Vivado Frequency Meter

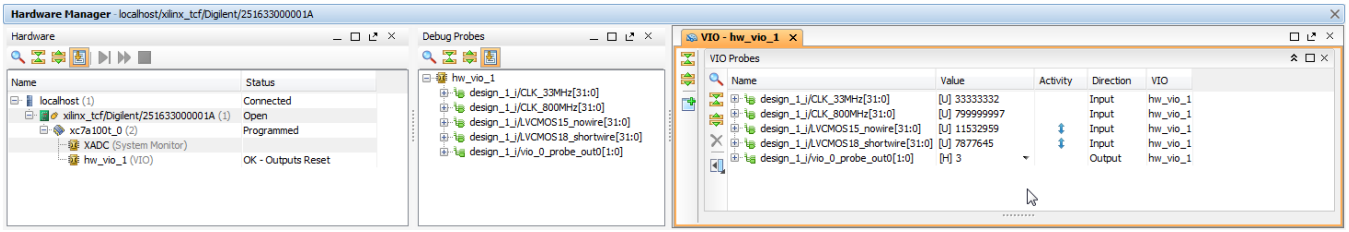
Frequency meter? So simple you may say. Yes, sure. But doing FPGA based frequency meter that can be instantly used by thousands of engineers? Hm.. once long time ago I did a JTAG controlled FPGA Frequency meter IP Core and PC GUI application for it also. I think I wanted to have some pictures and links to Xilinx and yes I did get at least semi-official "green" from them to-do so. The GUI application talked over LPT Based JTAG adapter and did some clever tricks to measure the frequency on some FPGA pin without the use of any known reference clock in the FPGA. As part of that work I did get from semi-official channels schematic of original Xilinx Parallel Cable IV. In order to support it I did create a Coolrunner JEDEC to VHDL conversion tool. I also had a ideas about integration with Chipscope IP Cores, so I did some reverse engineering on the Chipscope ICON.

This was long time ago.

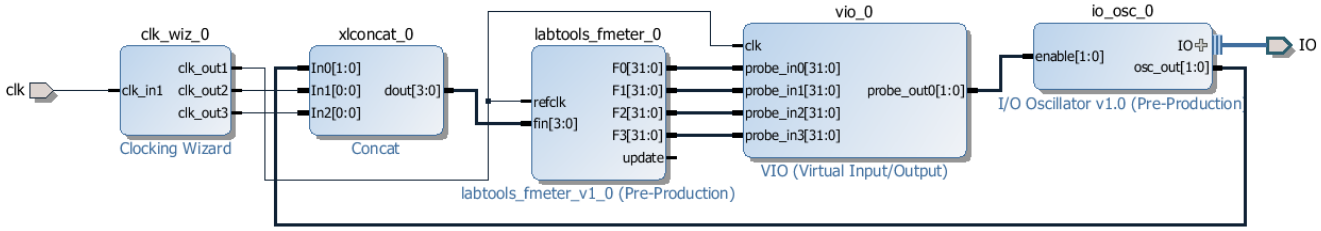
LPT Ports are long past history, so bringing back in live that old design makes no sense. Maybe the FPGA based simple Frequency counter is already done? There are options of course, Altium has on-chip instrumentation that includes Frequency meter (if I recall it correctly). But Altium FPGA support is something that is maybe fun for educational purposes not much usable for real life.

Vivado Labtools do not have Frequency meter.

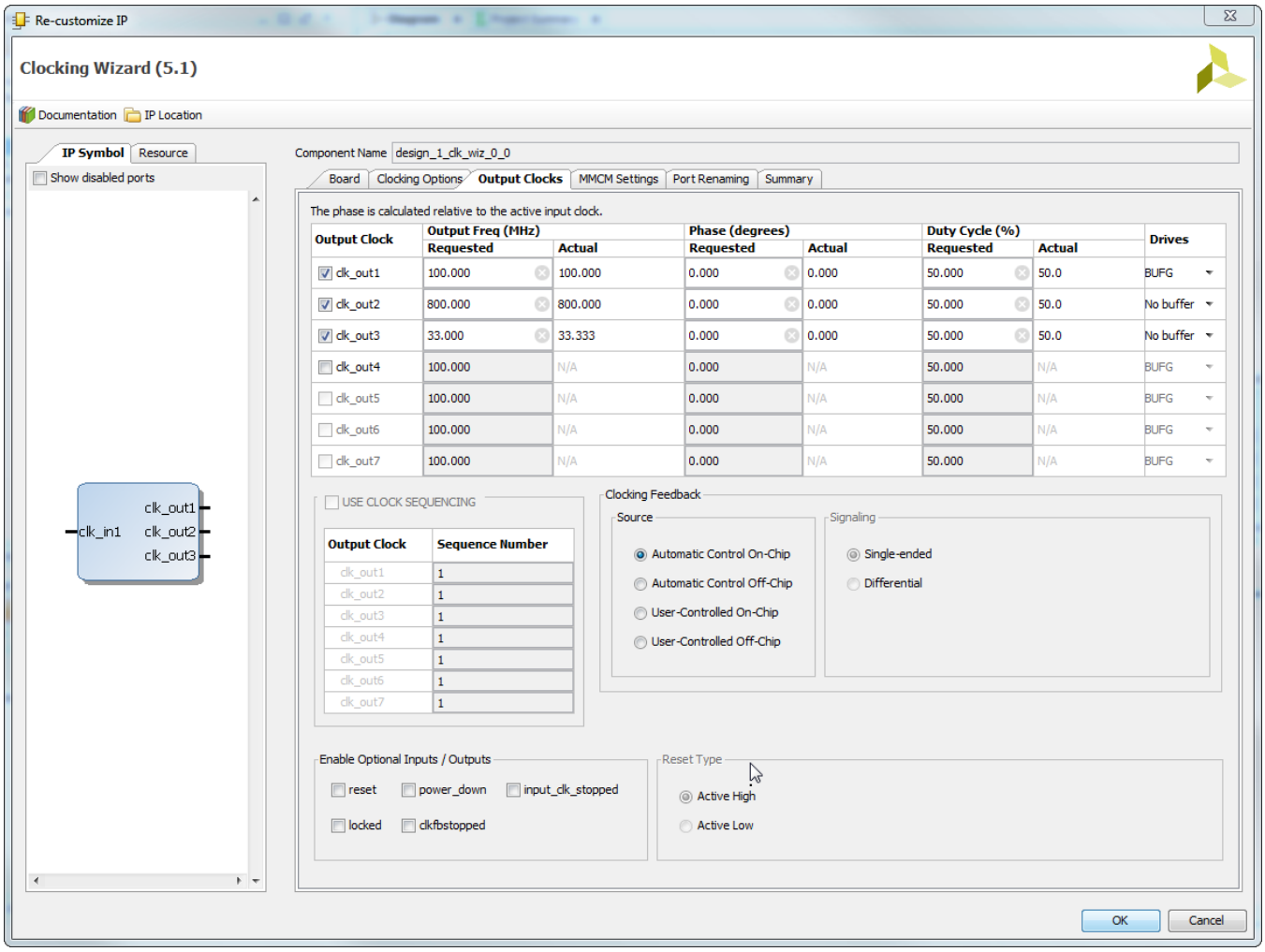
Who said that? They do now!



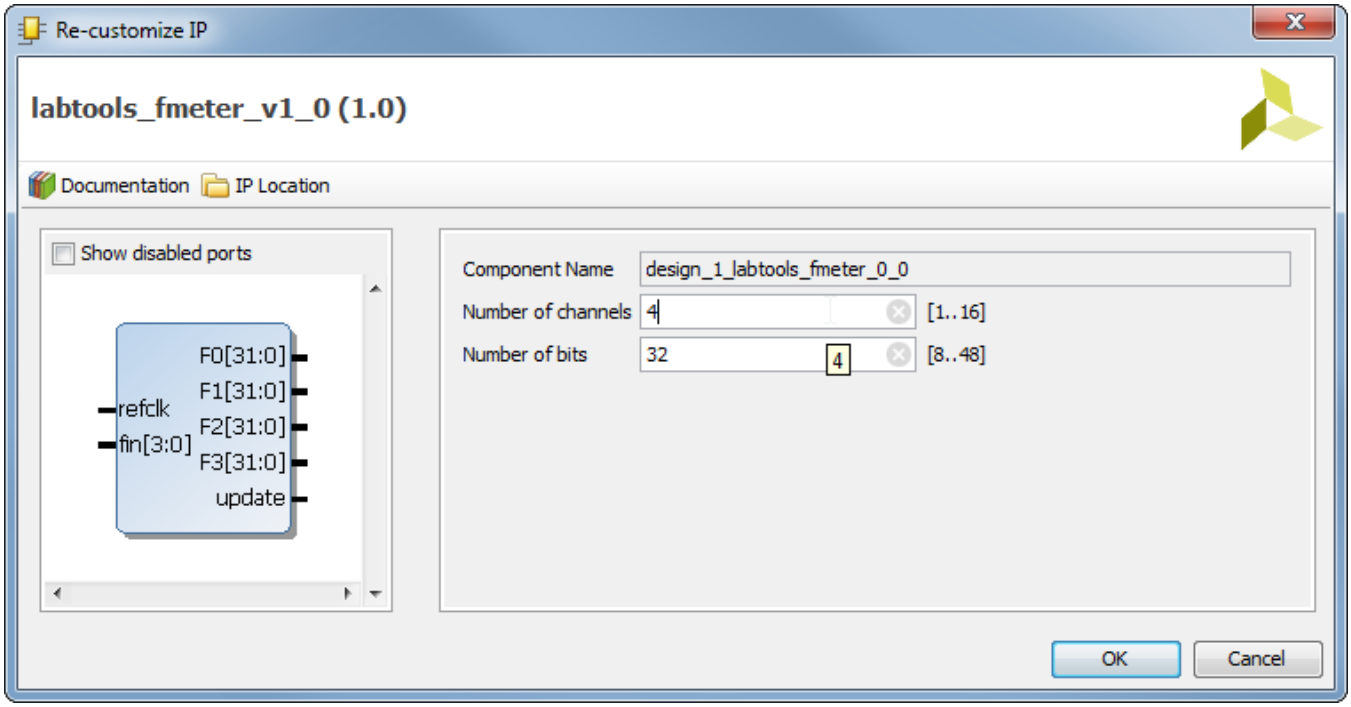
Here, TE0710 Art ix module as hardware test bench. Frequency Counter IP Core configured for 4 channels. Reference clock is from 100 MHz oscillator. Just for testing 33.3333 MHz and 800MHz signals from FPGA Clock PLL are connected to channels 0 and 1. Channels 2 and 3 are connected to I/O Oscillator IP Core. Channel 2 I/O pin is "empty" FPGA ball from DDR3 bank with 1.5V I/O Voltage. Channel 3 I/O pin is in 1.8V bank and is connected to on-board Ufl coaxial connector with PCB trace of about 12 mm length.



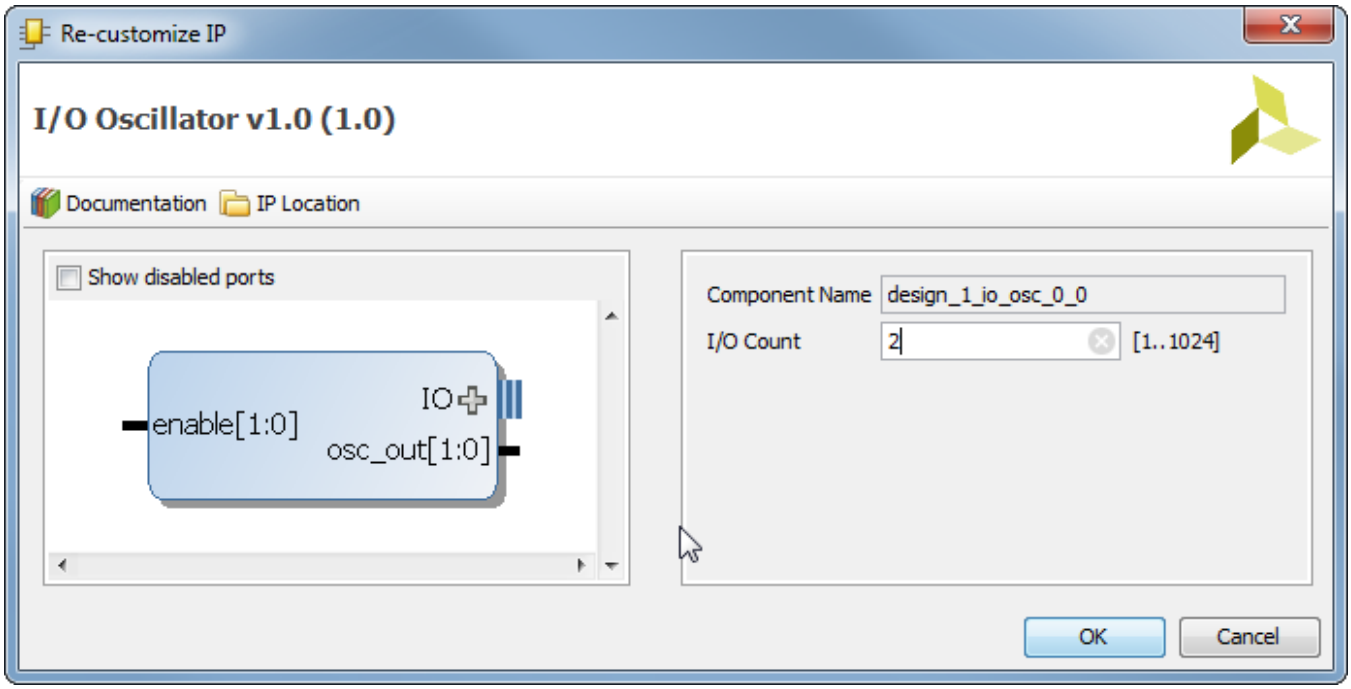
This the top level design.



Clock Wizard is used to configure the FPGA Clocking, I have selected 33.3333Mhz and 800MHz, please notice that "measurement" channels are configured with no BUFG.



Frequency meter IP Core configured for 4 channels.



I/O Oscillator configured for 2 channels. The default oscillator implementation uses no fabric logic only I/O Cell is used. This minimizes the impact of fabric delays and routing to the oscillation frequency.

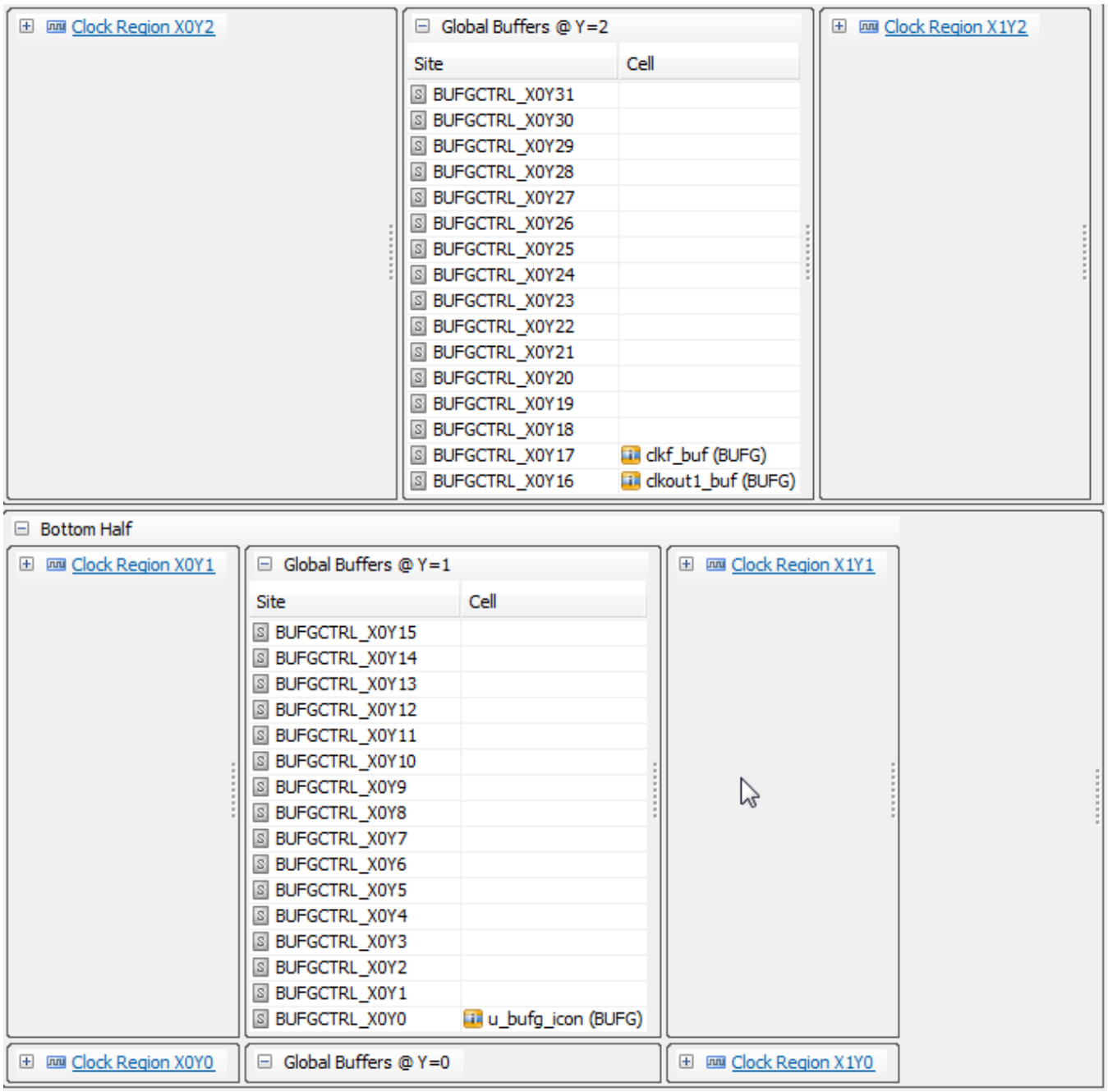
Utilization - Post-Implementation

Resource	Utilization	Available	Utilization %
FF	1588	126800	1.25
LUT	798	63400	1.26
Memory LUT	24	19000	0.13
I/O	3	210	1.43
DSP48	5	240	2.08
BUFG	3	32	9.38
MMCM	1	6	16.67

Graph **Table**

Post-Synthesis **Post-Implementation**

Utilization report: Frequency counter core uses no clock buffers (assuming the reference clock has a global buffer already). Frequency counter uses 1 DSP48E per channel, and 1 DSP48E for the reference clock. All clock measurement nets are driving exactly one load, the clock input of the DSP48E primitive, there is no other connections on that net. So it is irrelevant if the net is driven by some buffer or fabric local routing, the frequency performance of the channel is the same.



Clock resource allocation, clearly visible that all BUFG that are used are for the MMCM or Vivado ICON IP Core, none of the BUFG in the design is needed for the Frequency meter IP Core.