

Preloaded Logic Architecture Layer (in SPI Flash): Reference Architecture Layer description

The reference design architecture was built using Xilinx Embedded Development Kit (EDK). Basically, it is an embedded system on a chip with a Xilinx MicroBlaze 32-bit soft microprocessor. The MicroBlaze (MB) initializes and sets up the system.

Standard EDK cores are used to implement a serial interface (XPS_UARTLITE), an SPI FLASH interface (XPS_SPI), a timer / counter block (XPS_TIMER) and an interrupt controller (XPS_INTC).

Besides standard IP cores, the reference architecture contains three custom IP cores (aka reference IP custom block).

custom IP core block	XPS_I2C_SLAVE	XPS_NPI_DMA	XPS_FX2
brief description	forwards MicroBlaze API Commands (MB Commands) coming from the USB bus towards the MicroBlaze	custom DMA between DDR SDRAM and other multiple sources;	used for high speed bidirectional communication between the FPGA and a host computer (also USB host)
longer dscription			

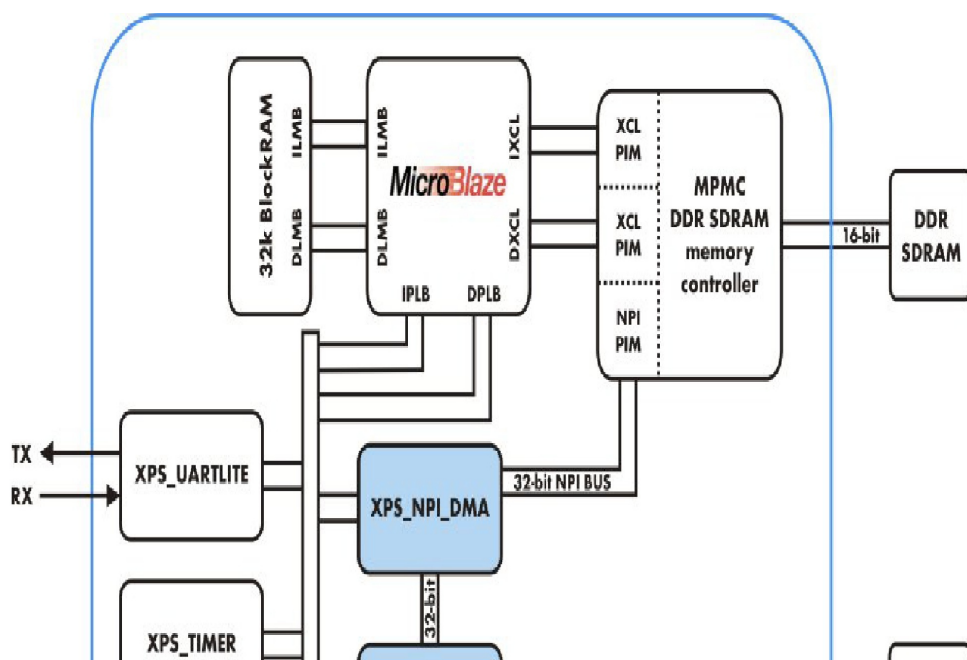
	XPS_I2C_SLAVE custom IP core block	XPS_NPI_DMA custom IP core block	XPS_FX2 custom IP core block
is used to deliver MB Command to FPGA's MicroBlaze soft P (through FX2MB_REGS) and retrieve "reply" (through MB2FX2_REGS)	✓	✗	✗
is used to realize a FPGA FX2 C connection	✓ (trough I2C)	✗	✓ (trough a 8 bit bus)
is used to realize a FPGA DRAM connection	✗	✓	✗

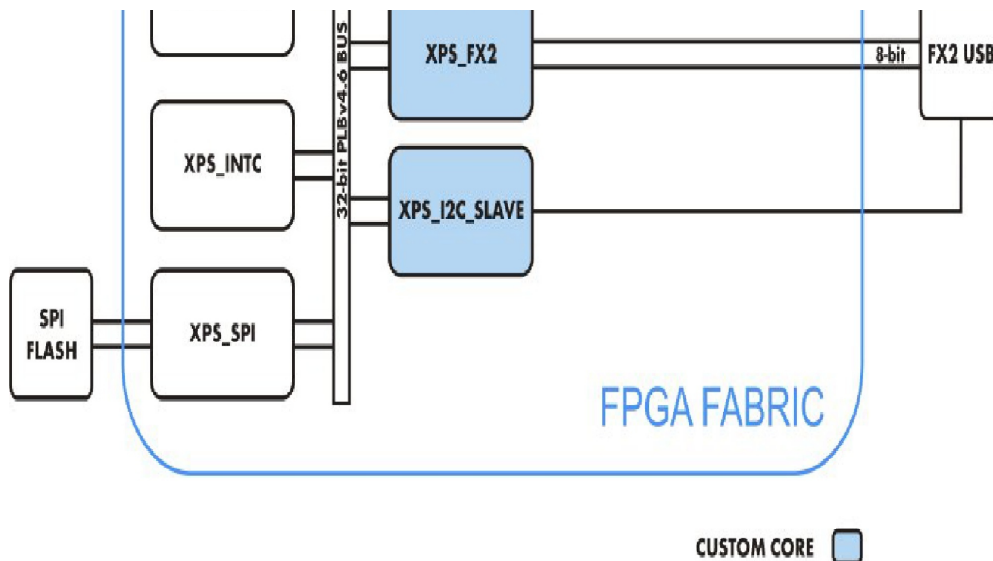
Reference IP custom blocks dependencies and connections

Legend:

- ✓ this symbol means that the custom IP block is used to (description in the corresponding colums)
- ✗ this symbol means that the custom IP block is NOT used to (description in the corresponding colums)

 XPS_NPI_DMA and XPS_FX2 could be used to realize a USB FX2 microcontroller DRAM connection and (using the [SW API Layer](#)) a USB (host computer) DRAM connection





Reference architecture block diagram.

XPS_I2C_SLAVE custom IP core block

It is a logic block for low speed bidirectional communication between the FPGA and a host PC. It is usually used for command, settings and status communication. It contains 6×32 -bit memory mapped registers:

- 3 for *host* => *FPGA* communication (FX22MB registers; FX22MB register 0 (aka FX22MB_REG0) is fundamental for [MicroBlaze API Commands \(MB Commands\)](#))
- 3 for *FPGA* => *host* communication (MB2FX2 registers)

When the host sends commands to the MicroBlaze (MB) soft embedded processor, an interrupt is triggered. When the MB writes data to MB2FX2_REG0, the interrupt (INT0) is sent to the Cypress EZ-USB FX2LP USB microcontroller (FX2 microcontroller from now on). When the FX2 microcontroller receives an interrupt, it reads all MB2FX2 registers. The FX2 microcontroller firmware elaborates these registers values.

Only for [FX22MB_REG0_GETVERSION command](#), the user should personally set an (FX2 microcontroller) interrupt, send a MB Commands, get an (FX2 USB microcontroller) interrupt. For all others MB Commands the interrupts are automatically managed by FX2 microcontroller firmware. See [MicroBlaze API Commands \(MB Commands\)](#).

VHDL code: [here](#).

MicroBlaze Driver code: [here](#).

A longer description: [here](#).

XPS_NPI_DMA custom IP core block

It is a high speed DMA (direct memory access) engine which connects to the MPMC (Multi-Port Memory Controller) VFBC (Video Frame Buffer Controller) port. It enables high speed data streaming to/from external memory (DDR SDRAM) and multiple sources. It can be controlled by a processor using 6×32 -bit memory mapped registers attached to the PLB (peripheral local bus). For more information about registers, see *Xilinx LogiCORE IP Multi-Port Memory Controller (MPMC)* data sheets ([Xilinx DS643](#)), *Video Frame Buffer Controller PIM* section.

When data is sent from the USB-host to a USB FX2 module high-speed endpoint (high speed communication channel), it is automatically stored into the RAM by the custom built DMA engine (XPS_NPI_DMA) at a specified buffer location. The reference design software running on the MicroBlaze verifies the transferred data at the end of transmission and sends to the USB host a notification about the data test (passed / failed).

VHDL code: [here](#).

MicroBlaze Driver code: [here](#).

A longer description: [here](#).

XPS_FX2 custom IP core block

It is a logic block for high speed bidirectional communication between the FPGA and a host PC. It contains 2 kbyte FIFOs for data buffering. More information about the 5×32 -bit memory mapped registers is contained in the `#project_root#/pcores/xps_fx2_v1_00_a/doc/` folder of the reference design project folder.

When data is sent from a USB FX2 module high-speed endpoint to the USB host, it is automatically fetched from the RAM via the custom DMA engine (XPS_NPI_DMA) and forwarded to the XPS_FX2 core in 1-kbyte packets. MicroBlaze throttles the throughput to prevent XPS_FX2 TX FIFO overflow.

VHDL code: [here](#).

MicroBlaze Driver code: [here](#).

A longer description: [here](#).

Source Code of the reference architecture

The source code of the reference architecture of the Trenz Electronic USB FX2 Technology Stack is released on GitHub ([TE0300](#), [TE0320](#) and [TE0630](#)).