

## Using the 21xx C-FFT library

*Last Modified: 08/12/98*

### Introduction:

One of the applications used very often on a DSP, is the Fast Fourier Transforms (FFT). A block of data is moved from the time domain to the frequency domain. This note shall assist the programmer to obtain the right results through correct buffer placement, when using the 16 bit C FFT libraries.

### Theory of programming.

There are several C library functions(assembly source code is also provided) shipped with Analog Devices 16-bit Development Tools computing the FFT of an array of data, whose size is a power of 2. Using the C-Compiler, will take care of proper twiddle tables, modifier values and result placement. To use the library shipped with the current C-Compiler, the user has just to add `#include <ffts.h>` to the main file header for calling the FFT library. FFT functions in this library are provided for array sizes of 8 to 1024 points.

The function prototype for the complex FFT is `fftN(r_inp, i_inp, r_outp, I_outp)` with N indicating the number of elements in the buffers. As the library is optimized for the ADSP-21xx processors, it expects a specific placement of the data buffers, to allow bit reversing for input data scrambling. The bit reversing requires a buffer start address to meet certain restrictions i.e. buffers with 16 elements may start on 0x0, 16, 32 This corresponds to the placement rules for circular buffers, so the linker utility ld21 has to be used, the qualifier CIRC.

The C-Compiler does not account for this issue, so the programmer has to care for proper placement by special definitions of these buffers. Currently this could be handled by a macro, or directly using inline assembler. The instruction to define a circular buffer is `.var/dm/ram/circ buffer_[length]`. This will allow the linker to place the data correctly Bit reversing will be

possible and the input data can be taken properly from the real input buffer `r_inp` and the imaginary input buffer `i_inp`, so the computed results will be placed in `r_outp` and `i_outp`.

### Software Tools Caveats.

The twiddle table coefficients necessary for calculating the FFT are stored in the standard C library and initialized as 16bit integer. As these must be fit into program memory (PM) it is necessary that the linker does properly understand the `-gcc` switch for correct data placement. This switch must be added, too, when manually linking different source module, otherwise improper data placement for PM will be found. If your linker ld21 shows lower version number than 2.18, an update can be obtained from [ftp.analog.com](http://ftp.analog.com).

### FFT application code example.

```
#include <ffts.h>
#define C 8

asm(".var/ram/dm/circ   Ireal_[8];      ");
asm(".var/ram/dm/circ   Iimag_[8];      ");
asm(".var/ram/dm/circ   Oreal_[8];      ");
asm(".var/ram/dm/circ   Oimag_[8];      ");
asm(".global Ireal_, Iimag_, Oreal_, Oimag_");

extern int Ireal[], Iimag[], Orealr[], Oimag[];
int k;

void main(void)
{
    for (k = 0; k < C; k++) Iimag[k] = 0;

    Ireal[0] = 1;
    Ireal[1] = 2;
    Ireal[2] = 3;
    Ireal[3] = 4;
    Ireal[4] = 5;
    Ireal[5] = 6;
    Ireal[6] = 7;
    Ireal[7] = 8;

    fft8(Ireal, Iimag, Oreal, Oimag);

    asm("idle;");
}
```

### References:

- 1) ADSP-2100 Family C- Tools Manual
- 2) ADSP-2100 Family C- Runtime Library
- 3) ADSP-2100 Assembler Tools & Simulator Manual
- 4) [ftp.analog.com/pub/dsp/dev\\_tool/21xx\\_tool](http://ftp.analog.com/pub/dsp/dev_tool/21xx_tool)