Core Overview

The parallel input/output (PIO) core provides an interface between an Avalon Memory-Mapped (Avalon-MM) slave port and general-purpose I/O ports allowing management of external devices in situations where a "bit banging" approach is sufficient. Some example uses are:

- Controlling LEDs
- Acquiring data from switches
- Controlling display devices
- Configuring and communicating with off-chip devices, such as application-specific standard products (ASSP)

The output FIFO I/O interface is compatible with the SMILE (Switch Matrix I/O Local Expansion) IP allowing the I/O configuration by PC application. The I/O configuration file can be also generated by the free FEWE tools (Fpga Easy Web Editor) provided by GEB Enterprise.

Functional Description

Each PIO core can provide up to 8 bits I/O ports. An intelligent host such as a microprocessor controls the PIO ports by reading and writing the register-mapped Avalon-MM interface. Under control of the host, the PIO core captures data on its inputs and drives data to its outputs. When the PIO ports are connected directly to I/O pins, the host can tristate the pins by writing control registers in the PIO core.

When integrated into a Qsys-generated system, the PIO core has two user-visible features:

- 8 bits I/O ports that has been connected to logic inside the FPGA.
- A memory-mapped register space with the following functions and registers:
- Register, DATA_REG at DATA_REG_ADR, Data Register in/out
- Register, DIR_REG at DIR_REG_ADR, Data Direction Register
- Function, at OS_REG_ADR, DATA_REG Bit Set
- Function, at OC_REG_ADR, DATA_REG Bit Clear

Data Input and Output

The PIO core I/O ports are connect to off-chip logic. Each PIO bits can be configured as inputs, or outputs under the control of DIR_REG. The hardware logic is separate for reading and writing the PIO I/O data register, it has been splitted in two registers, DATA_REG_IN and DATA_REG_OUT located at the same address.
Reading the data register returns:

- The value present on the input ports through the DATA_REG_IN for each bit programmed as input.
- The previously written value on the DATA_REG_OUT for each bit programmed has output.

The DATA_REG can be also managed bit per bit; a "1" on i-th bits during a write to OS_REG_ADR will set the corresponding bit in DATA_REG, a "1" on i-th bits during a write to OC_REG_ADR will reset.

Register Map

An Avalon-MM master peripheral, such as a CPU, controls and communicates with the PIO core via some registers, shown below. The registers width is always considered 32 bits in the addressing space.

Offset	Register Name	R/W	Descriptions
0	DATA_REG	R	Data value currently on pins that are configured such as input, previously written value on the bits configured such as output.
		W	New value to drive on PIO bits configured such as out
1	DIR_REG	R/W	Individual direction control for each I/O port. A value of 0 sets the direction to input; 1 sets the direction to output.
4	OS_REG	W	Specifies which bit of the DATA_REG to set. The value is not stored into a physical register in the IP core, it's used to change the DATA_REG register value only.
5	OC_REG	W	Specifies which bit of the DATA_REG to clr. The value is not stored into a physical register in the IP core, it's used to change the DATA_REG register value only.