Technical Notes on using Analog Devices' DSP components and development tools
Phone: (800) ANALOG-D, FAX: (781) 461-3010, EMAIL: dsp.support@analog.com, FTP: ftp.analog.com, WEB: www.analog.com/dsp

Initializing Variables in C using the 21xx C compiler

Last modified 9/2/98

Overview

This Engineer's Note will address the procedure of initializing variables in C using the 21xx C compiler. Example FFT code is provided at the end of this EE note.

For many applications using the 21xx family of DSPs, data is gathered by using one of the many IO interfaces on the DSP. However, there may be an occasion where an engineer may wish to initialize a variable with a data array in the internal memory of the DSP. This is not difficult to do using assembly code. However, this procedure is a little trickier for the engineer who is programming a 21xx DSPs using C code.

To initialize variables in C for the 21xx family of DSPs, it is necessary to define arrays in an external assembly module. Additionally, these arrays should be declared as ".global" in the assembly code. Then, these arrays may be called in your C code, by declaring them as "extern." Then, to initialize these variables, use a ".init" statement to initialize the arrays in the assembly module. For more details on the ".init" statement, please read page 3-33 of the Assembler Tools and Simulator Manual and read the example code in this EE note.

The example code on the next page performs a fft and an inverse fft upon two data arrays that are stored in data memory. These arrays, called real_input and imag_input, are initialized with a data file that contains 16 bit hex values in 1.15 data format. They are stored in the files named real.hex and imag.hex. The format of these files is demonstrated below:

Real.hex	Imag.hex	
0000	0000	
080a	0000	
100b	0000	
.	.	
.	.	252 more values
.	.	
d872	0000	

The arrays imag_output and real_output are the results of the fft function, which then are fed into the inverse fft function. The outputs of the inverse fft function, rout and iout, are nearly identical to the inputs, real_input and imag_input. Discrepancies are due to the limited resolution of the 21xx DSP family.

This code was compiled using VDSP, version 6.0, with the following command lines:

```
g21 cifft.c  -a adsp2181.ach -runhdr 2181_hdr.dsp -g -save-temps -c

asm21 var_init.dsp -c -cp -2181

ld21 cifft var_init 2181_hdr -a adsp2181 -e cifft -gcc -lib -x -g
```

ANALOG DEVICES

References:

ADSP-2100 Family User's Manual, Third Edition

ADSP-2100 Family C Runtime Library Manual,
Second Edition

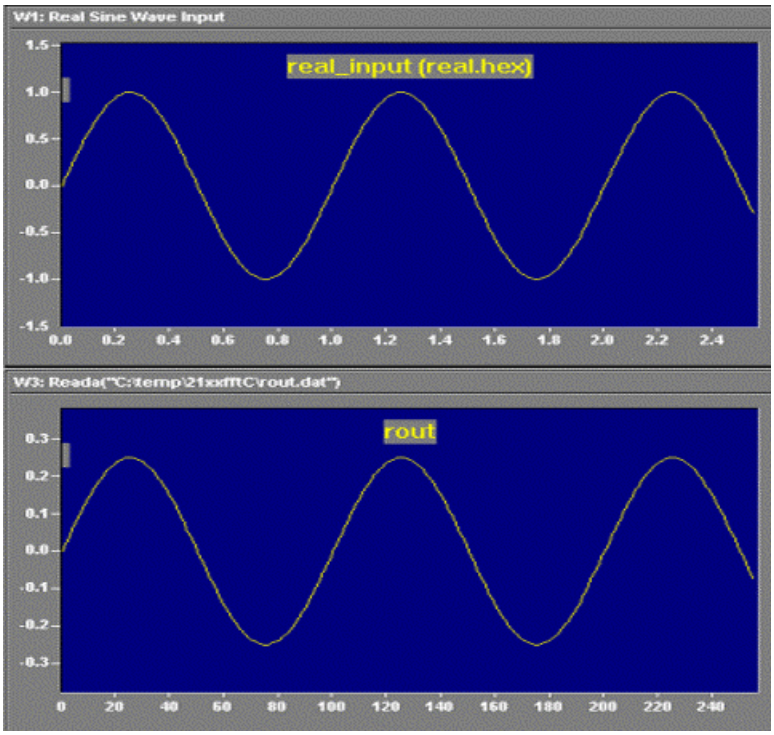
ADSP-2100 Family C Tools Manual, Second Edition

Listing fft_ifft.c

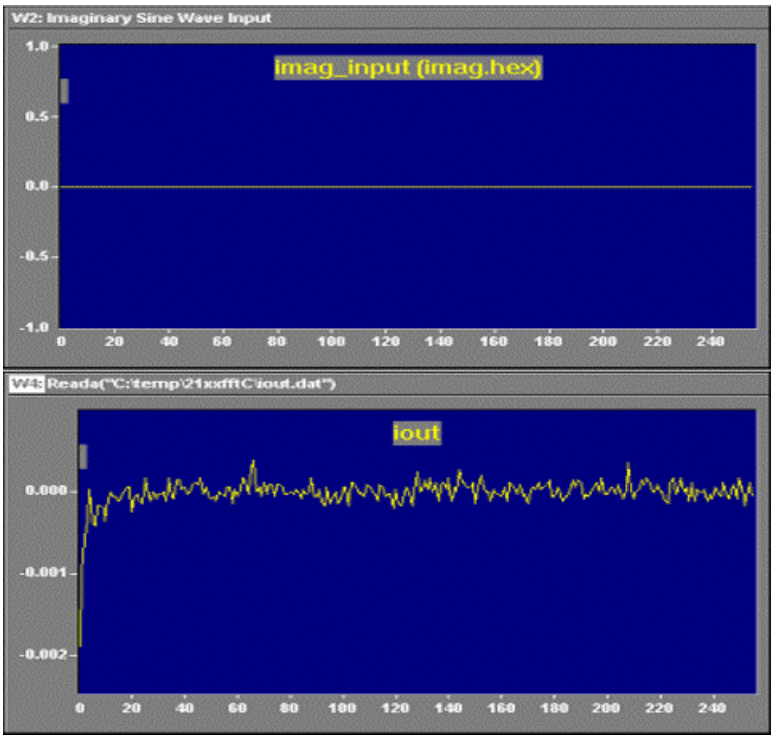
```
/* The code performs a fft and inverse fft upon an array of 256 data points which are stored in the variable s real_input and
imag_input.  These arrays are declared and initialized in the file array_init.dsp.  The output of this program, rout and iout, is
nearly identical to the inputs, real_input and imag_input.  Discrepancies are due to the limited resolution of the 21xx DSP family.
In this test case, a regular sine wave input was used for real_input and imag_input. */

#include <ffts.h>                                   /*include C callable fft functions */
#define N 256                                       /*number of data points, can be configured   */
                                                    /*valid values for N can be found in the C runtime */
                                                    /*library manual under fft or ifft.       */

extern int real_input[N], imag_input[N];            /* these variables are declared in array_init.dsp */
                                                    /* and describe the real and imaginary components */
                                                    /* of the input signal, a sine wave                */
int imag_output[N], real_output[N];                 /* these values are the result of the fft */
                                                    /* which are in turn passed to the inverse fft */
int rout[N], iout[N];                               /* results of the inverse fft */

main()
{
int i;
fft256(real_input, imag_input, real_output, imag_output);          /* call fft program */
ifft256(real_output, imag_output, rout, iout);                  /* call inverse fft program */
asm("idle;");                                                      /* after done, idle */
}
```

Listing, array_init.dsp

```
.module array_init;

.var/dm  real_input_[256];          /* declares array of 256 points */
.global   real_input_;              /* makes this array accessable by fft_ifft.c */
.INIT     real_input_: <real.hex>;  /* initializes the array with real part of a sine wave */

.var/dm  imag_input_[256];          /* declares array of 256 points */
.global   imag_input_;              /* makes this array accessable by fft_ifft.c */
.INIT     imag_input_: <imag.hex>;  /* initializes the array with imaginary part of a sine wave */

.endmod;
```



Real input and output from fft_ifft.c



Imaginary input and output from fft_ifft.c