

FAQ - FPMC-FPDP

Does the TEK Micro FPMC-FPDP output connector use the "inverted" or "non-inverted" FPDP connector configuration?

Our FPDP module connector uses the "inverted" configuration. When you are connecting FPDP modules from different vendors, check first which configuration each uses before cabling them together.

How can I figure out the "actual" frequency being programmed by the clock synthesizer on a FastLink Product?

Both modules use a PLL-based clock synthesizer to generate arbitrary output frequencies under software control. The PLL synthesizer can generally match the desired frequency within 1000 ppm, depending on the relationship between the reference frequency and the desired output.

If you have installed TEK's software drivers for the module, you can simply attempt to configure the module for the desired frequency and the driver will respond with the "actual" frequency being programmed.

If you have not yet installed TEK's software drivers for the module, you may use the Cypress BitCalc software (available at <http://www.cypress.com/design/progprods/clock/clocks.html>) to check for a specific frequency.

The "actual" frequency reported by either TEK's software or Cypress's BitCalc includes all effects due to the programmable options in the PLL but assumes that the reference frequency is precisely accurate. Errors in the reference frequency will create proportional errors in the output frequency.

For most of TEK's products, the reference frequency is an oscillator frequency (33.3333 MHz) frequency divided by two. The PCI bus clock can also be selected as a reference frequency. Note that some PCI/PMC platforms use 33.000 MHz and others use 33.333 MHz; the TEK drivers assume that the PCI bus frequency is 33.333 MHz.

For example, if the desired output frequency is 6.35 MHz, TEK's software will generate a PLL program control word which will have an output of 6.349206 MHz (-125 ppm), assuming a local bus clock of 33.333333 MHz and a reference clock of 16.666667 MHz.

How can I convert a .TTF file to a .H header file?

We provide a utility to perform this function.

If an FPMC module has an Altera FPGA, how is the programming accomplished (is special hardware/cabling required to download code)?

For all of our FPGA-based FPMC modules (and IP modules for that matter), the FPGA is downloaded using the Altera "passive serial" download capability. The user software has access to a control register which drives nCONFIG, DATA0 and DCLK to the FPGA and monitors the nSTATUS and CONF_DONE. This allows FPGA download to be implemented with no special hardware or cables.

Can the FPGA code be "boot" resident or does it need to be downloaded to the FPGA before every use?

Our PMC and IP modules require an FPGA download for every power-up or reset.

Our software uses the Altera TTF output and generates a C language include file which can then be built into the application program. To minimize program space, we typically use a simple run-length compression algorithm on the bitstream and expand it when downloading to the FPGA.

