Verifying the clock rate and the clock function of C66XX cores

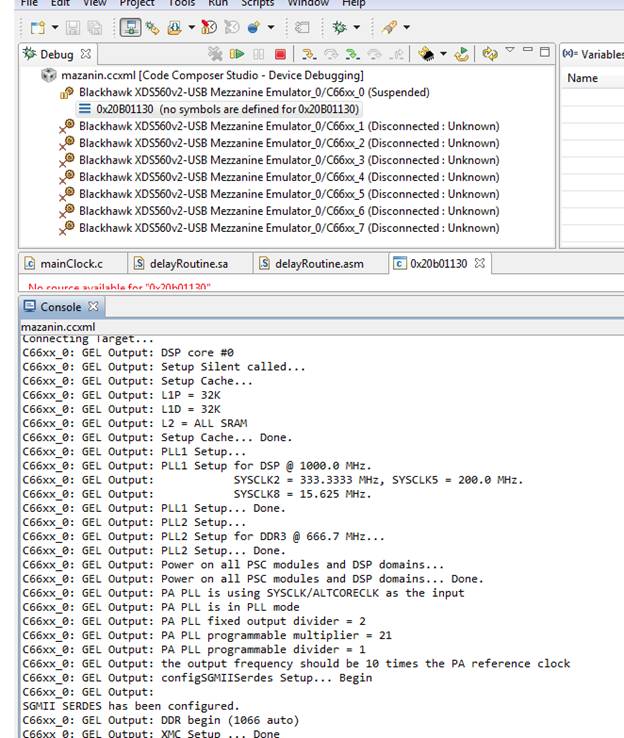
I was asked two questions – does the core runs at 1G, and is the clock accurate.  I will try to answer both questions.

First I develop a project to test the timing.  The project (attached) has two functions. A main function that I compile with full debug and no optimization, so I can step through the code and enable the clock and do whatever I want, and a second file which is linear assembly file that does N nops, where N is a parameter.  This file I compile with full optimization no debug.

Now I attach the assembly output code – delayRoutine.asm and you can see clearly that the function takes one cycle per nop.  There are 31 or 32 cycles that are associated with overhead  calling the routine, but other than this, the cycles’ consumption of the routine is fixed.

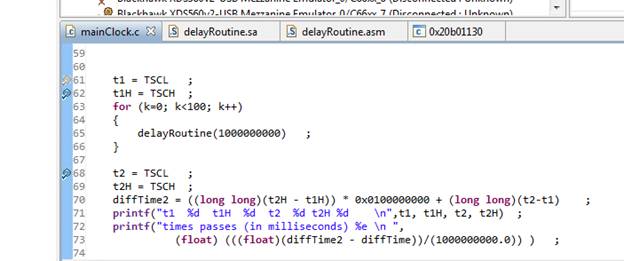
So I step through the main, enable the clock in CCS and change the parameter to the delayRoutine function, and I verify that for the parameter N the function consumes N+31 cycles.  So I change the parameter to 1000000000, In that case the function takes 1G cycles (plus 31) so about a second.

Next I connect the EVM and connect the target. The gel file configures the PLL and this is what I see:



I Use the standard gel file from the release.   **If you do not see the DSP set up for 1000 MHZ there is a problem.  The PLL initialization determines the DSP speed. It should be 1G or 1.25G or 0.8G depends on the setting.**

Next I run the code.  As you see I have a loop that I run as many times as I want. I change the loop to 100 and put a break point before and after the loop. The function looks as follows. – Do you see the two break point, on line 62 and line 68?



Them I did the following, I run to break point 62, and at the same time hit resume run and the start button on the stopwatch of my smart phone. It supposed to run 100 times, each time for 1G cycles or a second.

When the code reached line 68 I stop the stopwatch.  What I see is 1 minutes and 41 seconds, or 101 seconds. The extra second is due to my delay in click the stopwatch after noticing the code reach break point 68**.  So 1G cycles runs for 1 second AS IT SHOULD.**

Again, the project is attached. Please repeat the experiment.