Testing C code with Ultibo Bare Metal, Ultibo TFTP and Ultibo Bitmaps
01/28/17

Goal: This is in hopes of improving the speed of computing the JPEG 2000. The RPi2B or RPi3B will run Ultibo Bare Metal.

To transfer images over an Ethernet connection to a RPi2B or RPi3B.

Perform the JPEG 2000 lifting step which is the first step in the JPEG 2000.

The C code which which performs the Lifting step was develop by

Dan Gisselquist, Ph.D.
Gisselquist Technology, LLC

The C code that performs the DWT Lifting Step runs on x86_64 6 core is considerably faster.

time ./liftmain lena_rgb_512.png

real   0m0.090s
user   0m0.043s
sys   0m0.009s

The C code that performs the DWT Lifting Step runs on the x86_64 dual core and RPi3B is approximately the same.
On x86_64 dual core
time ./liftmain lena_rgb_512.png

real   0m0.356s
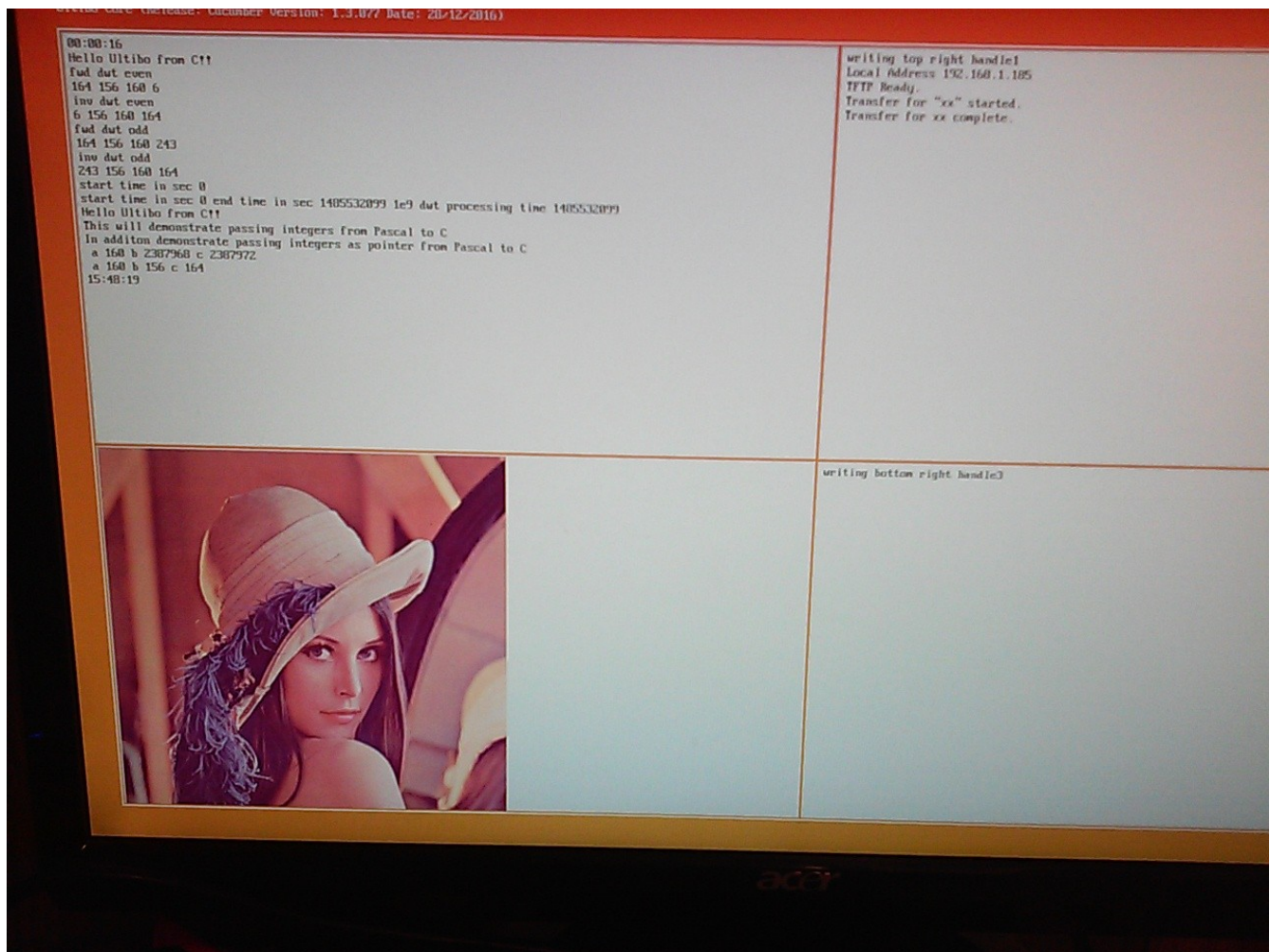user   0m0.209s
sys   0m0.040s

On a RPi3B
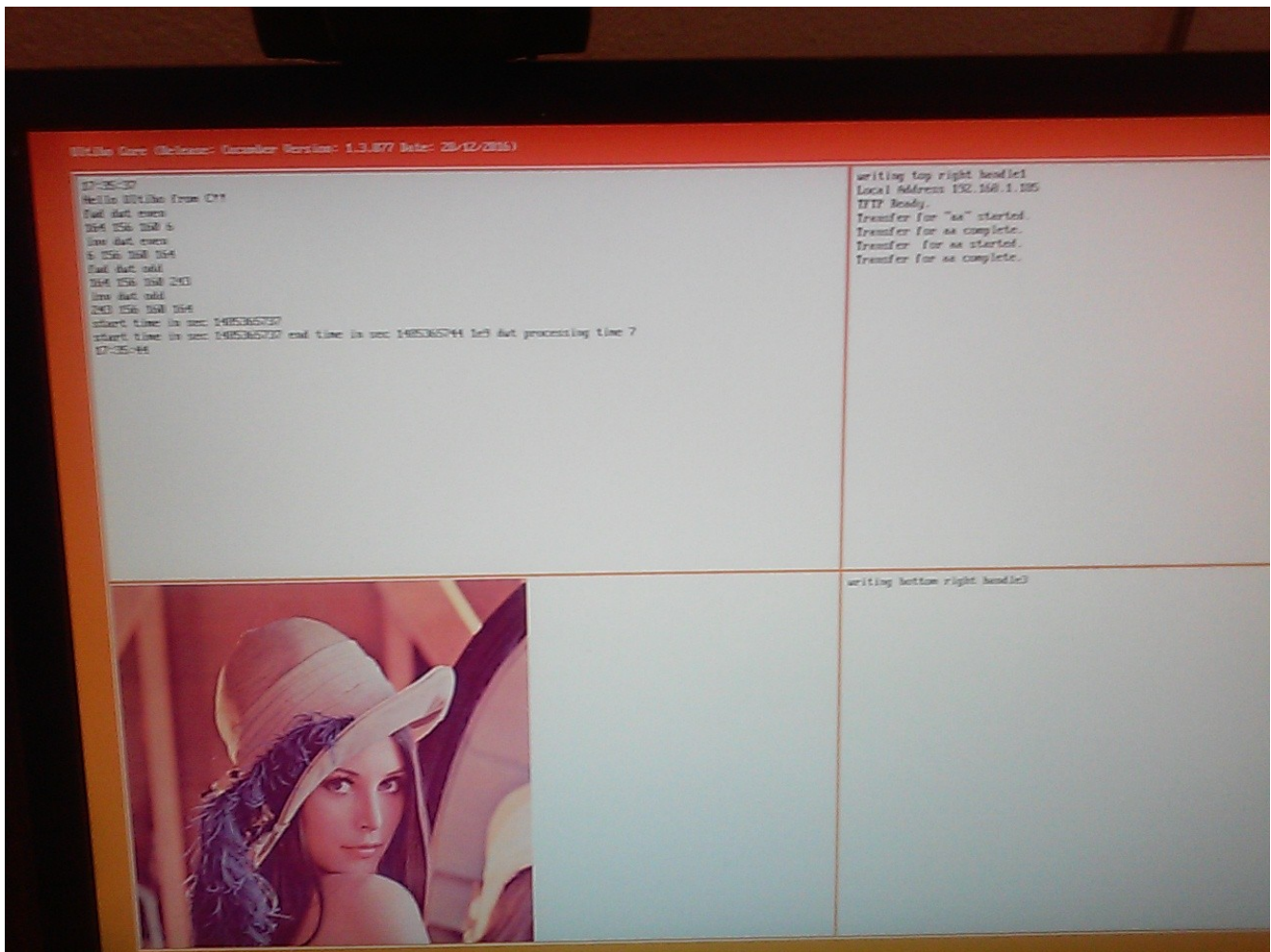./ltime ./liftmain lena_rgb_512.png

real   0m0.380s
user   0m0.230s
sys   0m0.010s

Status: Several Ultibo examples have been merged into a 4 program which displays a bitmap, calls a C library, and provides a TFTP server.

Ultibo Core (Release: Cucumber Version: 1.3.077 Date: 28/12/2016)

00:00:16
Hello Ultibo from C!!
fwd dwt even
164 156 160 6
inv dwt even
6 156 160 164
fwd dwt odd
164 156 160 243
inv dwt odd
243 156 160 164
start time in sec 0
start time in sec 0 end time in sec 1405532099 1e9 dwt processing time 1405532099
Hello Ultibo from C!!
This will demonstrate passing integers from Pascal to C
In additon demonstrate passing integers as pointer from Pascal to C
 a 160 b 2387968 c 2387972
 a 160 b 156 c 164
15:48:19

writing top right handle1
Local Address 192.168.1.105
TFTP Ready.
Transfer for "xx" started.
Transfer for xx complete.

writing bottom right handle3

In the current tests pointers in Pascal are passed to C.

Processing time on Raspbian takes over 10 times more than Ultibo

./epochtime
time sec 1485366551
start time in sec 1485366551 end time in sec 1485366624 1e9 dwt processing time 73

This only takes 7 sec to perform the 1e9 dwt on Ultibo

Topleft is where the C routine is being called. Bottomleft is a 512 x 512 bitmap
In the file test.c the contents of lifing.c
In the topright is the tftp process.
ultibo-tftp

A reasonably quick method of transferring files in an Ultibo project.
It uses Trival FTP based on RFC 1350

Approx upload times around 16 secs for kernel7.img of approx 2.2 MB

https://github.com/pjde/ultibo-tftp.git

tftp 192.168.1.185tftp> binary

tftp> put grn-out.32t aa
Sent 1048576 bytes in 5.0 seconds   1.67MBits/s
tftp> get aa bb
Received 1048580 bytes in 4.1 seconds   2.04MBits/s
tftp> quit

extern void   singlelift(int rb, int w, int * const ibuf, int * const obuf);
extern void   ilift(int rb, int w, int * const ibuf, int * const obuf);
extern void   lifting(int w, int *ibuf, int *tmpbuf);

Updating the kernel7.img Dynamically. First sending the image to a dummy file and retieving the dummy file to a new name. Testing the md5sum of the retrieved file and kernel7.img sent.

Adding the following uses allows a remote system to telnet into a running Ultibo RPi.
 { needed for telnet }
    Shell,
    ShellFilesystem,
    ShellUpdate,
    RemoteShell,
  { needed for telnet }
Transfer a new image to Ultibo RPi.
pi@raspberrypi3:~/jpeg-2000-test/bare-metal/LibC $ tftp 192.168.1.185
tftp> binary
tftp> put kernel7.img tt
Sent 2548944 bytes in 10.8 seconds
tftp> get tt ss
Received 2548944 bytes in 9.2 seconds
tftp> quit
pi@raspberrypi3:~/jpeg-2000-test/bare-metal/LibC $ md5sum kernel7.img ss
df2a3aaf79570eb16f344655fae65357 kernel7.img
df2a3aaf79570eb16f344655fae65357 ss

On a remote system ***"telnet 192.168.1.185"*** where the 192.168.1.185 is the IP of the RPi2B running Ultibo.

```
File  Edit  Tabs  Help
Ultibo Core (Release: Cucumber Version: 1.3.077 Date: 28/12/2016)
 (Type HELP for a list of available commands)
Available commands:
 HELP
 INFO
 VER
 TIME
 CLS
 RESTART
 SHUTDOWN
 UPTIME
 THREAD
 MEMORY
 DEVICE
 FILESYSTEM
 CONTROLLER
 DISK
 PARTITION
 VOLUME
 DRIVE
 CACHE
 DIR
 CD
 MD
 RD
 TYPE
 COPY
 MOVE
 DEL
 REN
 ATTRIB
 TOUCH
 VOL
 LABEL
 DELTREE
 XCOPY
 UPDATE
 LOGOUT
>
```

Need to delete the kernel7.img.
Need to copy the dummy file to the kernel7.img,
Need to restart for the new kernel to start.

telnet 192.168.1.185
Trying 192.168.1.185...
Connected to 192.168.1.185.
Escape character is '^]'.

Ultibo Core (Release: Cucumber Version: 1.3.077 Date: 28/12/2016)
(Type HELP for a list of available commands)
 Directory of C:\

 22-1-17 17:24:24   17932 bootcode.bin
 22-1-17 17:24:24   6621 fixup.dat
 28-1-17 17:33:54   2548944 kernel7.img
 28-1-17 17:33:54   2548944 tt
 22-1-17 17:24:24   786554 MyBitmap.bmp
 22-1-17 17:24:24   2817796 start.elf
 23-1-17 23:41:54   <DIR>   kernelDbitmap
 9-1-17 18:03:38   <DIR>   kerneldemo
 9-1-17 18:05:00   <DIR>   kernelLibCTest
 11-1-17 17:09:20   <DIR>   kernelTFTP
 12-1-17 11:49:44   <DIR>   kernelUDPServer
 24-1-17 15:39:18   4 lena_rgb_512.bmp
 13-1-17 20:03:06   <DIR>   My Files
 22-1-17 17:23:54   <DIR>   old_files
       7 file(s) 8726795 bytes
       7 dir(s)

C:\>del kernel7.img
 Directory of C:\

 22-1-17 17:24:24   17932 bootcode.bin
 22-1-17 17:24:24   6621 fixup.dat
 28-1-17 17:33:54   2548944 tt
 22-1-17 17:24:24   786554 MyBitmap.bmp
 22-1-17 17:24:24   2817796 start.elf
 23-1-17 23:41:54   <DIR>   kernelDbitmap
 9-1-17 18:03:38   <DIR>   kerneldemo
 9-1-17 18:05:00   <DIR>   kernelLibCTest
 11-1-17 17:09:20   <DIR>   kernelTFTP
 12-1-17 11:49:44   <DIR>   kernelUDPServer
 24-1-17 15:39:18   4 lena_rgb_512.bmp
 13-1-17 20:03:06   <DIR>   My Files
 22-1-17 17:23:54   <DIR>   old_files
       6 file(s) 6177851 bytes
       7 dir(s)

C:\ttopy tt kernel7.img
       1 file(s) copied

 Directory of C:\

 22-1-17 17:24:24   17932 bootcode.bin
 22-1-17 17:24:24   6621 fixup.dat
 28-1-17 17:33:54   2548944 tt
 22-1-17 17:24:24   786554 MyBitmap.bmp

```
22-1-17 17:24:24           2817796  start.elf
23-1-17 23:41:54      <DIR>        kernelDbitmap
9-1-17 18:03:38       <DIR>        kerneldemo
9-1-17 18:05:00       <DIR>        kernelLibCTest
11-1-17 17:09:20      <DIR>        kernelTFTP
12-1-17 11:49:44      <DIR>        kernelUDPServer
24-1-17 15:39:18             4  lena_rgb_512.bmp
13-1-17 20:03:06      <DIR>        My Files
22-1-17 17:23:54      <DIR>        old_files
28-1-17 17:33:54           2548944  kernel7.img
        7 file(s) 8726795 bytes
        7 dir(s)

Restarting in 1000 milliseconds
C:\>
```

This is needed to add the fpc compiler to the PATH.

export PATH=/home/pi/ultibo/core/fpc/bin:$PATH
echo $PATH
/
home/pi/ultibo/core/fpc/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/games:/usr/games

arm-none-eabi-gcc -O2 -mabi=aapcs -marm -march=armv7-a -mfpu=vfpv3-d16 -mfloat-abi=hard -c test.c

arm-none-eabi-ar rcs libtest.a test.o

fpc -vi -B -Tultibo -Parm -CpARMV7A -WpRPI2B @/home/pi/ultibo/core/fpc/bin/rpi2.cfg -O2 LibCTestRPi2.lpr

./build_liftmain.sh compiles lifting.c & liftmain.c --> liftmain

iftmain lena_rgb_512.png
        red-out.32t

line 101 lifting.c   const int   LVLS = 1; performs 1 level forward DWT

lines 230-246 in lifting.c when commented does not perform the inverse DWT.
/*
        for(lvl=(LVLS-1); lvl>=0; lvl--) {
                int   offset;

                w <<= 1;

                if (lvl)
                        offset = ov[lvl-1];
                else
                        offset = 0;

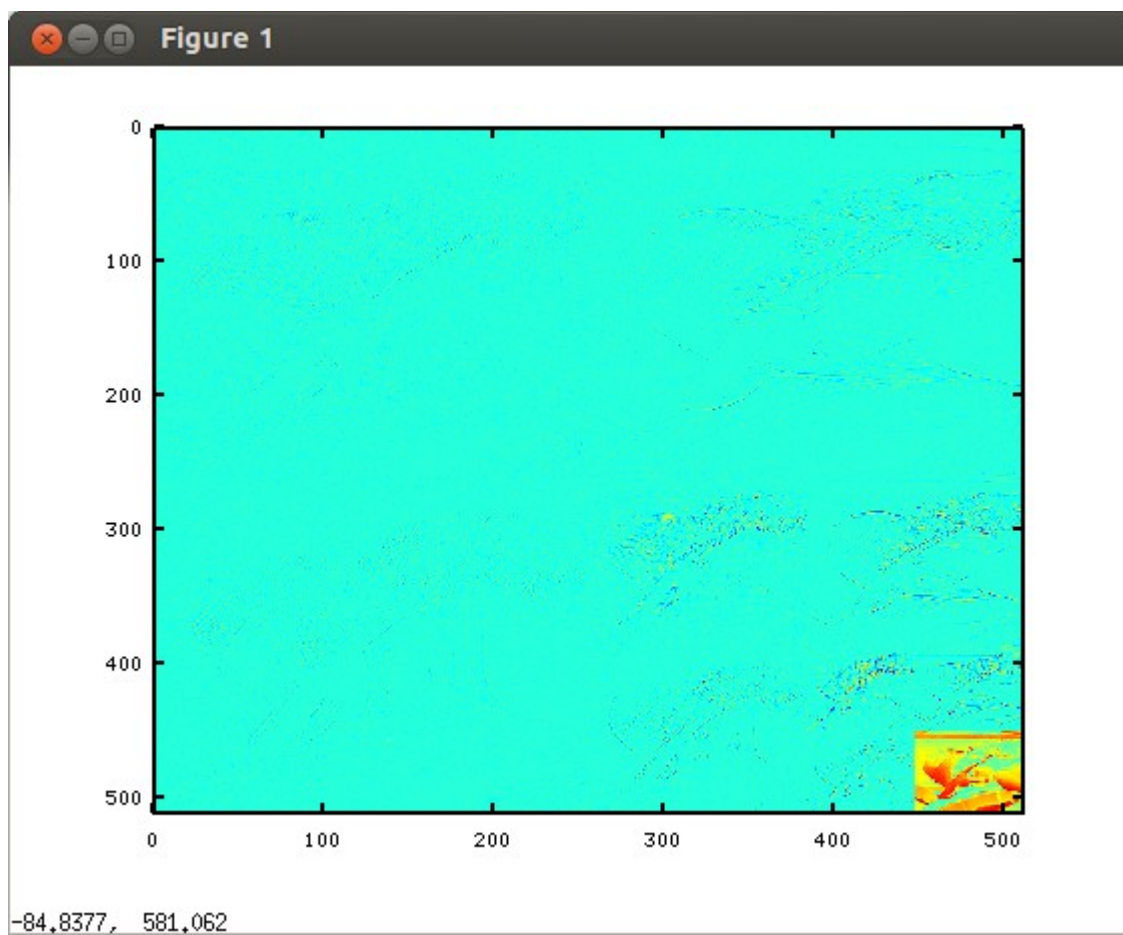```
        ip = &ibuf[offset];
        tp = &tmpbuf[offset];

        ilift(rb, w, ip, tp);
        ilift(rb, w, tp, ip);
    }
*/
```
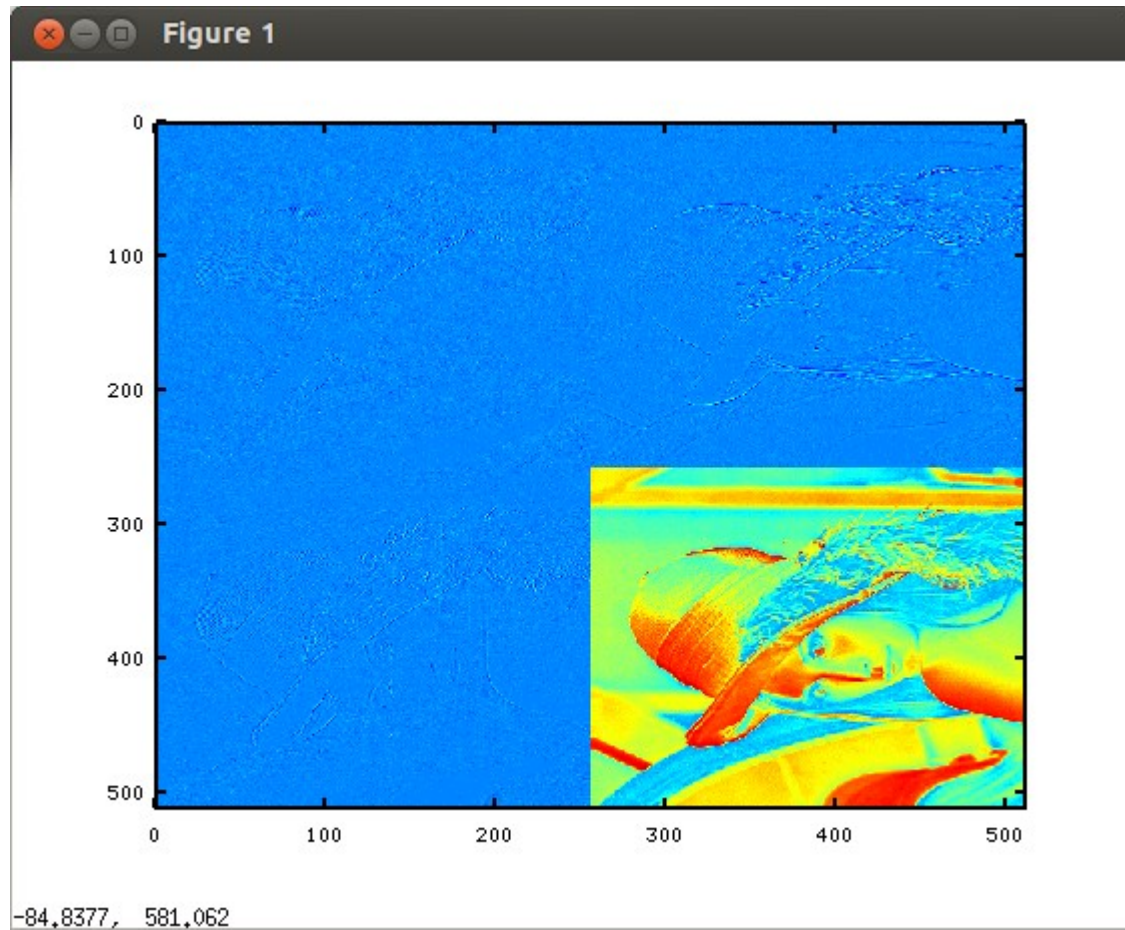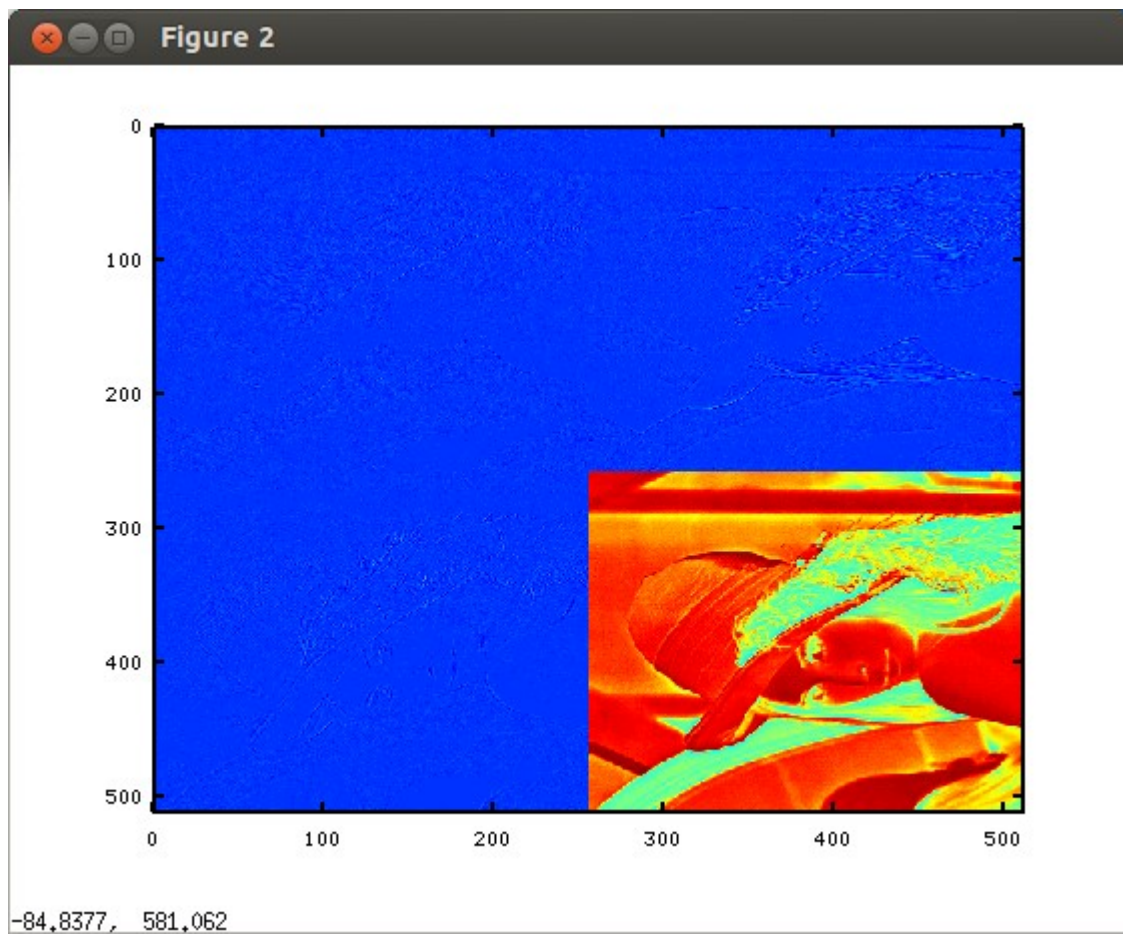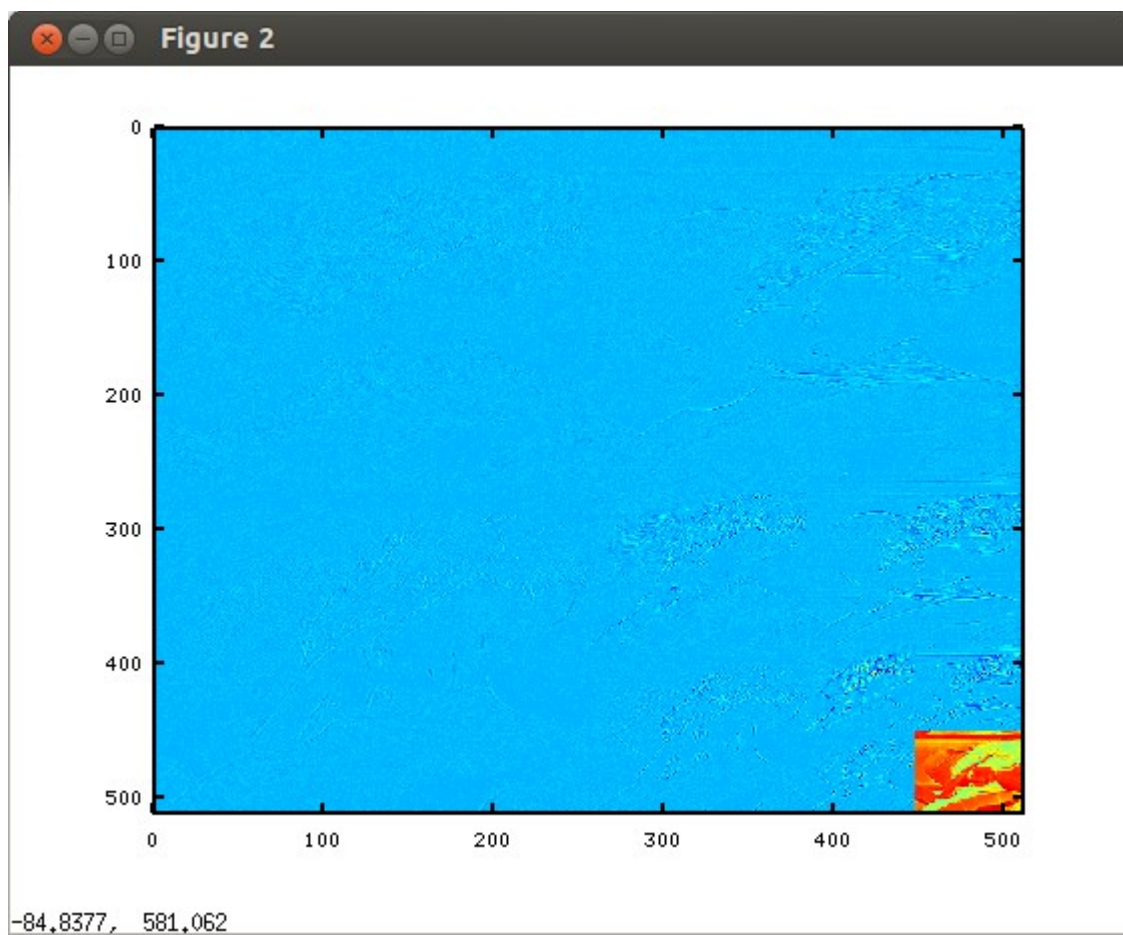


The image above is 1 level forward DWT red subband
The file red-out.32t

The image above is 3 levels forward DWT red subband
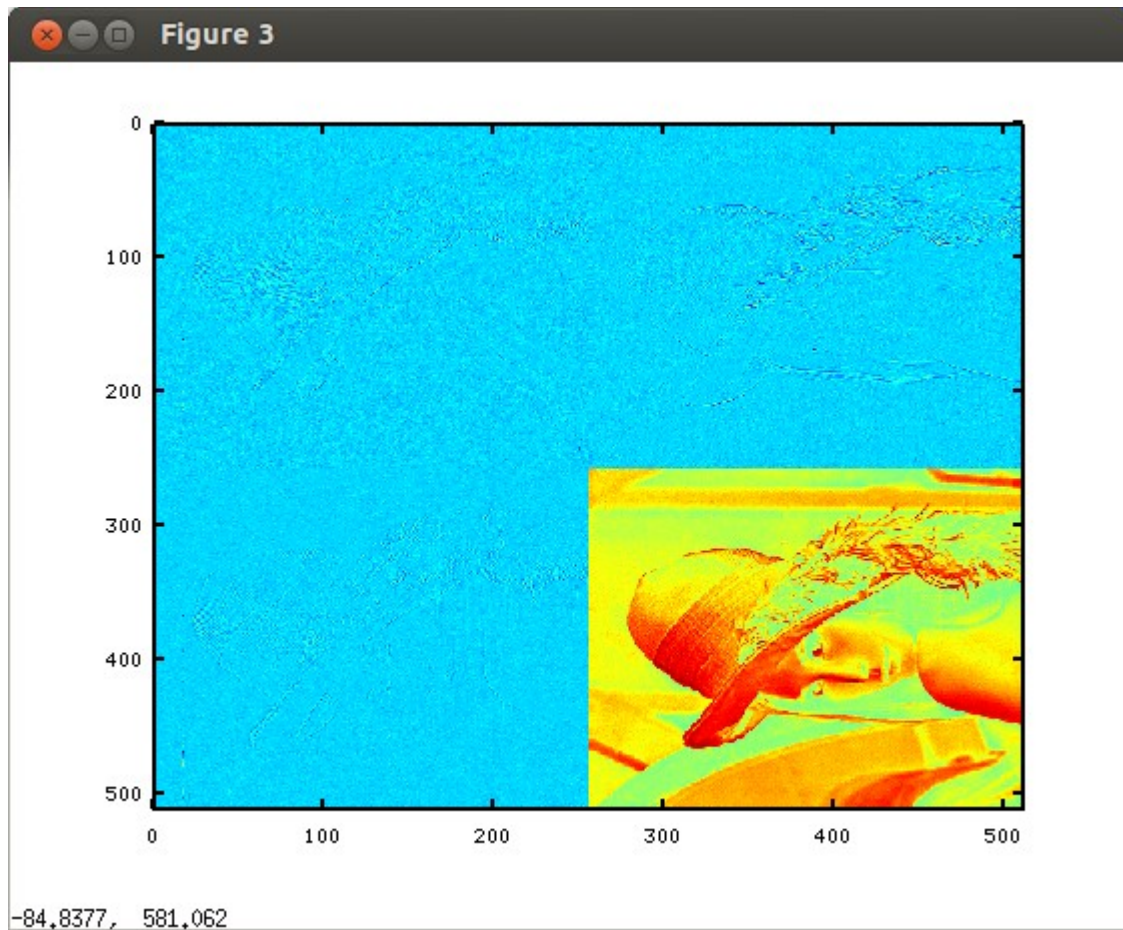The file red-out.32t

The image above is 1 level forward DWT grn subband
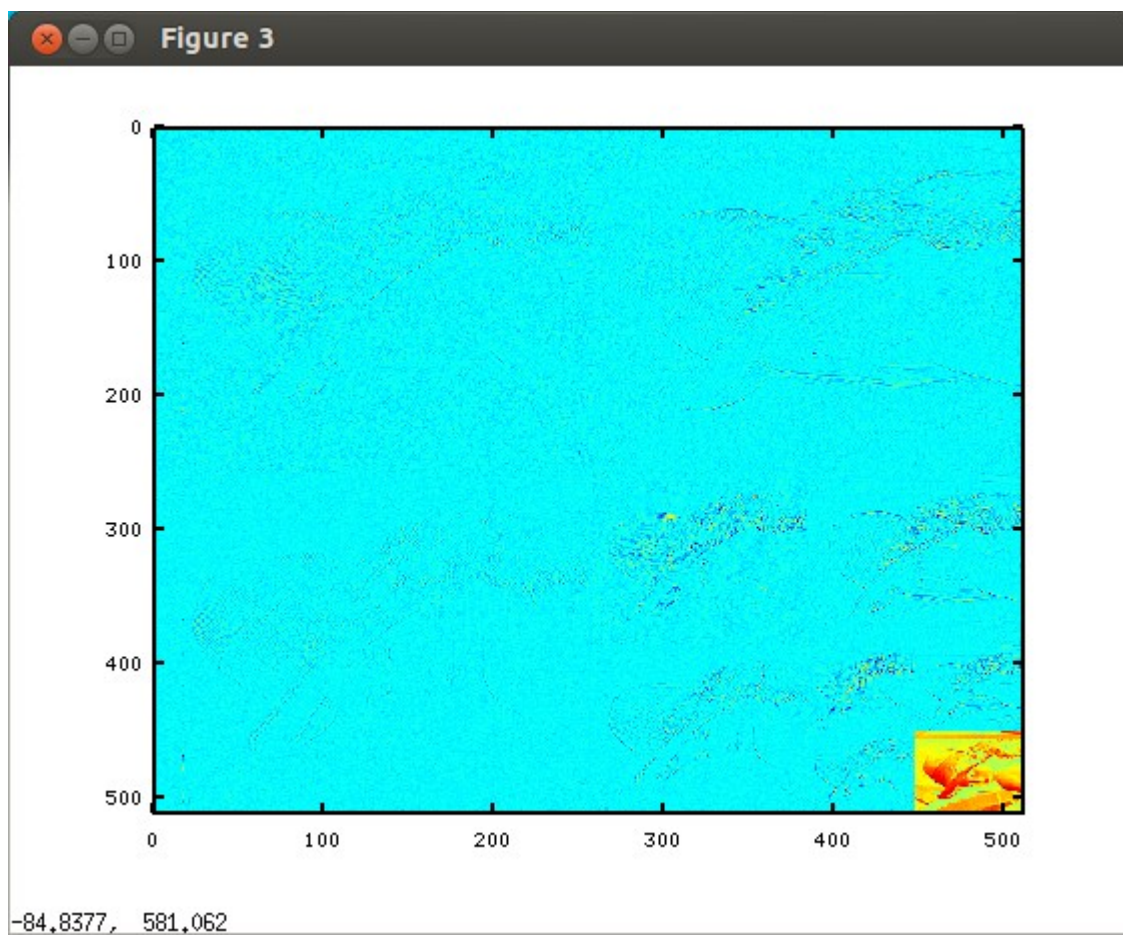
The file grn-out.32t

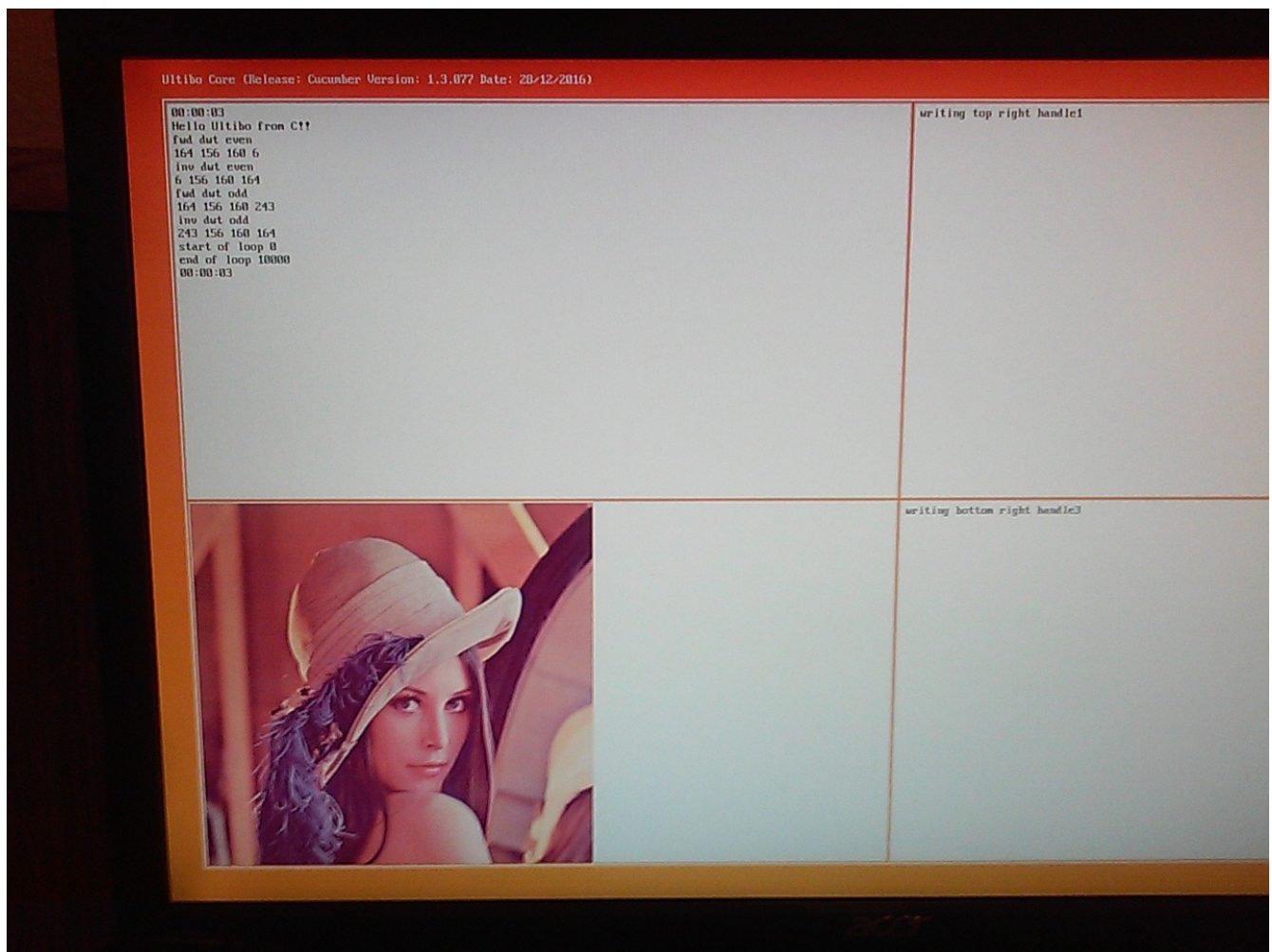The image above is 3 levels forward DWT grn subband
The file grn-out.32t

The image above is 1 level forward DWT blu subband
The file blu-out.32t

The image above is 3 levels forward DWT blu subband
The file blu-out.32t

```
Ultibo Core (Release: Cucumber Version: 1.3.077 Date: 28/12/2016)

00:00:03                                          writing top right handle1
Hello Ultibo from C!!
fwd dut even
164 156 160 6
inv dut even
6 156 160 164
fwd dut odd
164 156 160 243
inv dut odd
243 156 160 164
start of loop 0
end of loop 10000
00:00:03
                                                  writing bottom right handle3
```

The above image is the running on RPi3B as compiled for RPi2B on 01/23/17.