Clarification of Multi-Pass FFT usage

Environment & Usage

1. CCSv5.3, C6713 simulator
2. Use attached sample project to recreate the issue.
3. Select “EXPERIMENT” definition at the top of main.c. The available presets for 2048 points FFT experiments looks like below:

```
//Enable single pass FFT with the tone created SinGenerator (1000Hz + 8345Hz sizewave @ 48khz)
// ==> You will see the correct result in result_array_real/img buffer
#define EXPERIMENT_0 0

//Enable multi pass FFT with the tone created SinGenerator (1000Hz + 8345Hz sizewave @ 48khz), incremental process at "for"
// ==> You will see the correct result in result_array_real/img buffer
#define EXPERIMENT_1 1

//Enable multi pass FFT with the tone created SinGenerator (1000Hz + 8345Hz sizewave @ 48khz), decremental process at "for"
// ==> You will see the correct result in result_array_real/img buffer
#define EXPERIMENT_1b 2

//Enable multi pass FFT with the tone created sin()/cos() math function, incremental process at "for"
// ==> You will see the wrong result in result_array_real/img buffer
#define EXPERIMENT_2 3

//Enable multi pass FFT with the tone created sin()/cos() math function, decremental process at "for"
// ==> You will see the wrong result in result_array_real/img buffer, but it looks like better than EXPERIMENT_2. In this case, we saw "real" part in result_array_img and "img" part in result_a
#define EXPERIMENT_3 4
```

Current status/understandings

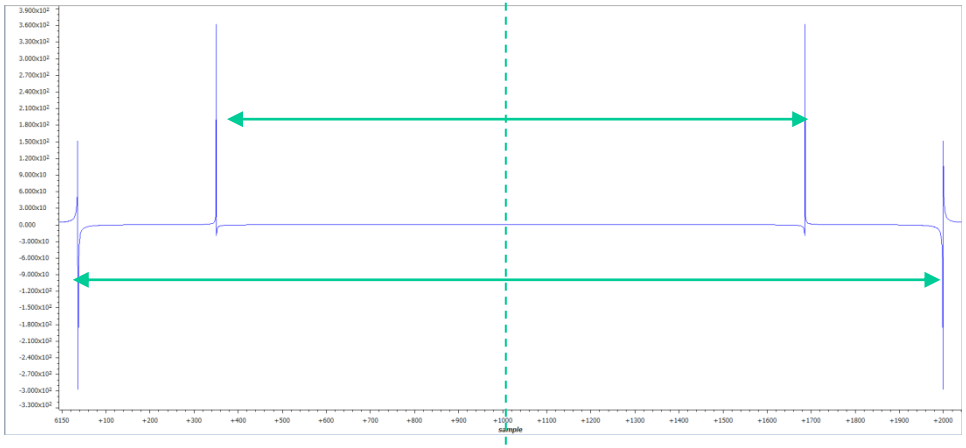
1. According to the answers posted before, “decrement the counter at for loop” is correct implementation for multi-pass FFT.

Verifications

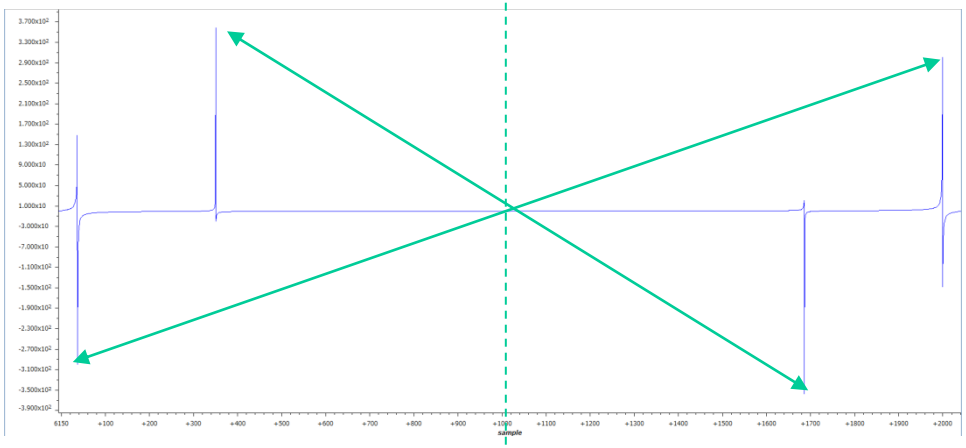
1. Lets me focus on EXPERIMENT_0, EXPERIMENT_1 and EXPERIMENT_1b.
2. In these case, the input for FFT is sine wave (1000Hz + 8345Hz @48khz FS), and FFT implementation is different each other:
    1. EXPERIMENT_0 => single-pass FFT
    2. EXPERIMENT_1 => multi-pass FFT (incremental operation at for loop)
    3. EXPERIMENT_1b => multi-pass FFT (decremental operation at for loop)
3. We should see the same FFT result for each case. The next slide show the results of each experiment.

Result : EXPERIMENT_0 (single-pass FFT)

result_array_real[] => even-function base on 1024 sample index



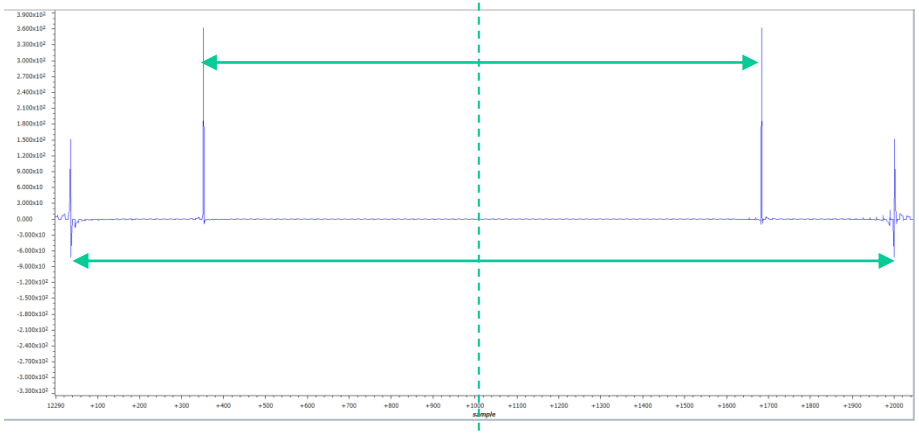
result_array_img[] =>odd-function base on 1024 sample index



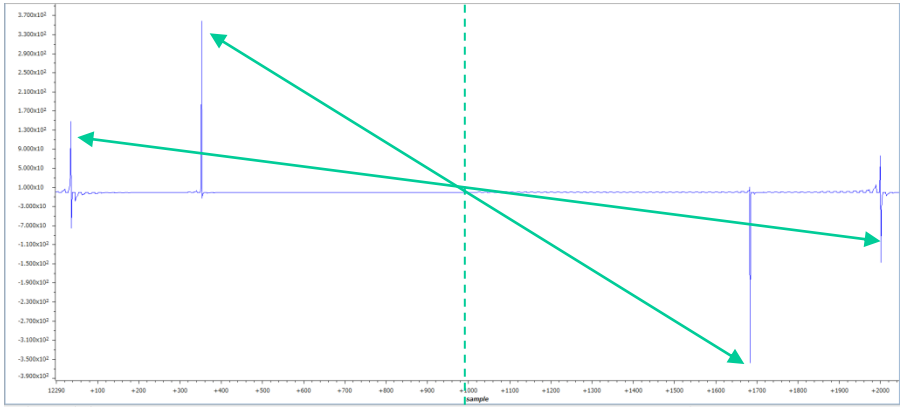
Result looks correct!

Result : EXPERIMENT_1 (multi-pass FFT, incremental operation at for loop)

result_array_real[] => even-function base on 1024 sample index



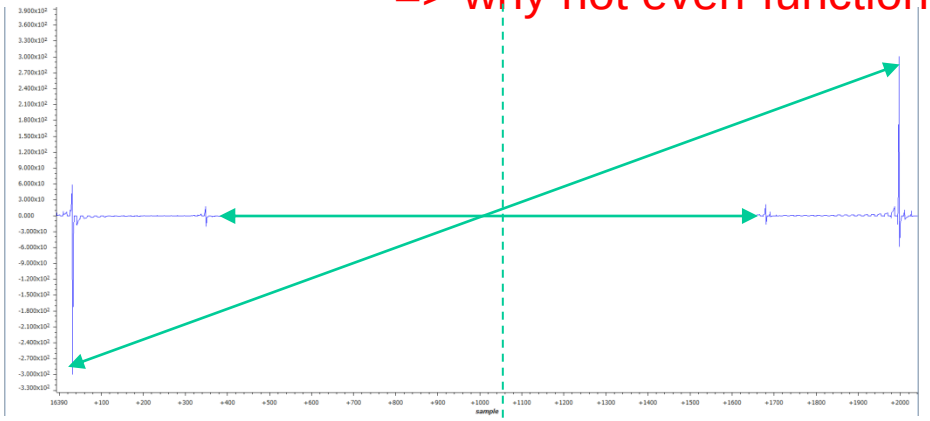
result_array_img[] => odd-function base on 1024 sample index



Result looks correct!
But it does not match to the
TI's answer. This experiment is
using the implementation of
"incremental operation at for loop"
and this should be invalid implementation!
Why can I get the correct output ?

Result : EXPERIMENT_1b (multi-pass FFT, decremental operation at for loop)

result_array_real[] => odd-function base on 1024 sample index
=> why not even-function ?



result_array_img[] => even-function base on 1024 sample index
=> why not odd-function ?

Result looks wrong!
And it does not match to the TI's answer. This experiment is using the implementation of "decremental operation at for loop" and this should be valid implementation!
Why can I get the incorrect output ?