

Have you done performance benchmarks for throughput from either HotLinks or Taxi cards to RAID?

We have done performance benchmarks in VxWorks / PowerPC environments to JBOD and RAID storage on Motorola and Force VMEbus carriers and easily sustained 33 MB/s. We have also done benchmarks on VME/RACEway systems using our own PowerRACE carrier cards and sustained 66 MB/s from dual HOTLinks into a JBOD/RAID FibreChannel interface.

Is it possible to put two FastPMC-HL266 boards in one computer?

It is possible to put pretty much as many boards as you have address space for into a computer. Our drivers use a "handle" data structure which contains all of the address information for the card being used, so each API function can be used to talk to any number of cards by creating a different handle instance (with appropriate CPU address space pointers) for each card. However, if you are using PMC-to-PCI adapters to carry multiple cards in one PCI bus, please be aware that using multiple PMC-to-PCI adapters on a single PCI bus will violate the capacitive loading specification and is therefore not recommended.

Is the I/O available on the backplane or only on the front panel?

The I/O is only available on the front panel. The PMC does not have the I/O connector installed to bring I/O to the P2 connector.

Can you explain what the extra memory is used for?

The extra memory may be used by the FPGA for whatever buffering may be required. For raw streaming I/O, the memory is divided into a RX FIFO and a TX FIFO. The memory is physically SRAM but exhibits FIFO semantics because the FPGA program implements a FIFO in each direction using the SRAM. Some other FPGA programs we have developed use the memory as a frame buffer or to format, rasterize, or otherwise process packets or frames of data.

The card has physical FIFOs between the HOTLink and the FPGA, but those are rather small (2K x 9bit). The memory-based FIFOs make the card much less dependent on PCI bus latencies, as the FIFO depth is very large (128K x 32 in each direction).

We do not plan to be fiber channel compliant, we want to provide RAW data with custom packet format.

You can easily transfer raw data using the card; the standard FPGA programs simply stream byte data into 32-bit words and then into the buffer memory. The buffer memory looks like a FIFO to the PCI 9080 interface controller, which can then perform DMA transfers to the PCI bus. The logic which turns the buffer memory into a FIFO is part of the FPGA, so it can be made to do something different if need be.

What software does the FPMC-HL266 require?

Our web page has both driver software and FPGA programs. The driver software is a VxWorks driver which is turnkey for the Motorola PowerPC MVME2604 and VMETRO i960 MIDAS-120R. I would expect other Motorola PPC carriers to essentially plug and play. Other CPUs / carriers may require some integration; a point of reference is that it took us 2 days to port the code from the 2604 to the MIDAS. All of the BSP-specific code has to do with PCI configuration and is located in the one file tekpciVxWorks.c.

The driver software is available as source and object code at no additional cost. The current release is at <http://www.tekmicro.com/products.fastlinks.fpmc-hl266.html> as a TAR/GZIP file. Our development environment is Tornado 1.1 under Solaris, but it should work fine under WinNT as well.

What is HSVB?

HSVB (High Speed Video Bus) is an EIA-422 16-bit parallel bus used by Raytheon Amber imaging sensors for transfer of frame data. Amber developed a HSVB-over-HOTLink variant some time ago, and we have an FPGA program that is intelligent about HSVB frames over HOTLink.

Where can I buy the copper interface connector to the FPMC-HL266 if I need to make my own cabling?

ITT Cannon p/n MDM-9SSA is the connector part number. Look on their website at http://www.ittcannon.com/service_support/dealer.asp to find a distributor near you or call 1-800-845-7000 for RPI Apex, an ITT Cannon rep.

Do the FPMC-HL266plus-CF and the FPMC-HL266plus-CO support twisted pair interface?

The -CF and -CO models of the FPMC-HL266plus do support a twisted pair interface. The stock models are designed for a 100 Ohm differential interface; the details of the termination circuit may be found on pages 13 and 14 of the manual (<http://www.tekmicro.com/tm236B.pdf>).

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 this PMC, the reference frequency is the PCI bus clock frequency divided by two. 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.