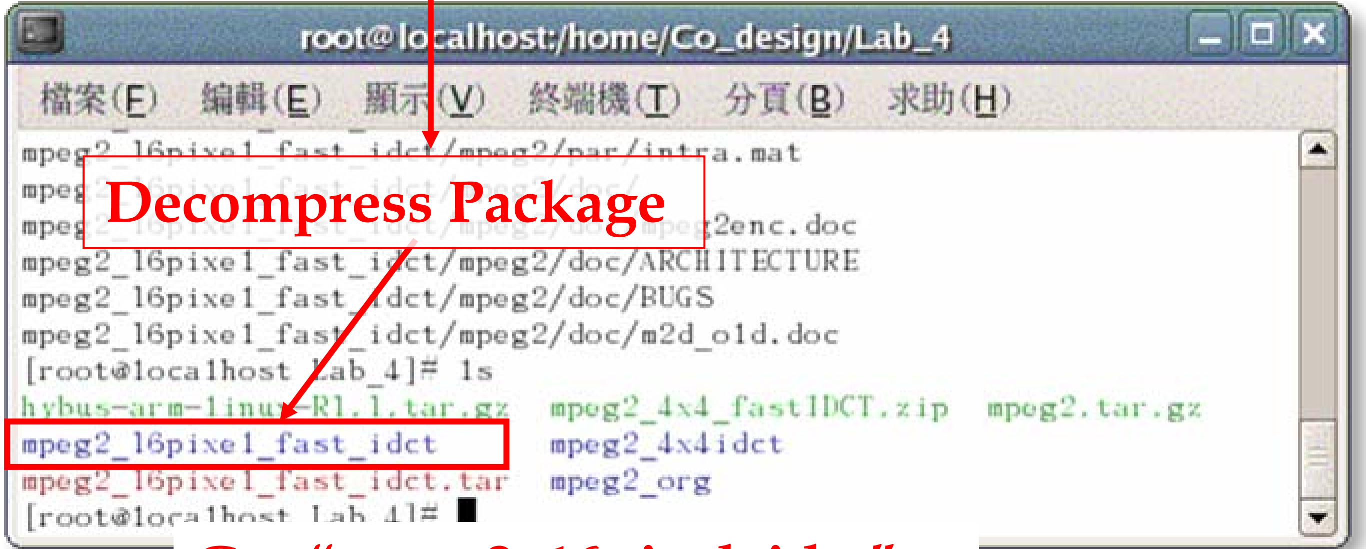
```
-----------------------------------------------------------------------
static void motion_compensation(
{
…
// if (Reference_IDCT_Flag)
//   Reference_IDCT(ld->block[comp]);
// else
//   Fast_IDCT(ld->block[comp]);
Fast_IDCT16(ld->block[comp]);
Add_Block(comp,bx,by,dct_type,(macroblock_type & MACROBLOCK_INTRA)==0);
}
-----------------------------------------------------------------------
```

# 4.5 IDCT Fast Algorithm - 2

- Decompress “mpeg2_16pixel_fast_idct.tar”
  - #Lab_4> tar –xvf mpeg2_16pixel_fast_idct.tar


root@localhost:/home/Co_design/Lab_4
檔案(F) 编輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
[root@localhost mpeg2_4x4idct]# ls
doc  Makefile  par  README  src  verify
[root@localhost mpeg2_4x4idct]# cd ..
[root@localhost Lab_4]# ls
hybus-arm-linux-R1.1.tar.gz  mpeg2_4x4_fastIDCT.zip  mpeg2_org
mpeg2_16pixel_fast_idct.tar  mpeg2_4x4idct           mpeg2.tar.gz
[root@localhost Lab_4]# tar -xvf mpeg2_16pixel_fast_idct.tar
```


root@localhost:/home/Co_design/Lab_4
檔案(F) 编輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

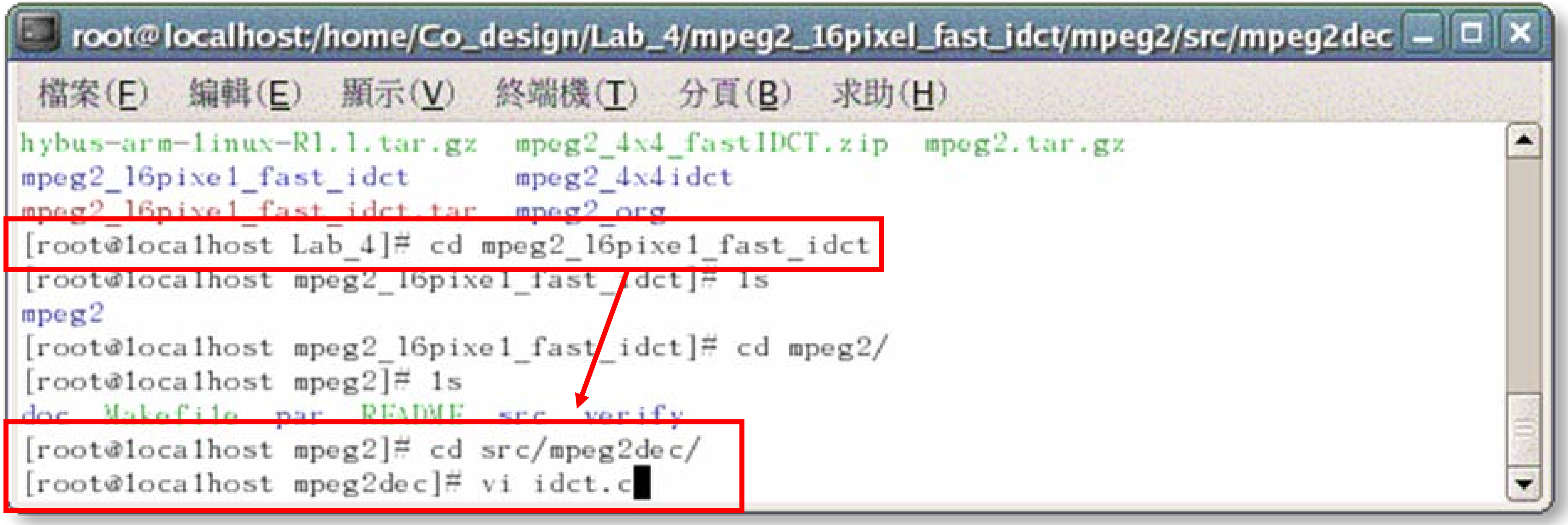
```
mpeg2_16pixel_fast_idct/mpeg2/par/intra.mat
mpeg2_16pixel_fast_idct/mpeg2/doc/
mpeg2_16pixel_fast_idct/mpeg2/doc/mpeg2enc.doc
mpeg2_16pixel_fast_idct/mpeg2/doc/ARCHITECTURE
mpeg2_16pixel_fast_idct/mpeg2/doc/BUGS
mpeg2_16pixel_fast_idct/mpeg2/doc/m2d_old.doc
[root@localhost Lab_4]# ls
hybus-arm-linux-R1.1.tar.gz  mpeg2_4x4_fastIDCT.zip  mpeg2.tar.gz
mpeg2_16pixel_fast_idct      mpeg2_4x4idct
mpeg2_16pixel_fast_idct.tar  mpeg2_org
[root@localhost Lab_4]#
```

Decompress Package

Get “mpeg2_16pixel_idct”

# 4.5 IDCT Fast Algorithm - 2

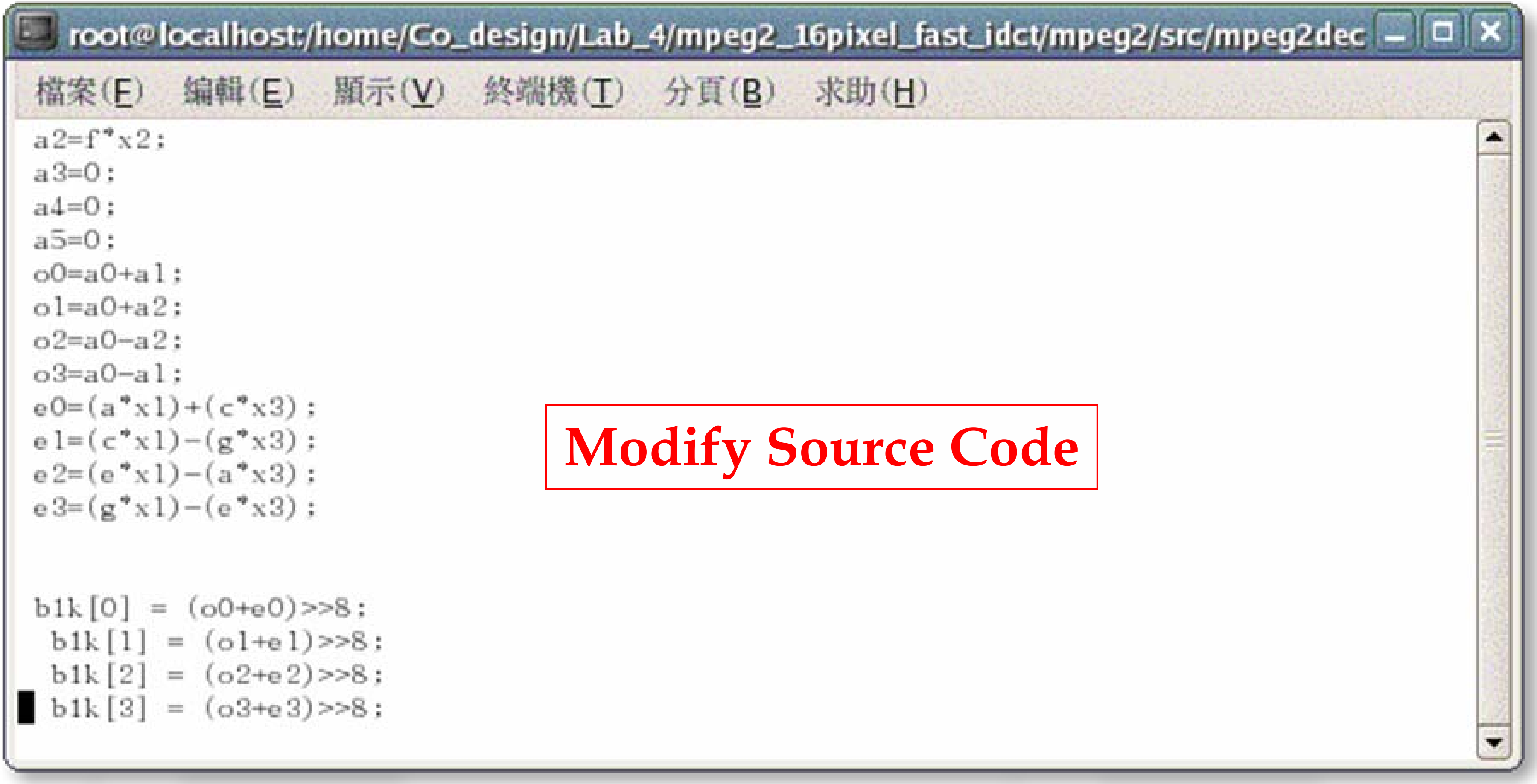
- The Files after Modify : getpic.c, idct.c
  - LAB4> cd mpeg2_16_idct/src/mpeg2dec/
  - Mpeg2dec>vi idct.c


root@localhost:/home/Co_design/Lab_4/mpeg2_16pixel_fast_idct/mpeg2/src/mpeg2dec
檔案(F)
編輯(E)
顯示(V)
終端機(T)
分頁(B)
求助(H)

```
hybus-arm-linux-R1.1.tar.gz    mpeg2_4x4_fastIDCT.zip   mpeg2.tar.gz
mpeg2_16pixel_fast_idct        mpeg2_4x4idct
mpeg2_16pixel_fast_idct.tar    mpeg2_org
[root@localhost Lab_4]# cd mpeg2_16pixel_fast_idct
[root@localhost mpeg2_16pixel_fast_idct]# ls
mpeg2
[root@localhost mpeg2_16pixel_fast_idct]# cd mpeg2/
[root@localhost mpeg2]# ls
doc  Makefile  par  README  src  verify
[root@localhost mpeg2]# cd src/mpeg2dec/
[root@localhost mpeg2dec]# vi idct.c
```

# 4.5 IDCT Fast Algorithm - 2

- Modify “idct.c” File
  - 16 pixel IDCT Fast Algorithm

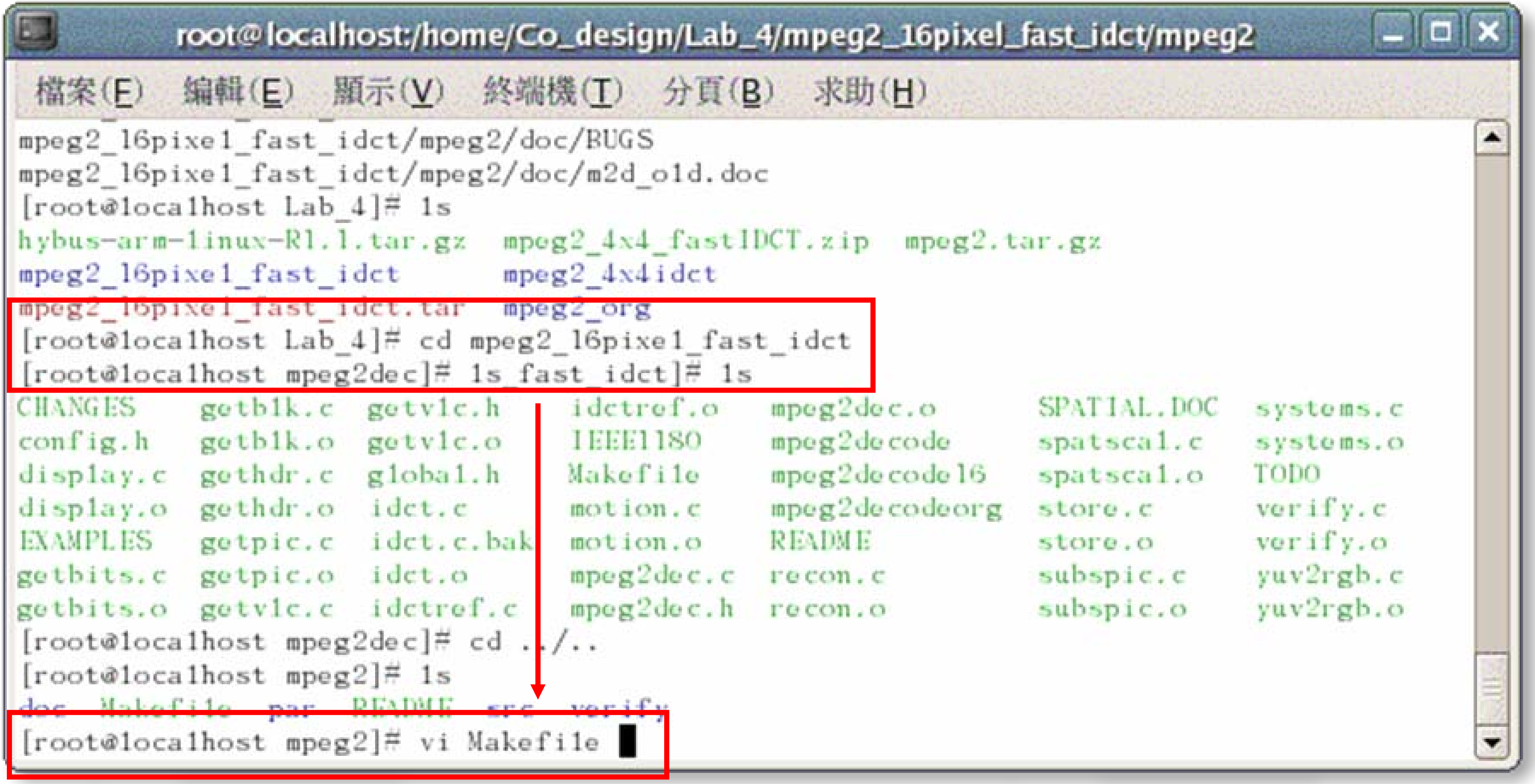

root@localhost:/home/Co_design/Lab_4/mpeg2_16pixel_fast_idct/mpeg2/src/mpeg2dec
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
a2=f*x2;
a3=0;
a4=0;
a5=0;
o0=a0+a1;
o1=a0+a2;
o2=a0-a2;
o3=a0-a1;
e0=(a*x1)+(c*x3);
e1=(c*x1)-(g*x3);
e2=(e*x1)-(a*x3);
e3=(g*x1)-(e*x3);


blk[0] = (o0+e0)>>8;
 blk[1] = (o1+e1)>>8;
 blk[2] = (o2+e2)>>8;
 blk[3] = (o3+e3)>>8;
```

Modify Source Code

# 4.5 IDCT Fast Algorithm - 2

- **Modify "Makefile"**
  - **Mpeg2_16_idct>vi Makefile**

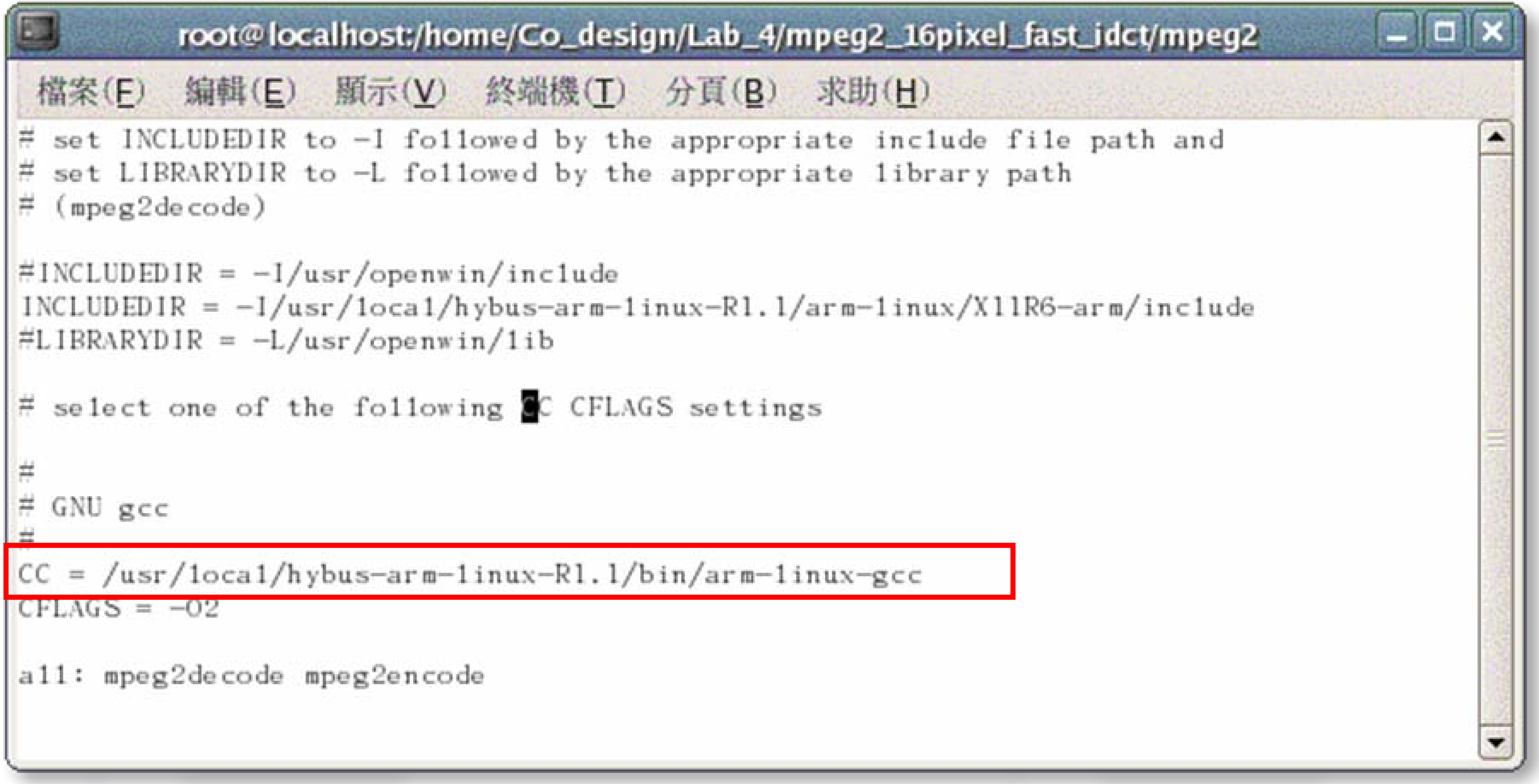


root@localhost:/home/Co_design/Lab_4/mpeg2_16pixel_fast_idct/mpeg2
檔案(F) 编輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
mpeg2_16pixel_fast_idct/mpeg2/doc/BUGS
mpeg2_16pixel_fast_idct/mpeg2/doc/m2d_old.doc
[root@localhost Lab_4]# ls
hybus-arm-linux-R1.1.tar.gz  mpeg2_4x4_fastIDCT.zip  mpeg2.tar.gz
mpeg2_16pixel_fast_idct      mpeg2_4x4idct
mpeg2_16pixel_fast_idct.tar  mpeg2_org
[root@localhost Lab_4]# cd mpeg2_16pixel_fast_idct
[root@localhost mpeg2dec]# ls_fast_idct]# ls
CHANGES    getblk.c  getvlc.h    idctref.o  mpeg2dec.o        SPATIAL.DOC  systems.c
config.h   getblk.o  getvlc.o    IEEE1180   mpeg2decode       spatscal.c   systems.o
display.c  gethdr.c  global.h    Makefile   mpeg2decode16     spatscal.o   TODO
display.o  gethdr.o  idct.c      motion.c   mpeg2decodeorg    store.c      verify.c
EXAMPLES   getpic.c  idct.c.bak  motion.o   README            store.o      verify.o
getbits.c  getpic.o  idct.o      mpeg2dec.c recon.c           subspic.c    yuv2rgb.c
getbits.o  getvlc.c  idctref.c   mpeg2dec.h recon.o           subspic.o    yuv2rgb.o
[root@localhost mpeg2dec]# cd ../..
[root@localhost mpeg2]# ls
doc  Makefile  par  README  src  verify
[root@localhost mpeg2]# vi Makefile
```

# 4.5 IDCT Fast Algorithm - 2

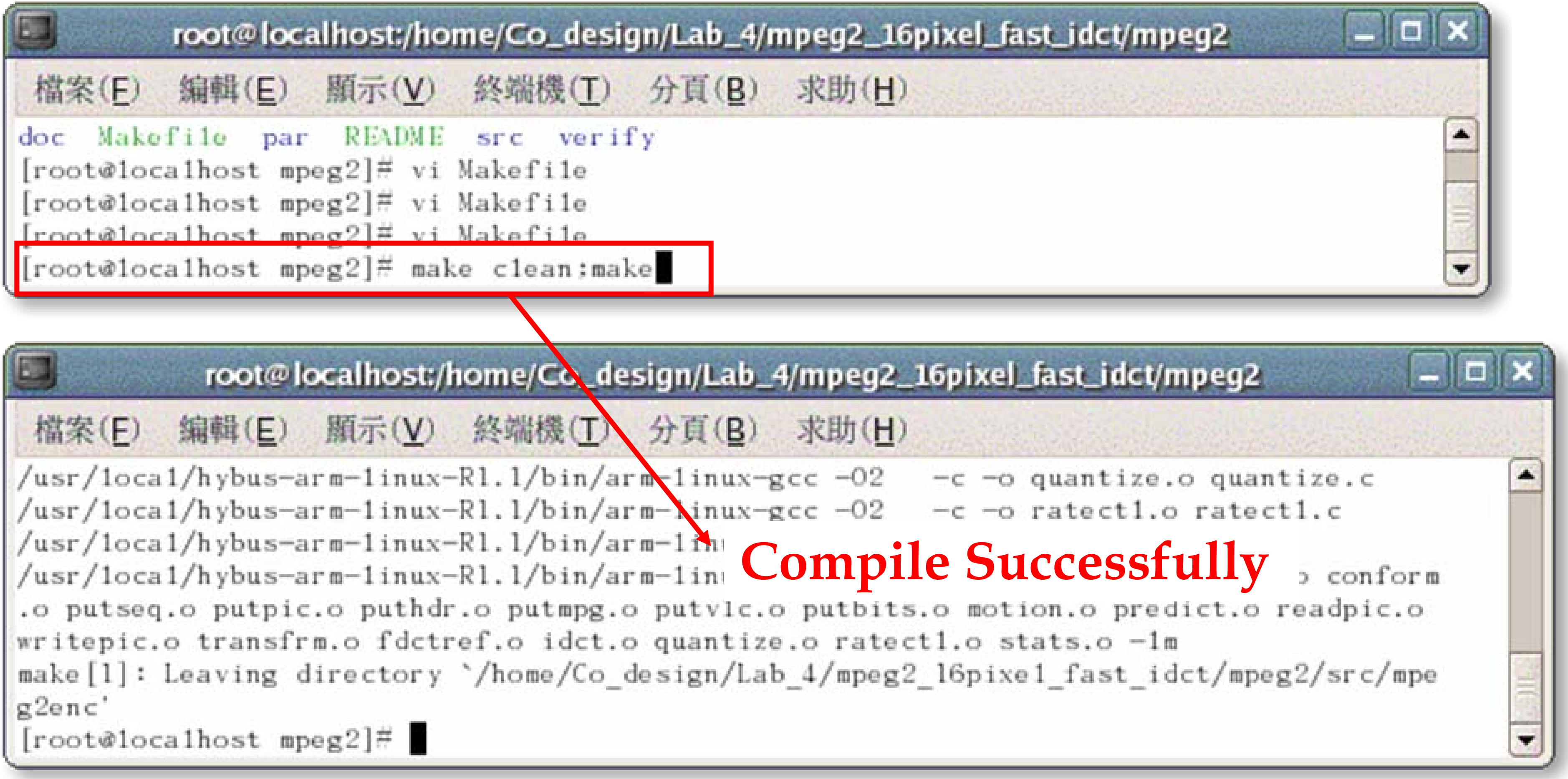
- Modify the Path of Cross-compiler Tool Chain
  - Modify CC= /usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc


root@localhost:/home/Co_design/Lab_4/mpeg2_16pixel_fast_idct/mpeg2
檔案(F)
編輯(E)
顯示(V)
終端機(T)
分頁(B)
求助(H)

```
# set INCLUDEDIR to -I followed by the appropriate include file path and
# 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 = /usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc
CFLAGS = -O2

all: mpeg2decode mpeg2encode
```

# 4.5 IDCT Fast Algorithm - 2

- Compile Source Code
  - mpeg2_16_idct>make clean;make


root@localhost:/home/Co_design/Lab_4/mpeg2_16pixel_fast_idct/mpeg2
檔案(F) 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]# vi Makefile
[root@localhost mpeg2]# vi Makefile
[root@localhost mpeg2]# vi Makefile
[root@localhost mpeg2]# make clean:make
```

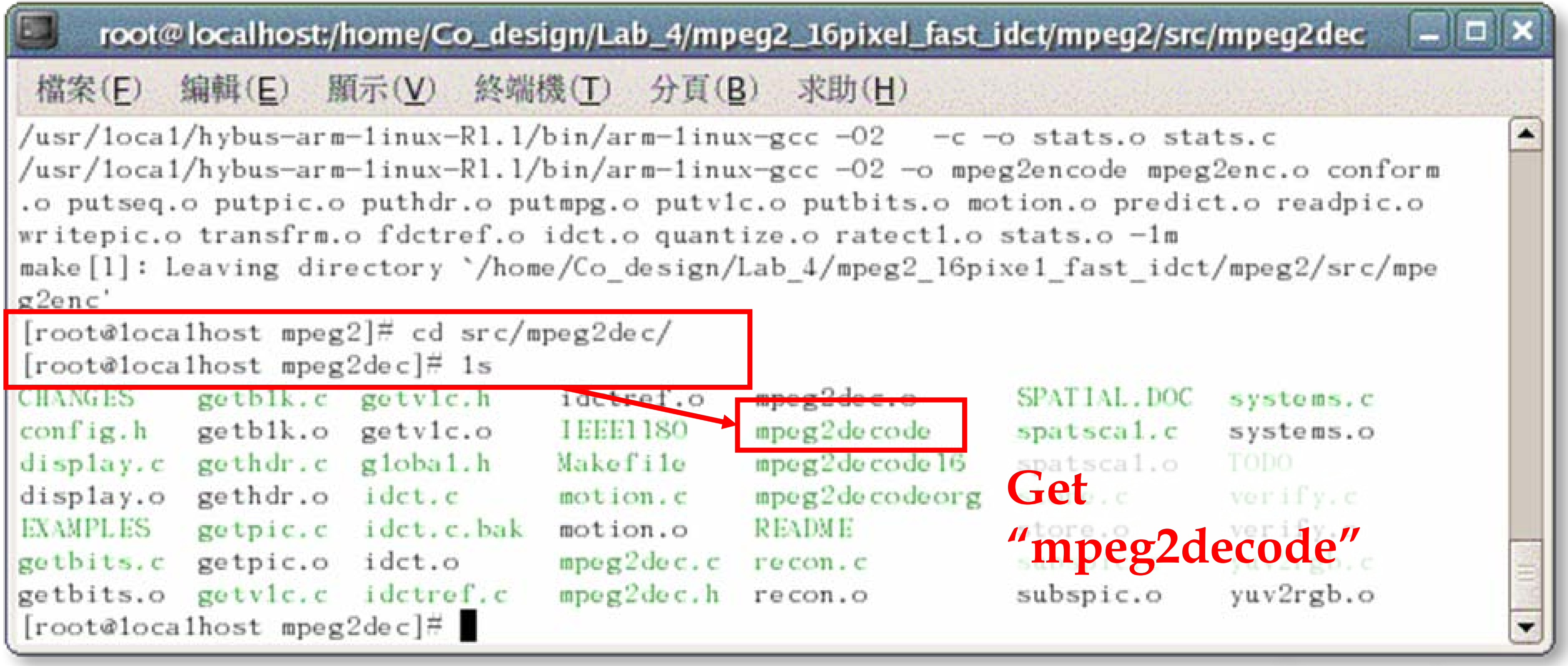
root@localhost:/home/Co_design/Lab_4/mpeg2_16pixel_fast_idct/mpeg2
檔案(F) 編輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o quantize.o quantize.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o ratectl.o ratectl.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-lin
/usr/local/hybus-arm-linux-R1.1/bin/arm-lin                                 conform
.o putseq.o putpic.o puthdr.o putmpg.o putvlc.o putbits.o motion.o predict.o readpic.o
writepic.o transfrm.o fdctref.o idct.o quantize.o ratectl.o stats.o -lm
make[1]: Leaving directory `/home/Co_design/Lab_4/mpeg2_16pixel_fast_idct/mpeg2/src/mpe
g2enc'
[root@localhost mpeg2]#
```

Compile Successfully

# 4.5 IDCT Fast Algorithm - 2

- Compile Source Code
  - mpeg2_16_idct>cd src/mpeg2dec/

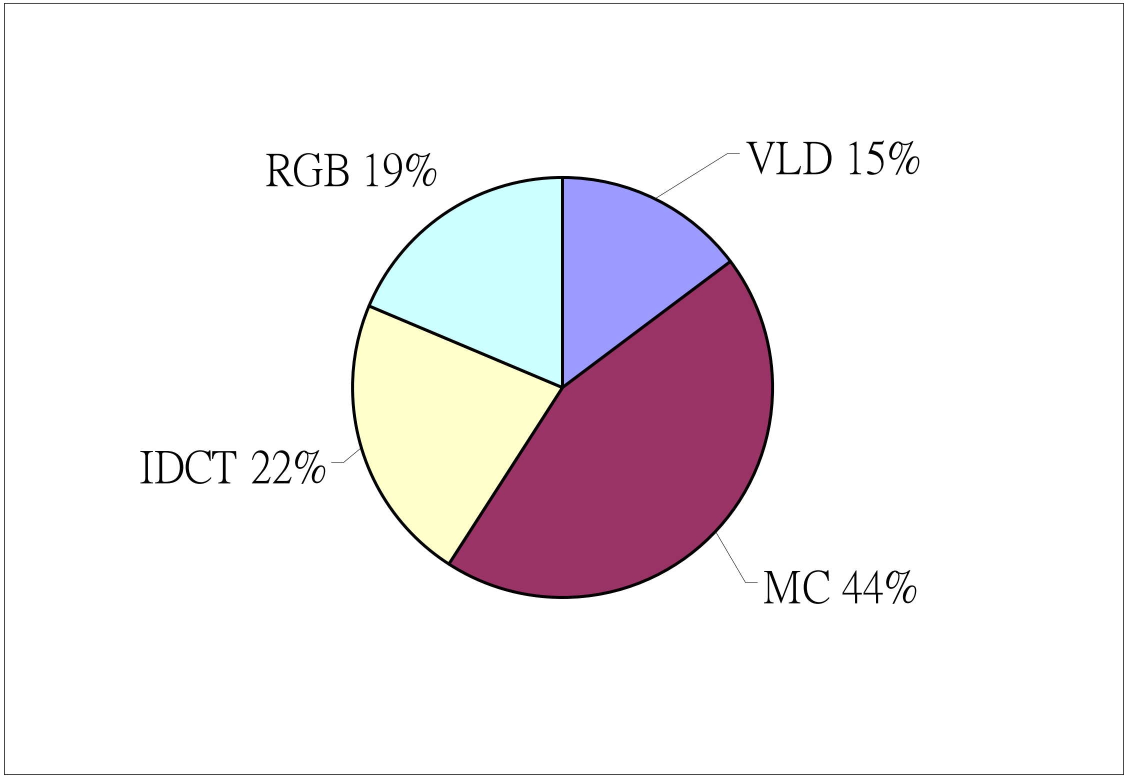

root@localhost:/home/Co_design/Lab_4/mpeg2_16pixel_fast_idct/mpeg2/src/mpeg2dec
檔案(F) 编輯(E) 顯示(V) 終端機(T) 分頁(B) 求助(H)

```
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2   -c -o stats.o stats.c
/usr/local/hybus-arm-linux-R1.1/bin/arm-linux-gcc -O2 -o mpeg2encode mpeg2enc.o conform
.o putseq.o putpic.o puthdr.o putmpg.o putvlc.o putbits.o motion.o predict.o readpic.o
writepic.o transfrm.o fdctref.o idct.o quantize.o ratectl.o stats.o -lm
make[1]: Leaving directory `/home/Co_design/Lab_4/mpeg2_16pixel_fast_idct/mpeg2/src/mpe
g2enc'
[root@localhost mpeg2]# cd src/mpeg2dec/
[root@localhost mpeg2dec]# ls
CHANGES    getblk.c  getvlc.h    idctref.o   mpeg2dec.o       SPATIAL.DOC  systems.c
config.h   getblk.o  getvlc.o    IEEE1180    mpeg2decode      spatscal.c   systems.o
display.c  gethdr.c  global.h    Makefile    mpeg2decode16    spatscal.o   TODO
display.o  gethdr.o  idct.c      motion.c    mpeg2decodeorg   store.c      verify.c
EXAMPLES   getpic.c  idct.c.bak  motion.o    README           store.o      verify.o
getbits.c  getpic.o  idct.o      mpeg2dec.c  recon.c          subspic.c    yuv2rgb.c
getbits.o  getvlc.c  idctref.c   mpeg2dec.h  recon.o          subspic.o    yuv2rgb.o
[root@localhost mpeg2dec]# 
```

Get "mpeg2decode"

# 4.5 IDCT Fast Algorithm - 2

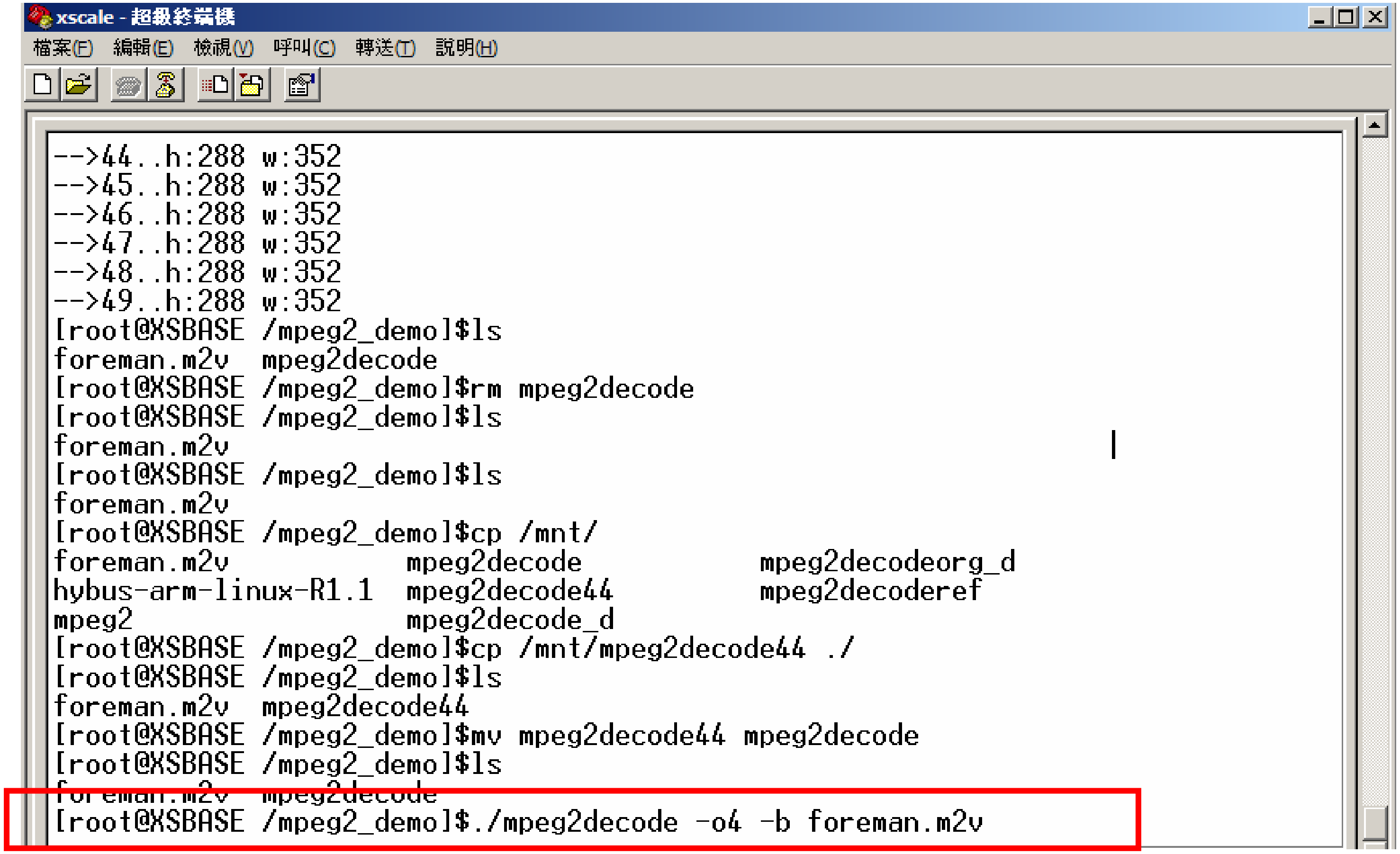
## ■ Performance Analysis



**IDCT : 49% -> 22%**

# 4.5 IDCT Fast Algorithm - 2

- Execute the MPEG-2 Decoder File
  - Execute MPEG-2 decoder with the fast IDCT algorithm
  - #mpeg2_demo>./mpeg2decode -o4 -b foreman.m2v



```
-->44..h:288 w:352
-->45..h:288 w:352
-->46..h:288 w:352
-->47..h:288 w:352
-->48..h:288 w:352
-->49..h:288 w:352
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v   mpeg2decode
[root@XSBASE /mpeg2_demo]$rm mpeg2decode
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v
[root@XSBASE /mpeg2_demo]$cp /mnt/
foreman.m2v           mpeg2decode           mpeg2decodeorg_d
hybus-arm-linux-R1.1  mpeg2decode44         mpeg2decoderef
mpeg2                 mpeg2decode_d
[root@XSBASE /mpeg2_demo]$cp /mnt/mpeg2decode44 ./
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v   mpeg2decode44
[root@XSBASE /mpeg2_demo]$mv mpeg2decode44 mpeg2decode
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v   mpeg2decode
[root@XSBASE /mpeg2_demo]$./mpeg2decode -o4 -b foreman.m2v
```

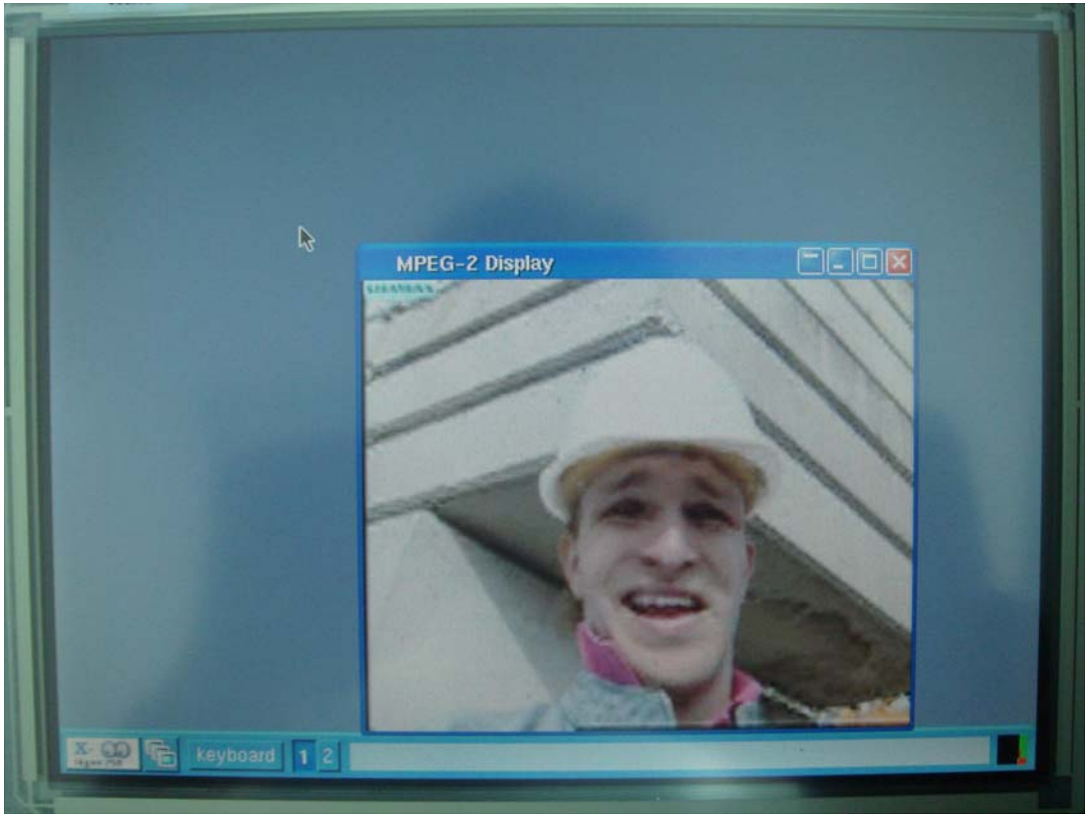
# 4.5 IDCT Fast Algorithm - 2

## Performance Report

- The Method to Improve the Performance
  - Fast Algorithm
    - IDCT Fast Algorithm – 2
- Total Execute Time
  - Execute Time =  5.94 Seconds
  - Frame Rate     =  8.41 fps
    - fps : Frame Per Second

# 4.5 IDCT Fast Algorithm - 2

- Display on TFT LCD
  - Execute with 16 pixel fast IDCT



# 4.6 Insert Assembly Code in C Language

4.1 Preparing

4.2 MPEG-2 Decoder Software

4.3 IDCT Source Code

4.4 IDCT Fast Algorithm - 1

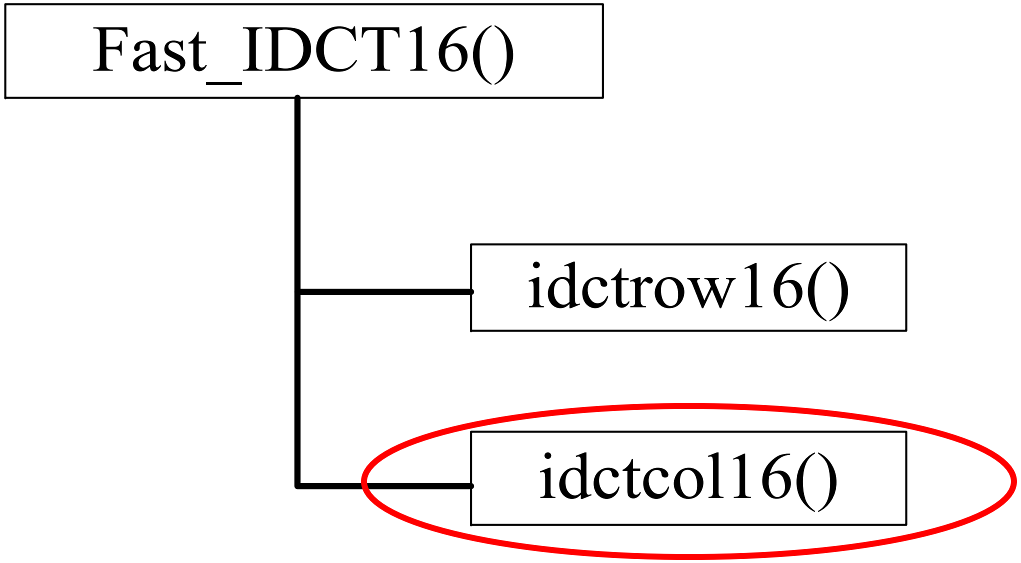
4.5 IDCT Fast Algorithm – 2

4.6 Insert Assembly Code in C Language

# 4.6 Insert Assembly Code in C Language

## Modify Function

- Modify Source Code
  - mpeg2_16pixel_asm.tar
  - Modify Files : idct.c
  - Modify Function
    - idctcol16();

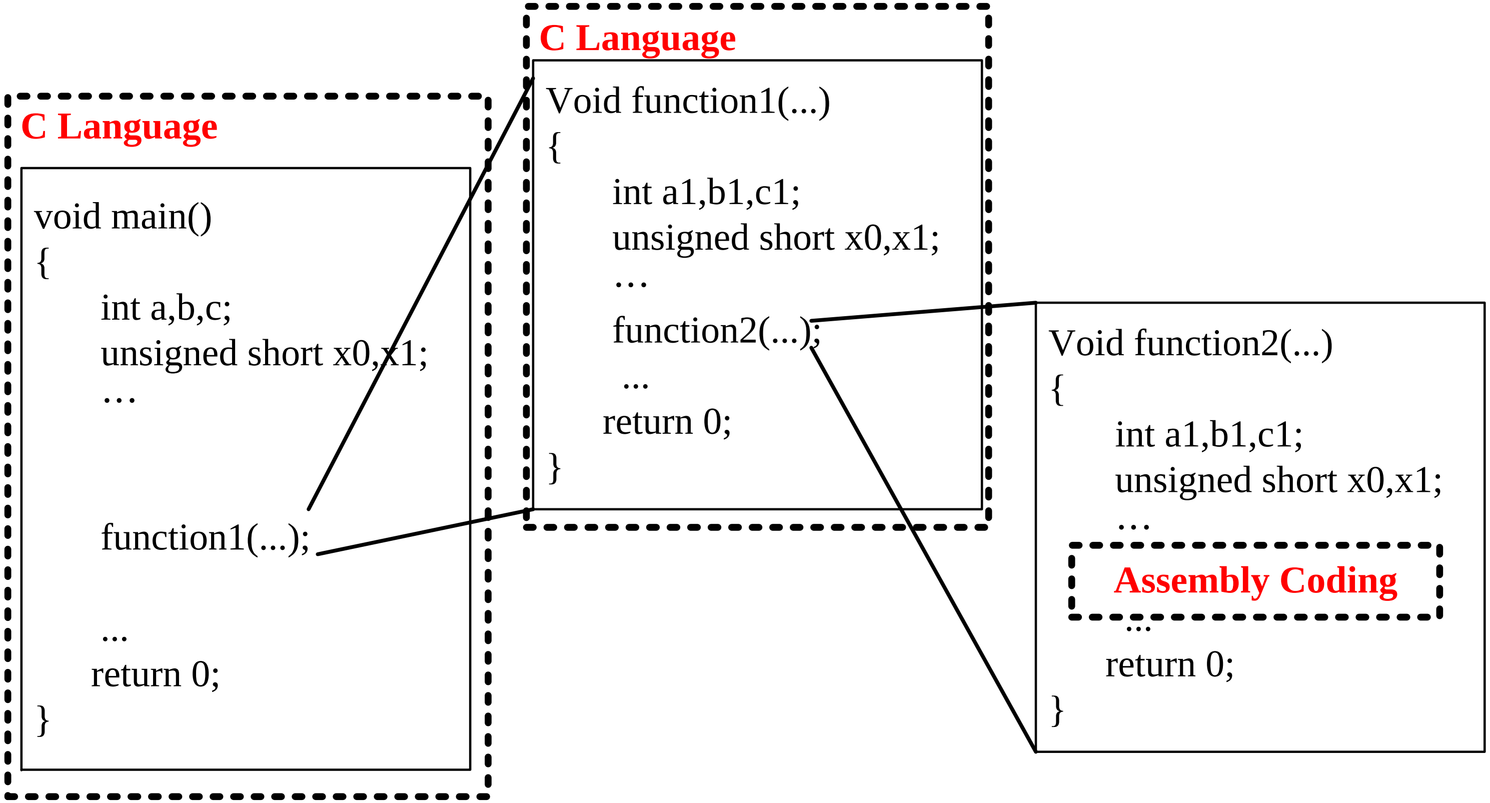


Assembly Coding

# 4.6 Insert Assembly Code in C Language

## ■ Insert Assembly Code in C Language

- ◆ Insert Assembly Code in C Language
  - ■ Assembly Coding in C Language



C Language

```
void main()
{
        int a,b,c;
        unsigned short x0,x1;
        …


        function1(...);


        ...
      return 0;
}
```

C Language

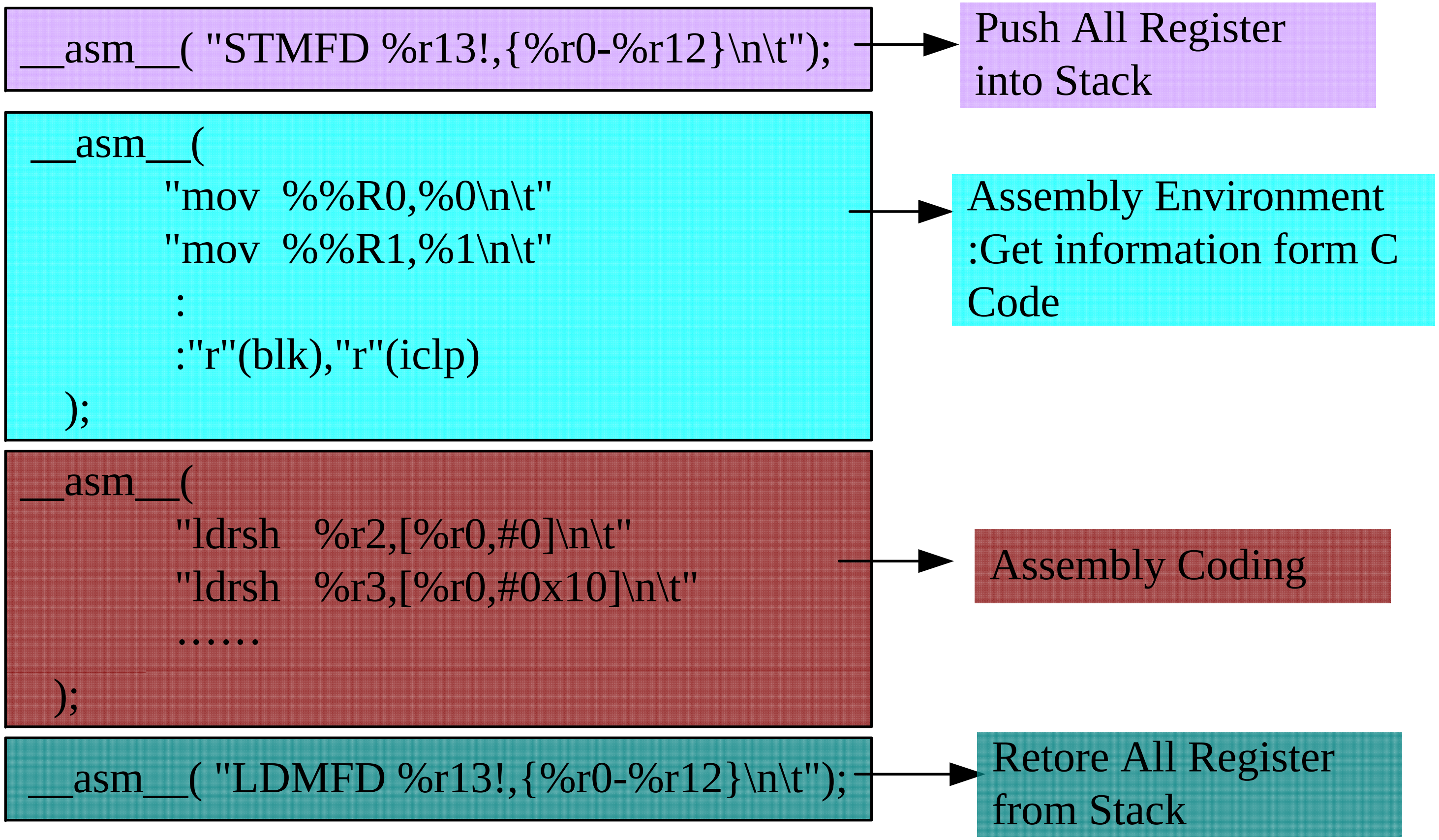
```
Void function1(...)
{
        int a1,b1,c1;
        unsigned short x0,x1;
        …
        function2(...);
         ...
      return 0;
}
```

```
Void function2(...)
{
        int a1,b1,c1;
        unsigned short x0,x1;
        …
        Assembly Coding
        ...
      return 0;
}
```

# 4.6 Insert Assembly Code in C Language

## ■ Insert Assembly Code in C Language

◆ **Insert Assembly Code in C Language**



```
__asm__( "STMFD %r13!,{%r0-%r12}\n\t");
```

Push All Register into Stack

```
__asm__(
            "mov  %%R0,%0\n\t"
            "mov  %%R1,%1\n\t"
             :
             :"r"(blk),"r"(iclp)
   );
```

Assembly Environment :Get information form C Code

```
__asm__(
             "ldrsh   %r2,[%r0,#0]\n\t"
             "ldrsh   %r3,[%r0,#0x10]\n\t"
             ……
   );
```

Assembly Coding

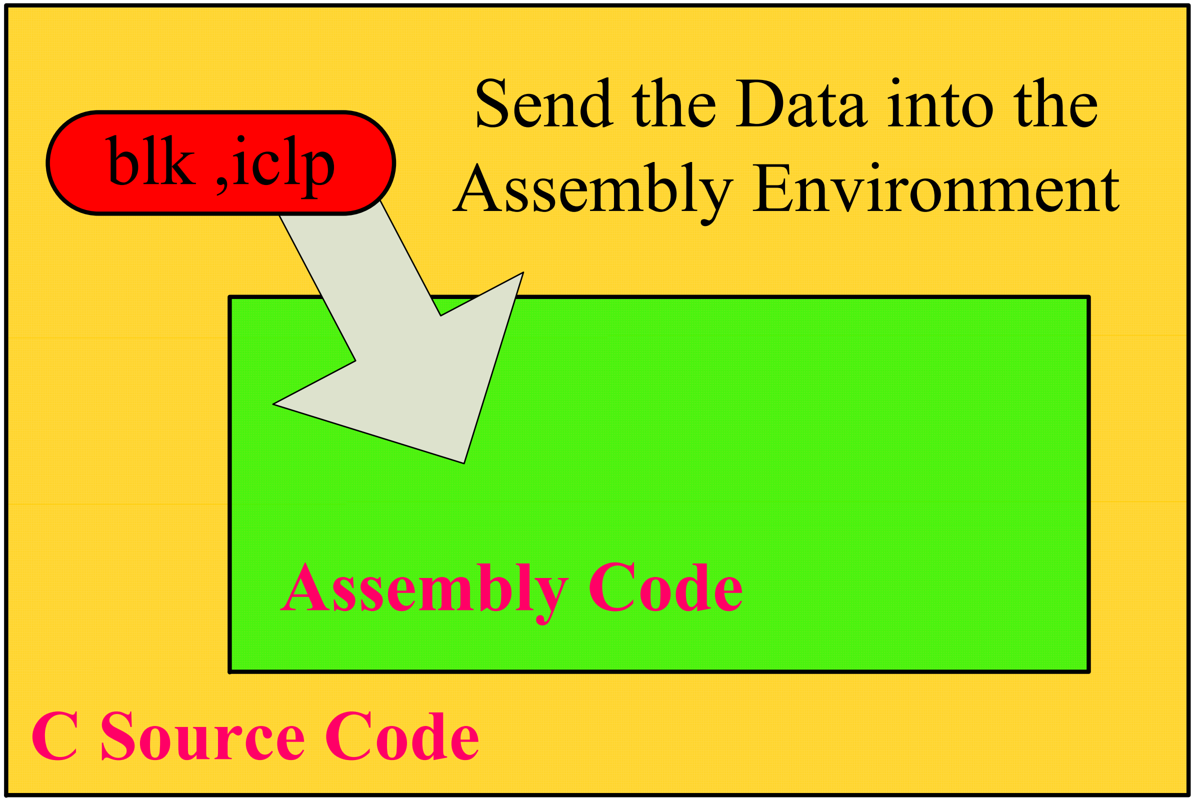
```
__asm__( "LDMFD %r13!,{%r0-%r12}\n\t");
```

Retore All Register from Stack

# 4.6 Insert Assembly Code in C Language

## ■ Insert Assembly Code in C Language

- ◆ Insert Assembly Code in C Language



# 4.6 Insert Assembly Code in C Language

## ■ C Source Code

- ◆ C Source Code to Modify
- ◆ File Name :idct.c

```
------------------------------------
 static void idctcol16(blk)
 short *blk;
{
  int x0, x1, x2, x3, x4;
  int x5, x6, x7, x8;
  int a0,a1,a2,a3,a4,a5;
  int o0,o1,o2,o3;
  int e0,e1,e2,e3;
  x0=blk[8*0];
  x1=blk[8*1];
  x2=blk[8*2];
  x3=blk[8*3];
  a0=d*x0;
  a1=b*x2;
  a2=f*x2;
------------------------------------
```

```
------------------------------------
  o0=a0+a1;
  o1=a0+a2;
  o2=a0-a2;
  o3=a0-a1;
  e0=(a*x1)+(c*x3);
  e1=(c*x1)-(g*x3);
  e2=(e*x1)-(a*x3);
  e3=(g*x1)-(e*x3);

 /* fourth stage */
blk[8*0] = iclp[(o0+e0)>>14];
blk[8*1] = iclp[(o1+e1)>>14];
blk[8*2] = iclp[(o2+e2)>>14];
blk[8*3] = iclp[(o3+e3)>>14];
blk[8*4] = iclp[(o3-e3)>>14];
blk[8*5] = iclp[(o2-e2)>>14];
blk[8*6] = iclp[(o1-e1)>>14];
blk[8*7] = iclp[(o0-e0)>>14];
 }
------------------------------------
```

# 4.6 Insert Assembly Code in C Language

## ■ Assembly Coding ( 1/5 )

- **File Name :idct.c**

```
----------------------------------------------------------------
static void idctcol16(blk)
short *blk;
{
  __asm__( "STMFD %r13!,{%r0-%r12}\n\t");/*Push All Registers into the
   __asm__(                                 Stack*/
           "mov  %%R0,%0\n\t"
           "mov  %%R1,%1\n\t“               /* Assembly Environment :
           :                                Get information form C Code */
           :"r"(blk),"r"(iclp)
          );
  __asm__(                                      /* Assembly Coding */
           "ldrsh   %r2,[%r0,#0]\n\t"          /* x0=blk[8*0] */
           "ldrsh   %r3,[%r0,#0x10]\n\t"       /* x1=blk[8*1] */
           "ldrsh   %r4,[%r0,#0x20]\n\t"       /* x2=blk[8*2] */
           "ldrsh   %r5,[%r0,#0x30]\n\t"       /* x3=blk[8*3] */

           "mov     %r9,#724\n\t"
           "mul     %r6,%r9,%r2\n\t"           /* a0=d*x0 */
  ------------------------------------------------------------------
```

# 4.6 Insert Assembly Code in C Language

## ■ Assembly Coding ( 2/5 )

- File Name :idct.c

```
-----------------------------------------------------------
"mov     %r9,#944\n\t“
"add     %r9,%r9,#2\n\t“
"mul     %r7,%r9,%r4\n\t"   /* a1=b*x2 */

"mov     %r9,#392\n\t“
"mul     %r8,%r9,%r4\n\t"   /* a2=f*x2 */

"add     %r2,%r6,%r7\n\t"   /* o0=a0+a1 */
"sub     %r7,%r6,%r7\n\t"   /* o3=a0-a1 */
"add     %r4,%r6,%r8\n\t"   /* o1=a0+a2 */
"sub     %r8,%r6,%r8\n\t"   /* o2=a0-a2 */

"mov    %r11,#1004\n\t"
"mul    %r12,%r11,%r3\n\t"  /* a*x1 */
"mov    %r11,#0x350\n\t"    /* 353 */
"add    %r11,%r11,#3\n\t"
"mul    %r10,%r11,%r5\n\t"  /* c*x3 */
"add    %r6,%r10,%r12\n\t"  /* e0=(a*x1)+(c*x3)--r6 */


---------------------------------------------------------
```

# 4.6 Insert Assembly Code in C Language

## ■ Assembly Coding ( 3/5 )

◆ File Name :idct.c

```
--------------------------------------------------------------------
"mov    %r11,#0x350\n\t“
"add    %r11,%r11,#3\n\t“
"mul    %r12,%r11,%r3\n\t"  /* c*x1 */

"mov    %r11,#0xc8\n\t"
"mul    %r10,%r11,%r5\n\t"  /* g*x3 */
"add    %r9,%r10,%r12\n\t"  /* e1=(c*x1)+(g*x3)--r9 */

"mov    %r11,#0x230\n\t“     /* 239 */
"add    %r11,%r11,#9\n\t"
"mul    %r12,%r11,%r3\n\t"  /* e*x1 */
"mov    %r11,#0x3ec\n\t"
"mul    %r10,%r11,%r5\n\t"  /* a*x3 */
"add    %r10,%r10,%r12\n\t" /* e0=(e*x1)+(a*x3)--r10 */

"mov    %r11,#0xc8\n\t"
"mul    %r12,%r11,%r3\n\t"  /* g*x1 */
"mov    %r11,#0x230\n\t“
"add    %r11,%r11,#9\n\t“
"mul    %r3,%r11,%r5\n\t"   /* e*x3 */
"add    %r11,%r12,%r3\n\t"  /* e0=(a*x1)+(c*x3)--r11 */
--------------------------------------------------------------------
```

# 4.6 Insert Assembly Code in C Language

## ■ Assembly Coding ( 4/5 )

- File Name :idct.c

```
-----------------------------------
/* blk[8*0]=iclip[(o0+e0)>>14] */
 "add    %r3,%r2,%r6\n\t"
 "mov    %r5,r3,asr #14\n\t"
 "add    %r3,%r1,%r5,lsl #1\n\t"
 "ldrh   %r5,[%r3]\n\t"
 "strh   %r5,[%r0,#0]\n\t"

/* blk[8*1]=iclip[(o1+e1)>>14] */
 "add    %r3,%r4,%r9\n\t"
 "mov    %r5,r3,asr #14\n\t"
 "add    %r3,%r1,%r5,lsl #1\n\t"
 "ldrh   %r5,[%r3]\n\t"
 "strh   %r5,[%r0,#0x10]\n\t"

/* blk[8*2]=iclip[(o2+e2)>>14] */
 "add    %r3,%r8,%r10\n\t"
 "mov    %r5,r3,asr #14\n\t"
 "add    %r3,%r1,%r5,lsl #1\n\t"
 "ldrh   %r5,[%r3]\n\t"
 "strh   %r5,[%r0,#0x20]\n\t"
-----------------------------------
```

```
-----------------------------------
/* blk[8*3]=iclip[(o3+e3)>>14] */
 "add    %r3,%r7,%r11\n\t"
 "mov    %r5,r3,asr #14\n\t"
 "add    %r3,%r1,%r5,lsl #1\n\t"
 "ldrh   %r5,[%r3]\n\t"
 "strh   %r5,[%r0,#0x30]\n\t"

/* blk[8*4]=iclip[(o3-e3)>>14] */
"sub    %r3,%r7,%r11\n\t"
"mov    %r5,r3,asr #14\n\t"
"add    %r3,%r1,%r5,lsl #1\n\t"
"ldrh   %r5,[%r3]\n\t"
"strh   %r5,[%r0,#0x40]\n\t"

/* blk[8*5]=iclip[(o2-e2)>>14] */
"sub    %r3,%r8,%r10\n\t"
"mov    %r5,r3,asr #14\n\t"
"add    %r3,%r1,%r5,lsl #1\n\t"
"ldrh   %r5,[%r3]\n\t"
"strh   %r5,[%r0,#0x50]\n\t"
-----------------------------------
```

# 4.6 Insert Assembly Code in C Language

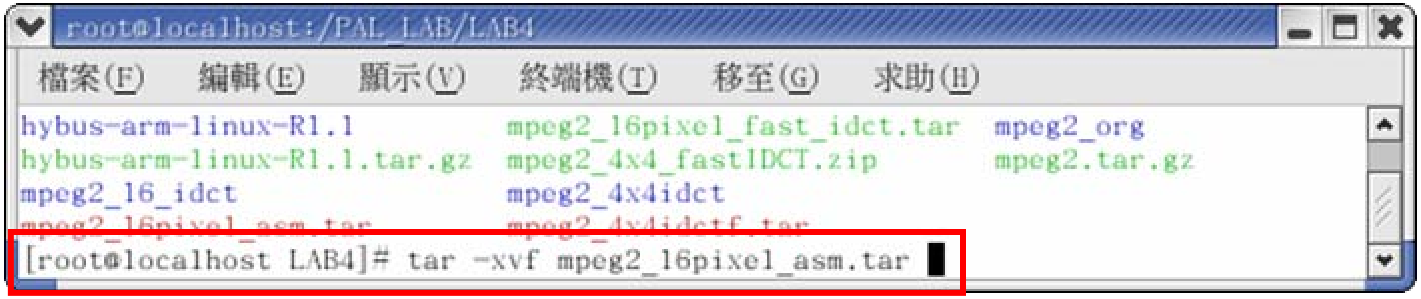
## ■ Assembly Coding ( 5/5 )

- File Name :idct.c

```
--------------------------------------------------------------------
          /* blk[8*6]=iclip[(o1-e1)>>14] */
            "sub    %r3,%r4,%r9\n\t"
            "mov    %r5,r3,asr #14\n\t"
            "add    %r3,%r1,%r5,lsl #1\n\t"
            "ldrh   %r5,[%r3]\n\t"
            "strh   %r5,[%r0,#0x60]\n\t"

          /* blk[8*7]=iclip[(o0-e0)>>14] */
            "sub    %r3,%r2,%r6\n\t"
            "mov    %r5,r3,asr #14\n\t"
            "add    %r3,%r1,%r5,lsl #1\n\t"
            "ldrh   %r5,[%r3]\n\t"
            "strh   %r5,[%r0,#0x70]\n\t"
     );

    __asm__( "LDMFD %r13!,{%r0-%r12}\n\t");   /*  Retore All Register from Stack  */
 }

--------------------------------------------------------------------
```

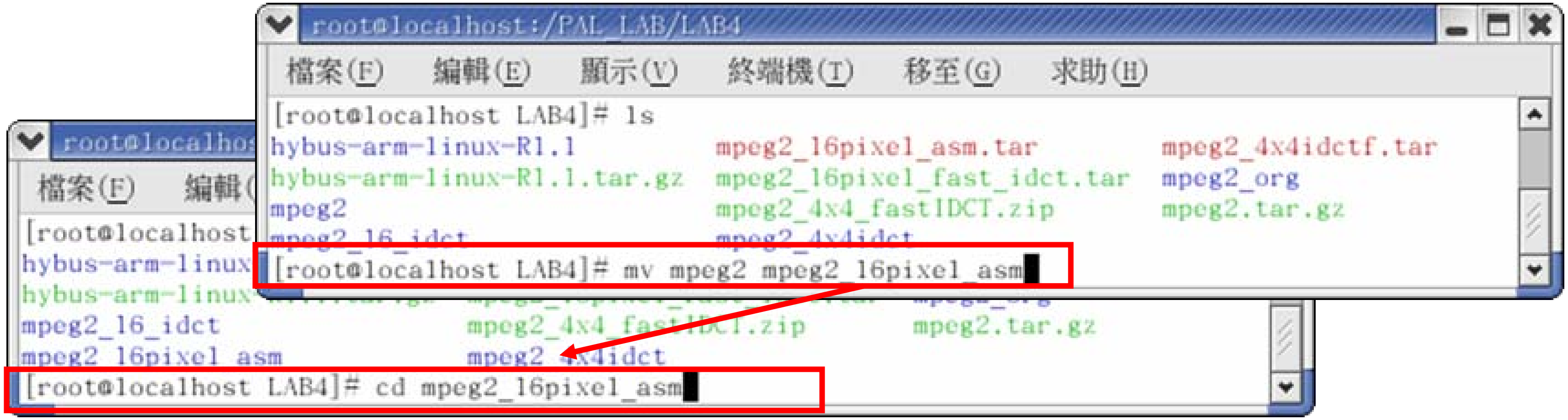
# 4.6 Insert Assembly Code in C Language

- Decompress “mpeg2_16pixel_asm.tar”
  - #LAB4> tar –xvf mpeg2_16pixel_asm.tar


root@localhost:/PAL_LAB/LAB4
檔案(F) 编輯(E) 顯示(V) 終端機(T) 移至(G) 求助(H)

```
hybus-arm-linux-R1.1           mpeg2_16pixel_fast_idct.tar  mpeg2_org
hybus-arm-linux-R1.1.tar.gz    mpeg2_4x4_fastIDCT.zip       mpeg2.tar.gz
mpeg2_16_idct                  mpeg2_4x4idct
mpeg2_16pixel_asm.tar          mpeg2_4x4idctf.tar
[root@localhost LAB4]# tar -xvf mpeg2_16pixel_asm.tar
```

- Change the Name of the Directory and Enter
  - #LAB4>mv mpeg2 mpeg2_16pixel_asm
  - #LAB4> cd mpeg2_16pixel_asm

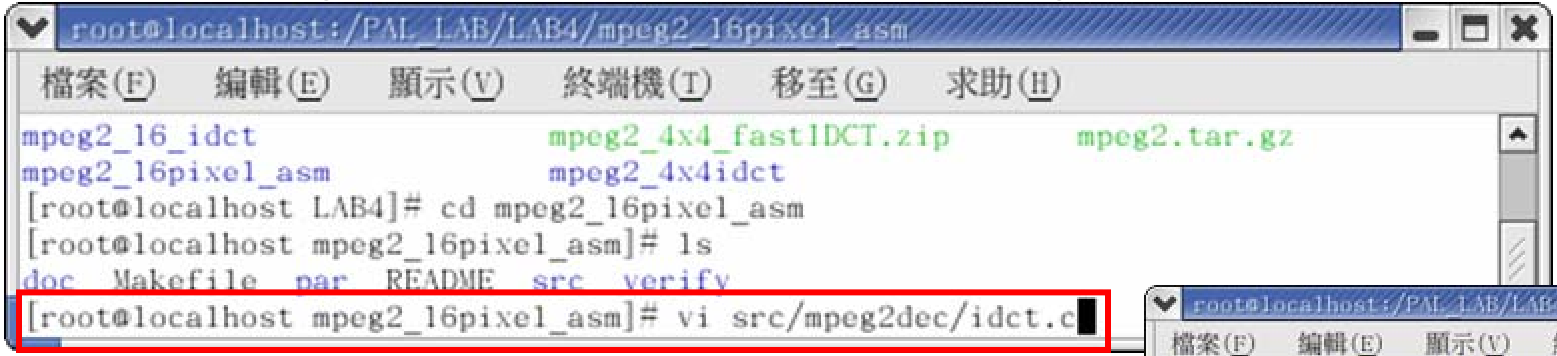

root@localhost:/PAL_LAB/LAB4
檔案(F) 编輯(E) 顯示(V) 終端機(T) 移至(G) 求助(H)
root@localhos
檔案(F) 编輯(

```
[root@localhost LAB4]# ls
hybus-arm-linux-R1.1         mpeg2_16pixel_asm.tar        mpeg2_4x4idctf.tar
hybus-arm-linux-R1.1.tar.gz  mpeg2_16pixel_fast_idct.tar  mpeg2_org
mpeg2                        mpeg2_4x4_fastIDCT.zip       mpeg2.tar.gz
mpeg2_16_idct                mpeg2_4x4idct
[root@localhost LAB4]# mv mpeg2 mpeg2_16pixel_asm
```

```
[root@localhost
hybus-arm-linux
hybus-arm-linux
mpeg2_16_idct                mpeg2_4x4_fastIDCT.zip       mpeg2.tar.gz
mpeg2_16pixel_asm            mpeg2_4x4idct
[root@localhost LAB4]# cd mpeg2_16pixel_asm
```

# 4.6 Insert Assembly Code in C Language

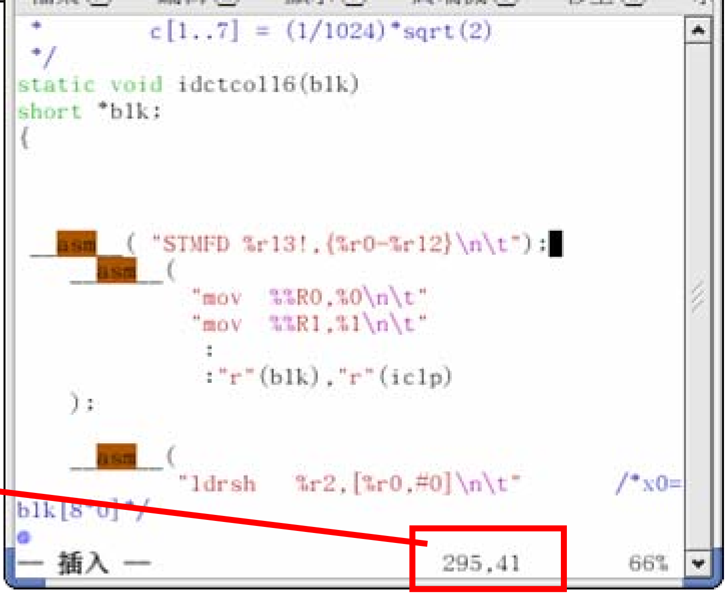
- Assembly Coding
  - Source Code : src/mpeg2dec/idct.c
  - #mpeg2_16pixel_asm>vi src/mpeg2dec/idct.c



```
mpeg2_16_idct              mpeg2_4x4_fastIDCT.zip       mpeg2.tar.gz
mpeg2_16pixel_asm          mpeg2_4x4idct
[root@localhost LAB4]# cd mpeg2_16pixel_asm
[root@localhost mpeg2_16pixel_asm]# ls
doc  Makefile  par  README  src  verify
[root@localhost mpeg2_16pixel_asm]# vi src/mpeg2dec/idct.c
```

- Assembly Code
  - File : idct.c
  - Line 295

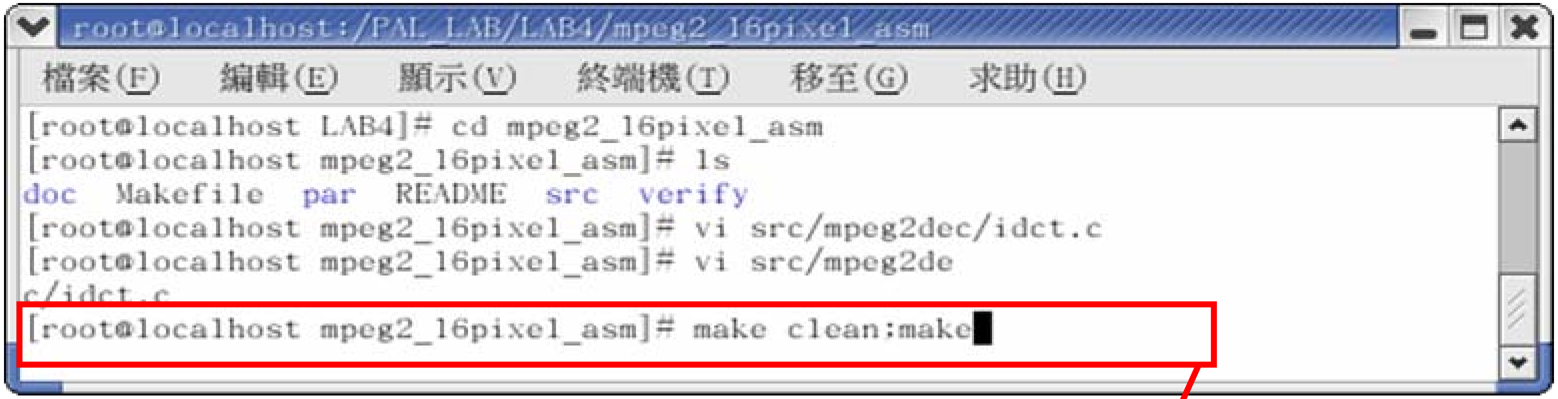
Assembly Code
Line 295



```
 *       c[1..7] = (1/1024)*sqrt(2)
 */
static void idctcol16(blk)
short *blk;
{



  __asm__ ( "STMFD %r13!,{%r0-%r12}\n\t");
    __asm__(
          "mov   %%R0,%0\n\t"
          "mov   %%R1,%1\n\t"
           :
           :"r"(blk),"r"(iclp)
    );

    __asm__(
         "ldrsh   %r2,[%r0,#0]\n\t"           /*x0=
blk[8*0]*/
@
-- 插入 --                          295,41          66%
```

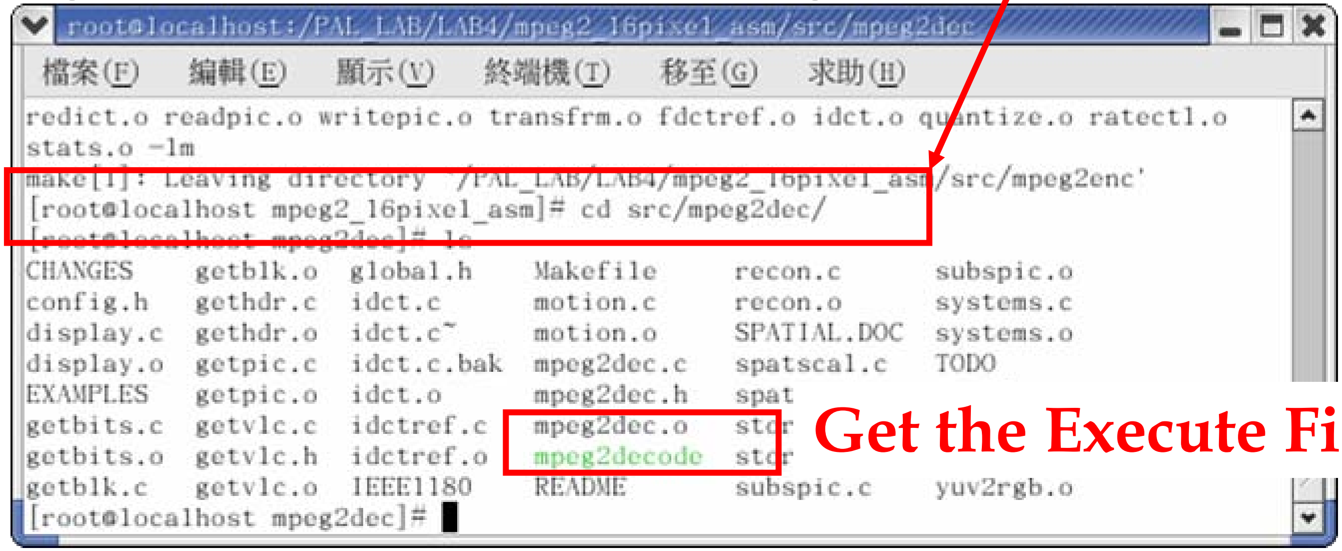
# 4.6 Insert Assembly Code in C Language

- Cross Compile the Source Code
  - #mpeg2_16pixel_asm>make clean;make


root@localhost:/PAL_LAB/LAB4/mpeg2_16pixel_asm
檔案(F) 編輯(E) 顯示(V) 終端機(T) 移至(G) 求助(H)

```
[root@localhost LAB4]# cd mpeg2_16pixel_asm
[root@localhost mpeg2_16pixel_asm]# ls
doc  Makefile  par  README  src  verify
[root@localhost mpeg2_16pixel_asm]# vi src/mpeg2dec/idct.c
[root@localhost mpeg2_16pixel_asm]# vi src/mpeg2de
c/idct.c
[root@localhost mpeg2_16pixel_asm]# make clean:make
```

  - #mpeg2_16pixel_asm>cd src/mpeg2dec/

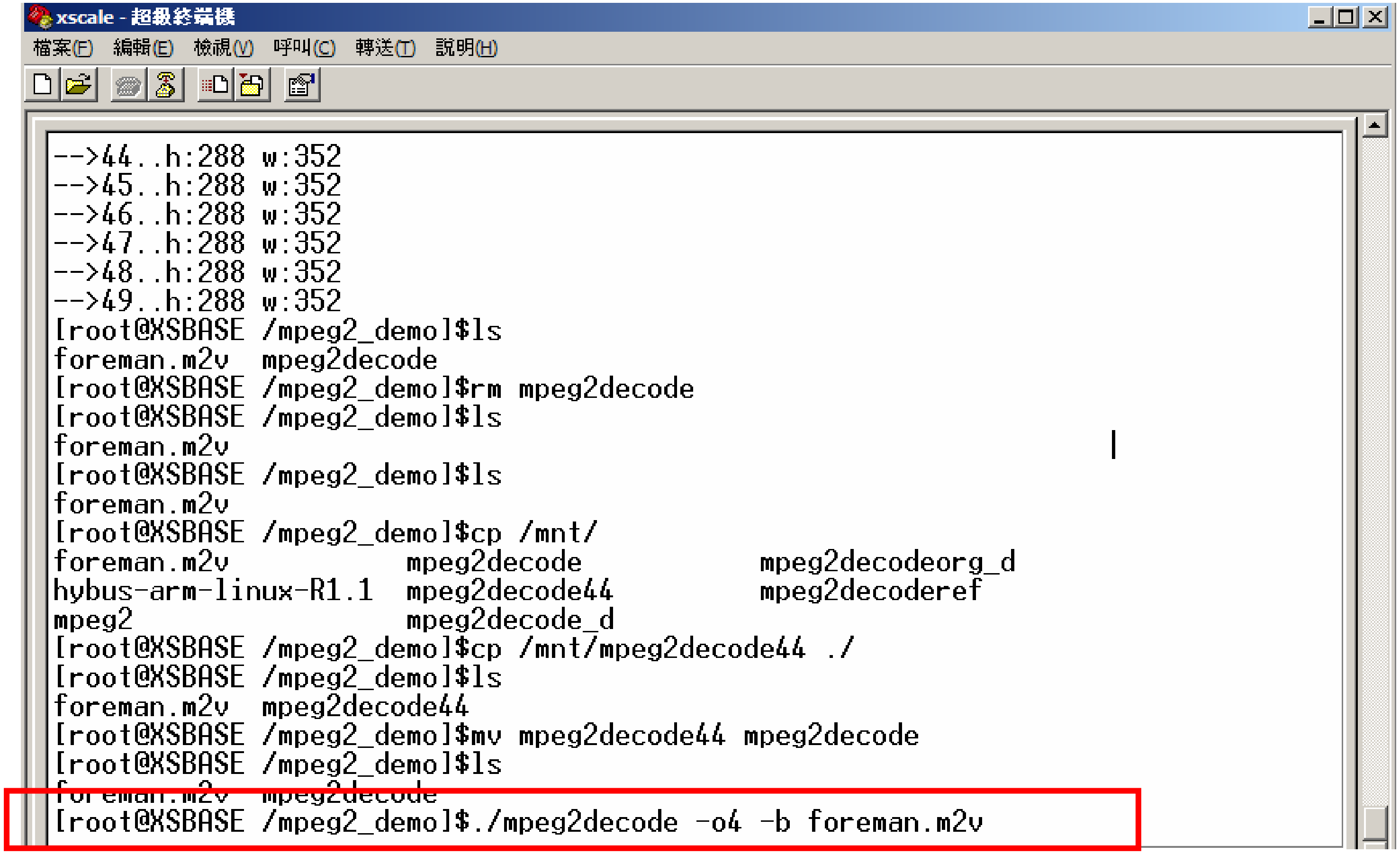

root@localhost:/PAL_LAB/LAB4/mpeg2_16pixel_asm/src/mpeg2dec
檔案(F) 編輯(E) 顯示(V) 終端機(T) 移至(G) 求助(H)

```
redict.o readpic.o writepic.o transfrm.o fdctref.o idct.o quantize.o ratectl.o
stats.o -lm
make[1]: Leaving directory `/PAL_LAB/LAB4/mpeg2_16pixel_asm/src/mpeg2enc'
[root@localhost mpeg2_16pixel_asm]# cd src/mpeg2dec/
[root@localhost mpeg2dec]# ls
CHANGES    getblk.o  global.h    Makefile     recon.c      subspic.o
config.h   gethdr.c  idct.c      motion.c     recon.o      systems.c
display.c  gethdr.o  idct.c~     motion.o     SPATIAL.DOC  systems.o
display.o  getpic.c  idct.c.bak  mpeg2dec.c   spatscal.c   TODO
EXAMPLES   getpic.o  idct.o      mpeg2dec.h   spat
getbits.c  getvlc.c  idctref.c   mpeg2dec.o   stor
getbits.o  getvlc.h  idctref.o   mpeg2decode  stor
getblk.c   getvlc.o  IEEE1180    README       subspic.c    yuv2rgb.o
[root@localhost mpeg2dec]#
```

Get the Execute File

# 4.6 Insert Assembly Code in C Language

- Execute the MPEG-2 Decoder File
  - Execute MPEG-2 decoder
  - #mpeg2_demo>./mpeg2decode -o4 -b foreman.m2v



```
-->44..h:288 w:352
-->45..h:288 w:352
-->46..h:288 w:352
-->47..h:288 w:352
-->48..h:288 w:352
-->49..h:288 w:352
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v  mpeg2decode
[root@XSBASE /mpeg2_demo]$rm mpeg2decode
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v
[root@XSBASE /mpeg2_demo]$cp /mnt/
foreman.m2v           mpeg2decode           mpeg2decodeorg_d
hybus-arm-linux-R1.1  mpeg2decode44         mpeg2decoderef
mpeg2                 mpeg2decode_d
[root@XSBASE /mpeg2_demo]$cp /mnt/mpeg2decode44 ./
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v  mpeg2decode44
[root@XSBASE /mpeg2_demo]$mv mpeg2decode44 mpeg2decode
[root@XSBASE /mpeg2_demo]$ls
foreman.m2v  mpeg2decode
[root@XSBASE /mpeg2_demo]$./mpeg2decode -o4 -b foreman.m2v
```

# 4.6 Insert Assembly Code in C Language

## Performance Report

- The Method to Improve the Performance
  - Assembly Coding
- Total Execute Time
  - Execute Time = 5.69 Seconds
  - Frame Rate = 8.78 fps
    - fps : Frame Per Second

# 4.6 Insert Assembly Code in C Language

- Display on TFT LCD
  - Execute MPEG-2 Decoder