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

Using the 21xx C-FFT library

Last modified 03/05/98

Introduction

One of the very often used applications running on DSPs are fast Fourier Transforms (FFT) to move a block of data from the time domain to the frequency domain. This note shall assist the programmer to obtain the right results through correct buffer placement.

Theory of programming

From Analog Devices are several FFT source modules available, computing the input data stored in a data array of a power of 2. The easiest way of programming such an application is using the C-Compiler, as this will care for proper twiddle tables, modifier values and results 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. The user will find FFT functions in this library 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 start address where a buffer of a power of 2 may be placed. 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 advised to do so using the qualifier CIRC.

Unluckily 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 advise the linker for correct data placement. 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 our ftp site.

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;
```

ANALOG DEVICES

```
       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