```
/* Code originally taken from the following URL:
     http://svn.arhuaco.org/svn/src/emqbit/tools/emqbit-bench/
*/

/*
 * Authors:
 *    Jorge Victorino
 *    Andres Calderon   andres.calderon@emqbit.com
 *
 * This program is free software; you can redistribute it and/or modify it
 * under the terms of the GNU General Public License as published by the
 * Free Software Foundation; either version 2 of the License, or (at your
 * option) any later version.
 */



#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <time.h>
#if defined(_TMS320C6X)
#elif defined(__GNUC__)
  #include <sys/time.h>
#endif

#include "cfft.h"
#include "common.h"

typedef unsigned long long timestamp_t;

static timestamp_t get_timestamp ()
{
#if defined(_TMS320C6X)
  // There is no gettimeofday in DSP RTS or DSP/BIOS
  return (timestamp_t) clock();
#elif defined(__GNUC__)
  struct timeval now;
  gettimeofday (&now, NULL);
  return  now.tv_usec + (timestamp_t)now.tv_sec * 1000000;
#endif
}

static complex *new_complex_vector(int size);

int main ()
{
  int i;
  int N, n;
  int nTimes;
  float secs;
  timestamp_t t0, t1;

  for (N = (1 << MINPOW2), n = 0; N < (1 << MAXPOW2); N = N << 1, n++)
```

```c
  {
    complex *in = new_complex_vector(N);
    complex *out = new_complex_vector(N);

    fft_init (N);
    // Copy input data and do one FFT
    memcpy (out, in, (N) * sizeof (complex));
    fft_exec (N, out);

    nTimes = ITERATIONS;

    t0 = get_timestamp();

    for (i = 0; i < nTimes; i++)
    {
      memcpy (out, in, (N) * sizeof (complex));
      fft_exec (N, out);
    }

    t1 = get_timestamp();

    secs = (t1 - t0) / 1000000.0L;

    free (in);
    free (out);
    fft_end ();

    fprintf (stderr, "N=%d,nTimes=%d: %g s\n", N, nTimes, secs);
  }

  return 0;
}

static complex *new_complex_vector(int size)
{
  int i;

  complex *new;

  new = (complex *) malloc(sizeof(complex) * size);

  for(i = 0; i < size; ++i)
  {
    new[i].r = (float)rand()/(float)RAND_MAX - 0.5;
    new[i].i = (float)rand()/(float)RAND_MAX - 0.5;
  }

  return new;
}
```

```
/* Code originally taken from the following URL:
     http://svn.arhuaco.org/svn/src/emqbit/tools/emqbit-bench/
*/

/*
 * Authors:
 *    Jorge Victorino
 *    Andres Calderon   andres.calderon@emqbit.com
 *
 * This program is free software; you can redistribute it and/or modify it
 * under the terms of the GNU General Public License as published by the
 * Free Software Foundation; either version 2 of the License, or (at your
 * option) any later version.
 */


#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "cfft.h"
#include "common.h"

complex *tableW;
int *bndx;
int *ndx;

void fft_init (int N)
{
  int i, j;

  tableW = malloc ((N / 2) * sizeof (complex));
  bndx = malloc (N * sizeof (int));
  ndx = malloc ((N / 2) * sizeof (int));

  ndx[0] = 0;
  for (i = 1; i < N / 2; i = i * 2)
  {
    for (j = 0; j < i; j++)
    {
      ndx[j] *= 2;
      ndx[j + i] = ndx[j] + 1;
    }
  }

  bndx[0] = 0;
  for (i = 1; i < N; i = i * 2)
  {
    for (j = 0; j < i; j++)
    {
      bndx[j] *= 2;
      bndx[j + i] = bndx[j] + 1;
    }
```

```c
  }

  for (i = 0; i < N / 2; i++)
  {
    tableW[i].r = cos (ndx[i] * 2.0F * M_PI / (float) N);
    tableW[i].i = -sin (ndx[i] * 2.0F * M_PI / (float) N);
  }
}

void fft_end ()
{
  free (ndx);
  free (bndx);
  free (tableW);
}

void fft_exec (int N, complex * in)
{
  unsigned int n = N;
  unsigned int a, b, i, j, k, r, s;
  complex w, p;

  for (i = 1; i < N; i = i * 2)
  {
    n = n >> 1;
    for (k = 0; k < i; k++)
    {
      w = tableW[k];

      r = 2 * n * k;
      s = n * (1 + 2 * k);

      for (j = 0; j < n; j++)
      {
        a = j + r;
        b = j + s;
        cmult (p, w, in[b]);      //6 flop
        csub (in[b], in[a], p);   //2 flop
        cadd (in[a], in[a], p);   //2 flop
      }
    }
  }
}
```

```
/* Code originally taken from the following URL:
     http://svn.arhuaco.org/svn/src/emqbit/tools/emqbit-bench/
*/

/*
 * Authors:
 *    Jorge Victorino
 *    Andres Calderon   andres.calderon@emqbit.com
 *
 * This program is free software; you can redistribute it and/or modify it
 * under the terms of the GNU General Public License as published by the
 * Free Software Foundation; either version 2 of the License, or (at your
 * option) any later version.
 */

#ifndef COMMON_H
#define COMMON_H

#define MINPOW2 4

#define MAXPOW2 15

#define ITERATIONS 100

#ifndef M_PI
#define M_PI           3.14159265358979323846
#endif

#endif
```

```
/* Code originally taken from the following URL:
     http://svn.arhuaco.org/svn/src/emqbit/tools/emqbit-bench/
*/

/*
 * Authors:
 *    Jorge Victorino
 *    Andres Calderon   andres.calderon@emqbit.com
 *
 * This program is free software; you can redistribute it and/or modify it
 * under the terms of the GNU General Public License as published by the
 * Free Software Foundation; either version 2 of the License, or (at your
 * option) any later version.
 */

#ifndef _CFFT_H_
#define _CFFT_H_

// Prevent C++ name mangling
#ifdef __cplusplus
extern "C" {
#endif

/***********************************************************
* Global Macro Declarations                                *
***********************************************************/

#define pi_2 1.57079632679489661923F

#define abs2(v)  (v.r*v.r + v.i*v.i)

#define angle(v) atan2f(v.i,v.r)

#define cmult(c,a,b) c.r=a.r*b.r - a.i*b.i, \
                     c.i=a.r*b.i + a.i*b.r

#define csub(c,a,b)  c.r=a.r - b.r, \
                     c.i=a.i - b.i

#define cadd(c,a,b)  c.r=a.r + b.r, \
                     c.i=a.i + b.i


/***********************************************************
* Global Typedef Declarations                              *
***********************************************************/

typedef struct {
    float r;
    float i;
} complex;


/***********************************************************
* Global Variable Declarations                             *
***********************************************************/



/***********************************************************
* Global Function Declarations                             *
***********************************************************/

extern void fft_init();
extern void fft_end();
extern void fft_exec(int N, complex* in);



/***********************************************************
* End file                                                 *
***********************************************************/

#ifdef __cplusplus
}
#endif

#endif //_CFFT_H_
```

```
#############################################################################
# Makefile                                                                  #
#                                                                           #
# Builds the emqbit benchmark source for ARM and DSP                        #
#############################################################################
#
#
#############################################################################
#                                                                           #
#   Copyright (C) 2010 Texas Instruments Incorporated                       #
#     http://www.ti.com/                                                    #
#                                                                           #
#############################################################################
#
#
#############################################################################
#                                                                           #
#  Redistribution and use in source and binary forms, with or without       #
#  modification, are permitted provided that the following conditions      #
#  are met:                                                                 #
#                                                                           #
#    Redistributions of source code must retain the above copyright         #
#    notice, this list of conditions and the following disclaimer.          #
#                                                                           #
#    Redistributions in binary form must reproduce the above copyright      #
#    notice, this list of conditions and the following disclaimer in the    #
#    documentation and/or other materials provided with the                 #
#    distribution.                                                          #
#                                                                           #
#    Neither the name of Texas Instruments Incorporated nor the names of    #
#    its contributors may be used to endorse or promote products derived    #
#    from this software without specific prior written permission.          #
#                                                                           #
#  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS      #
#  "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT        #
#  LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR    #
#  A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT     #
#  OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,    #
#  SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT         #
#  LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,    #
#  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY    #
#  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT      #
#  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE    #
#  OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.     #
#                                                                           #
#############################################################################

#   ----------------------------------------------------------------------------
#   Name of the ARM GCC cross compiler & archiver
#   ----------------------------------------------------------------------------
ARM_TOOLCHAIN_PREFIX  ?= arm-none-linux-gnueabi-
ifdef ARM_TOOLCHAIN_PATH
ARM_CC := $(ARM_TOOLCHAIN_PATH)/bin/$(ARM_TOOLCHAIN_PREFIX)gcc
ARM_AR := $(ARM_TOOLCHAIN_PATH)/bin/$(ARM_TOOLCHAIN_PREFIX)ar
```

```
else
ARM_CC := $(ARM_TOOLCHAIN_PREFIX)gcc
ARM_AR := $(ARM_CROSS_COMPILE)ar
endif

# Get any compiler flags from the environmentre
ARM_CFLAGS = $(CFLAGS)
ARM_CFLAGS += -std=gnu99 \
-Wdeclaration-after-statement -Wall -Wno-trigraphs \
-fno-strict-aliasing -fno-common -fno-omit-frame-pointer \
-c -O3
ARM_LDFLAGS = $(LDFLAGS)
ARM_LDFLAGS+=-lm -lpthread
ARM_ARFLAGS = rcs

#   ----------------------------------------------------------------------------
#   Name of the DSP C6RUN compiler & archiver
#   TI C6RunLib Frontend (if path variable provided, use it, otherwise assume
#   the tools are in the path)
#   ----------------------------------------------------------------------------
C6RUN_TOOLCHAIN_PREFIX=c6runlib-
ifdef C6RUN_TOOLCHAIN_PATH
C6RUN_CC := $(C6RUN_TOOLCHAIN_PATH)/bin/$(C6RUN_TOOLCHAIN_PREFIX)cc
C6RUN_AR := $(C6RUN_TOOLCHAIN_PATH)/bin/$(C6RUN_TOOLCHAIN_PREFIX)ar
else
C6RUN_CC := $(C6RUN_TOOLCHAIN_PREFIX)cc
C6RUN_AR := $(C6RUN_TOOLCHAIN_PREFIX)ar
endif

C6RUN_CFLAGS = -c -O3
C6RUN_ARFLAGS = rcs --C6Run:replace_malloc


#   ----------------------------------------------------------------------------
#   List of source files
#   ----------------------------------------------------------------------------
EXEC_SRCS := main_cfft.c main_bench.c
EXEC_ARM_OBJS := $(EXEC_SRCS:%.c=gpp/%.o)
EXEC_DSP_OBJS := $(EXEC_SRCS:%.c=dsp/%.o)

LIB_SRCS := cfft.c distance.c
LIB_ARM_OBJS := $(LIB_SRCS:%.c=gpp_lib/%.o)
LIB_DSP_OBJS := $(LIB_SRCS:%.c=dsp_lib/%.o)

#   ----------------------------------------------------------------------------
#   Makefile targets
#   ----------------------------------------------------------------------------
.PHONY : dsp_exec gpp_exec dsp_lib gpp_lib dsp_clean gpp_clean all clean

all: dsp_exec gpp_exec
clean: gpp_clean dsp_clean


gpp_exec: gpp/.created gpp_lib $(EXEC_ARM_OBJS)
```

```
    $(ARM_CC) $(ARM_LDFLAGS) $(CINCLUDES) -o bench_arm gpp/main_bench.o bench_arm.lib
    $(ARM_CC) $(ARM_LDFLAGS) $(CINCLUDES) -o cfft_arm gpp/main_cfft.o cfft_arm.lib

gpp_lib: gpp_lib/.created $(LIB_ARM_OBJS)
    $(ARM_AR) $(ARM_ARFLAGS) bench_arm.lib gpp_lib/distance.o
    $(ARM_AR) $(ARM_ARFLAGS) cfft_arm.lib gpp_lib/cfft.o

gpp/%.o : %.c
    $(ARM_CC) $(ARM_CFLAGS) $(CINCLUDES) -o $@ $<

gpp_lib/%.o : %.c
    $(ARM_CC) $(ARM_CFLAGS) $(CINCLUDES) -o $@ $<

gpp/.created:
    @mkdir -p gpp
    @touch gpp/.created

gpp_lib/.created:
    @mkdir -p gpp_lib
    @touch gpp_lib/.created

gpp_clean:
    @rm -Rf bench_arm cfft_arm bench_arm.lib cfft_arm.lib
    @rm -Rf gpp gpp_lib


dsp_exec: dsp/.created dsp_lib $(EXEC_DSP_OBJS)
    $(ARM_CC) $(ARM_LDFLAGS) $(CINCLUDES) -o bench_dsp dsp/main_bench.o bench_dsp.lib
    $(ARM_CC) $(ARM_LDFLAGS) $(CINCLUDES) -o cfft_dsp dsp/main_cfft.o cfft_dsp.lib

dsp_lib: dsp_lib/.created $(LIB_DSP_OBJS)
    $(C6RUN_AR) $(C6RUN_ARFLAGS) bench_dsp.lib dsp_lib/distance.o
    $(C6RUN_AR) $(C6RUN_ARFLAGS) cfft_dsp.lib dsp_lib/cfft.o

dsp/%.o : %.c
    $(ARM_CC) $(ARM_CFLAGS) $(CINCLUDES) -o $@ $<

dsp_lib/%.o : %.c
    $(C6RUN_CC) $(C6RUN_CFLAGS) $(CINCLUDES) -o $@ $<

dsp/.created:
    @mkdir -p dsp
    @touch dsp/.created

dsp_lib/.created:
    @mkdir -p dsp_lib
    @touch dsp_lib/.created

dsp_clean:
    @rm -Rf bench_dsp cfft_dsp bench_dsp.lib cfft_dsp.lib
    @rm -Rf dsp dsp_lib

install:
    scp bench_arm bench_dsp cfft_arm cfft_dsp root@beagle4:c6run_target/examples/c6runlib/emqbit
```