Parallel Port for Altera DE Boards

For Quartus II 8

1 Core Overview

The Parallel Port interface provides data transfer in either input or output (or both) directions. The transfer is done in parallel and it may involve from 1 to 32 bits. The number of bits, *n*, and the direction of transfer are specified by the user through Altera's SOPC Builder. This core is a version of the *PIO Core with Avalon® Interface* adapted for use with Altera's DE2/DE1 boards.

2 Functional Description

The Parallel Port registers are accessible as memory-mapped locations. A full description of the Parallel Port module can be found in the document PIO Core with Avalon Interface, which is available in the literature section of Altera's web site. This core is different than the PIO Core with Avalon Interface as it has some presets to configure the Parallel Port for the Simple I/O on the DE2/DE1 Boards, such as the LEDs, Switches, etc.

3 Instantiating the Core in SOPC Builder

See the PIO Core with Avalon Interface for details.

4 Software Programming Model

4.1 Register Map

The PIO can have up to four registers, as shown in Table 1. These registers have a configurable data width, *n*, which can be set through the SOPC Builder Component Wizard.

Not all of these registers are generated in a given PIO interface. For example, the *Direction* register is included only when a bidirectional interface is specified.

4.1.1 *Data* Register

This register holds the *n* bits of data that are transferred between the PIO interface and the Nios II processor. It can be implemented as an input, output, or a bidirectional register by the SOPC Builder.

Table 1. PIO register map				
Offset in bytes	**Register name**		**Read/Write**	**Bits** $(n-1)\ldots 0$
0	`data`	Input	R	Data value currently on PIO inputs.
		Output	W	New value to drive on PIO outputs.
4	`direction`		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.
8	`interruptmask`		R/W	IRQ enable/disable for each input port. Setting a bit to 1 enables interrupts for the corresponding port.
12	`edgecapture`		R/W	Edge detection for each input port.

4.1.2 *Direction* Register

The *direction* register defines the direction of the transfer for each of the n data bits when a bidirectional interface is generated. A value of 0 sets the direction to input; a value of 1 sets the direction to output.

4.1.3 *Interruptmask* Register

The *interruptmask* register is used to enable interrupts from the input lines connected to the PIO.

4.1.4 *Edgecapture* Register

The *edgecapture* register indicates when a change of logic value is detected in the signals on the input lines connected to the PIO.

Consult the PIO Core with Avalon Interface document for a full description of these registers and their functionality.

4.2 Programming with the Parallel Ports

The Parallel Port core is packaged with C-language device drivers accessible through the hardware abstraction layer (HAL). These functions implement basic operations for the Parallel Port.

To use the functions, the C code must include the statement:

```
#include "altera_up_avalon_parallel_port.h"
```

4.2.1 alt_up_parallel_port_open_dev

Prototype: `alt_up_parallel_port_dev* alt_up_parallel_port_open_dev(const char *name)`
Include: `<altera_up_avalon_parallel_port.h>`
Parameters: `name` – the parallel port name. For example, if the parallel port name in SOPC Builder is "green_leds", then *name* should be "/dev/green_leds"
Returns: The corresponding device structure, or NULL if the device is not found.
Description: Open the parallel port device specified by *name* .

4.2.2 alt_up_parallel_port_read_data

Prototype: `unsigned int alt_up_parallel_port_read_data(alt_up_parallel_port_dev *parallel_port)`
Include: `<altera_up_avalon_parallel_port.h>`
Parameters: `parallel_port` – struct for the parallel port device .
Returns: data – The data read for the parallel port.
Description: Read from the data register of the parallel port.

4.2.3 alt_up_parallel_port_write_data

Prototype: `void alt_up_parallel_port_write_data(alt_up_parallel_port_dev *parallel_port, unsigned data)`
Include: `<altera_up_avalon_parallel_port.h>`
Parameters: `parallel_port` – struct for the parallel port device.
`data` – The data to be written to the parallel port.
Description: Write to the data register of the parallel port.

4.2.4 alt_up_parallel_port_read_direction

Prototype: `unsigned int alt_up_parallel_port_read_direction(alt_up_parallel_port_dev *parallel_port)`
Include: `<altera_up_avalon_parallel_port.h>`
Parameters: `parallel_port` – struct for the parallel port device.
Returns: direction – The direction read for the parallel port.
Description: Read from the direction register of the parallel port.

4.2.5 alt_up_parallel_port_set_port_direction

Prototype: `void alt_up_parallel_port_set_port_direction(alt_up_parallel_port_dev *parallel_port, unsigned direction)`
Include: `<altera_up_avalon_parallel_port.h>`
Parameters: `parallel_port` – struct for the parallel port device.
Description: Set the direction register of the parallel port.

4.2.6 alt_up_parallel_port_set_all_bits_to_input

Prototype: `void alt_up_parallel_port_set_all_bits_to_input(alt_up_parallel_port_dev *parallel_port)`
Include: `<altera_up_avalon_parallel_port.h>`
Parameters: `parallel_port` – struct for the parallel port device.
Description: Set the direction of all bits of the parallel port to be inputs.

4.2.7 alt_up_parallel_port_set_all_bits_to_output

Prototype: `void alt_up_parallel_port_set_all_bits_to_output(alt_up_parallel_port_dev *parallel_port)`
Include: `<altera_up_avalon_parallel_port.h>`
Parameters: `parallel_port` – struct for the parallel port device.
Description: Set the direction of one bits of the parallel port to be outputs.

4.2.8 alt_up_parallel_port_set_bit_to_input

Prototype: `void alt_up_parallel_port_set_bit_to_input(alt_up_parallel_port_dev *parallel_port, unsigned int bit)`
Include: `<altera_up_avalon_parallel_port.h>`
Parameters: `parallel_port` – struct for the parallel port device.
`bit` – The bit of the parallel port to be set as an input.
Description: Set the direction of one bit of the parallel port to be input.

4.2.9 alt_up_parallel_port_set_bit_to_output

Prototype: `void alt_up_parallel_port_set_bit_to_output(alt_up_parallel_port_dev *parallel_port, unsigned int bit)`
Include: `<altera_up_avalon_parallel_port.h>`
Parameters: `parallel_port` – struct for the parallel port device.
`bit` – The bit of the parallel port to be set as an output.
Description: Set the direction of one bit of the parallel port to be output.

4.2.10 alt_up_parallel_port_read_interrupt_mask

Prototype: `unsigned int alt_up_parallel_port_read_interrupt_mask(alt_up_parallel_port *parallel_port)`
Include: `<altera_up_avalon_parallel_port.h>`
Parameters: `parallel_port` – struct for the parallel port device.
Returns: data – The current interrupt mask of the parallel port.
Description: Read from the interrupt mask register of the parallel port.

4.2.11 alt_up_parallel_port_set_interrupt_mask

Prototype: `void alt_up_parallel_port_set_interrupt_mask(alt_up_parallel_port_dev *parallel_port, unsigned mask)`
Include: `<altera_up_avalon_parallel_port.h>`
Parameters: `parallel_port` – struct for the parallel port device.
`mask` – The interrupt mask to be set in the parallel port.
Description: Set the interrupt mask register of the parallel port.

4.2.12 alt_up_parallel_port_read_edge_capture

Prototype: `unsigned int alt_up_parallel_port_read_edge_capture(alt_up_parallel_port_d *parallel_port)`
Include: `<altera_up_avalon_parallel_port.h>`
Parameters: `parallel_port` – struct for the parallel port device.
Returns: data – The current edge capture register of the parallel port.
Description: Read from the edge capture register of the parallel port.

4.2.13 alt_up_parallel_port_clear_edge_capture

Prototype: `void alt_up_parallel_port_clear_edge_capture(alt_up_parallel_port_dev *parallel_port)`
Include: `<altera_up_avalon_parallel_port.h>`
Parameters: `parallel_port` – struct for the parallel port device.
Description: Clear the edge capture register of the parallel port.

4.2.14 alt_up_parallel_port_read_interrupt_pending

Prototype: `unsigned int alt_up_parallel_port_read_interrupt_pending(alt_up_parallel_p *parallel_port)`
Include: `<altera_up_avalon_parallel_port.h>`
Parameters: `parallel_port` – struct for the parallel port device.
Returns: pending – The interrupt pending bit field of the parallel port.
Description: Read interrupt pending information of the parallel port.

■