HILEVEL Technology, Inc.

Q'n Apps #S49

Q: How do I test a PLL?

Phase Locked Loop

Use of the MX feature of the Titan and Griffin systems is one way to measure frequency. There is a ñtrickò that uses the digital part of the tester to accomplish this. Consider the following equation:

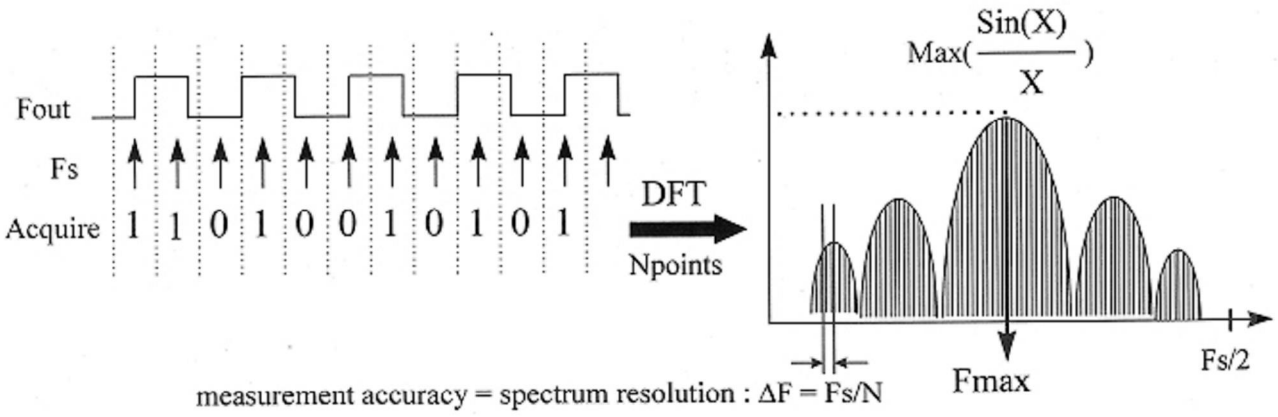
$$Fout = Fmax \pm Fs/N$$

Example: Fout known at 1%?
N = 100 points is enough.

Follow these basic steps to setup for frequency measurement. The illustration below helps to clarify the process.

1) Connect the PLL output to a tester system channel. Define this channel as DUT output with some nominal threshold level.
2) In the vectors window, define the expected output value as all L. Now when you click the RUN button, the Analysis window will show a number of Hôs followed by number of Lôs, followed by number of Hôs, then a number of Lôs, and so on. When you display the Analysis data as a Vector Graph you will see a square waveform coming from the PLL output.
3) Now for the tricky part: think of the system channel the PLL is connected to as a single bit digitizer, operating at the Test Rate frequency, named Fs in the drawing below. When you will pass trace data for this pin to the Fast Fourier Transform function, you will receive the frequency spectrum data. Search for the MAX value. This is the searched PLL frequency. You can also derive jitter (statistically) based on this frequency spectrum data.
4) What if the PLL freq is more than the Test Rate? Connect the PLL to 2, 3, or more sys channels and for each of them define a different compare strobe to have it distributed equally within time, thus effectively multiplying Fs.

The ACT/Asp C code for automating this will use the functions appearing on the next page.



ACT/Asp C code functions for performing FFT frequency measurement:

AspFunFunctionalTest() - collect output data
AspReadTraceMemory() - transfer data into PC (or better AspReadSinglePebTraceMemory() - faster)
AspDecodeMemoryBit()-to read single channel data
AspFFTComplex() - convert into frequency domain
Eventually - int AspFFTComplexToMag() - get frequency magnitude data
Search the FFT result for max value - this is Fpll.

For detailed function descriptions, please see ACT Help.

See the next page for a C code example of a frequency measure program using the above ACT/Asp functions.

```
#define NOOFBITS          (1024) /* number of samples to collect, will define frequency resolution and accuracy */
#define FOUT              (12)  /* frequency output system channel number */
#define TEST_RATE  (10000000.0) /* functional test rate in Hz, here 10MHz*/

double ReadFrequency(long address)
{
   int j, bit;
   long trace[NOOFBITS][MAX_PES];//where to collect read memory
   long veclines[NOOFBITS];//helper table, formal parameter not used here
   ANPVALUE anresult[NOOFBITS];//read data to be used for frequency calculation
   long fftbincount, maxbinno = 0;
   double magnitude = 0.0, frequency = 0.0;
   double *FFTRetFreq, *FFTReResultY, *FFTImResultY, *FFTResultMag,*FFTResultPhase;//variables used for FFT functions

   //collect trace
   AspReadTraceMemory ( address,NOOFBITS,trace, veclines);
   //decode digital code
   for(j=0;j<NOOFBITS;j++)
   {
            bit = AspDecodeMemoryBit ( FOUT,trace[j] );
            anresult[j].vector_number = j;// required for timing calculation
            anresult[j].voltage = bit;//here we set captured data
            anresult[j].phase = 0;//not used - set to 0
            anresult[j].flags = 0;//not used - set to 0
   }
   //perforom FFT
   fftbincount = AspFFTComplex(anresult,NOOFBITS, TEST_RATE , &FFTRetFreq, &FFTReResultY,&FFTImResultY);
   AspFFTComplexToMag(FFTReResultY,FFTImResultY,fftbincount,&FFTResultMag,&FFTResultPhase);

   //search for maximum amplutude - fundamental frequency
   for(j=0;j<fftbincount;j++)
   {
            if(FFTResultMag[j] > magnitude)
                     maxbinno = j;
   }
   frequency = FFTRetFreq[maxbinno];// return fundamental frequency
   //free memory allocated in AspFFT... functions
   free(FFTRetFreq);
   free(FFTReResultY);
   free(FFTImResultY);
   free(FFTResultMag);
   free(FFTResultPhase);
   return frequency;
}
```

C code example for frequency measurement