CF + Interface

Introduction

Compact Flash (CF) is a mass storage device format used in portable electronic devices like digital cameras, PDAs and cellular phones. For storage, Compact Flash typically uses flash memory in a standardized enclosure.

The design is implemented in VHDL. The Lattice iCEcube2™ Place and Route tool integrated with the Synopsys Synplify Pro® synthesis tool is used for the implementation of the design. The design can be targeted to other iCE40™ FPGA product family devices.

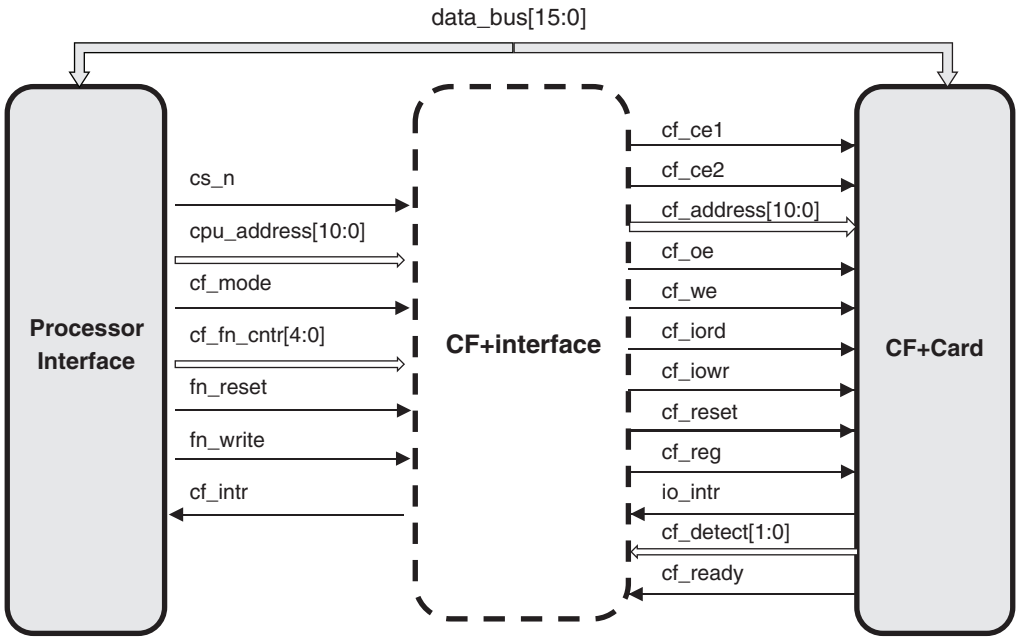
Figure 1 shows the System Block Diagram for the CF+ interface.

Features

- Supports PC Card ATA using IO
- Supports PC Card ATA using memory

System Block Diagram

Figure 1. System Block Diagram



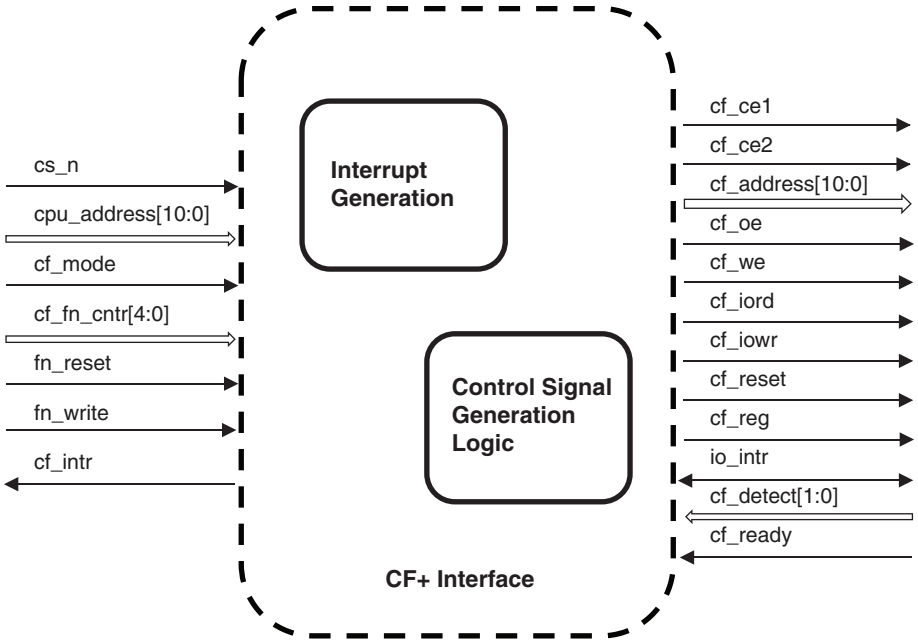
Signal Description

Table 1. Signal Description

Signal	Width	Type	Description
cpu_address	11	Input	Address bus from host
cf_mode	1	Input	Selection lines to choose from I/O and memory mode operations
cf_fn_cntrl	5	Input	Selection lines to choose from I/O & memory R/W mode operations
fn_reset	1	Input	Asynchronous active low reset pin asserted by host
fn_write	1	Input	Rising edge generates signals based on cf_fn_cntrl
cf_intr	1	Output	Interrupt line to the host, Active low for card read
cf_ready	1	Input	Asserted low during power up or reset & made high in the memory mode
cf_detect	2	Input	Active low card detect signals
io_intr	1	Input	Interrupt request from I/O device
cs_n	1	Input	Active low chip select
cf_reg	1	Output	Low during I/O operations, used in memory mode
cf_reset	1	Output	Compact flash reset pin, Active high
cf_iowr	1	Output	Active low I/O write select line
cf_iord	1	Output	Active low I/O read select line
cf_we	1	Output	Active low write enable line used to write to configuration registers
cf_oe	1	Output	Active low output enable line
cf_address	11	Output	Address Bus
cf_ce1	1	Output	Active low card select line
cf_ce2	1	Output	Active low card select line

Design Module Description

Figure 2. Functional Block Diagram



Interrupt Generation

An interrupt is generated and communicated to the host processor on the cf_intr line when a valid card insertion is detected.

Control Signal Generation Logic

The CF+ interface design monitors host side transaction requests to the CF+ card or the I/O card. Depending on the host processor request, the appropriate control signals are sent to the CF+ along with the address.

The sequence of operation is as follows:

1. Host processor pulls cs_n low and enables the CF+ card interface
2. The pins cf_detect[1:0] goes low when CompactFlash card inserted into the socket
3. CF+ Interface generates a low on cf_intr and communicates to the host processor that CF+ Interface is now activated
4. Host generates a high on fn_write indicating to the Interface that the host is ready for data transaction
5. All the transactions of the interface, CF+ card and host processor are synchronized with the fn_write signal.

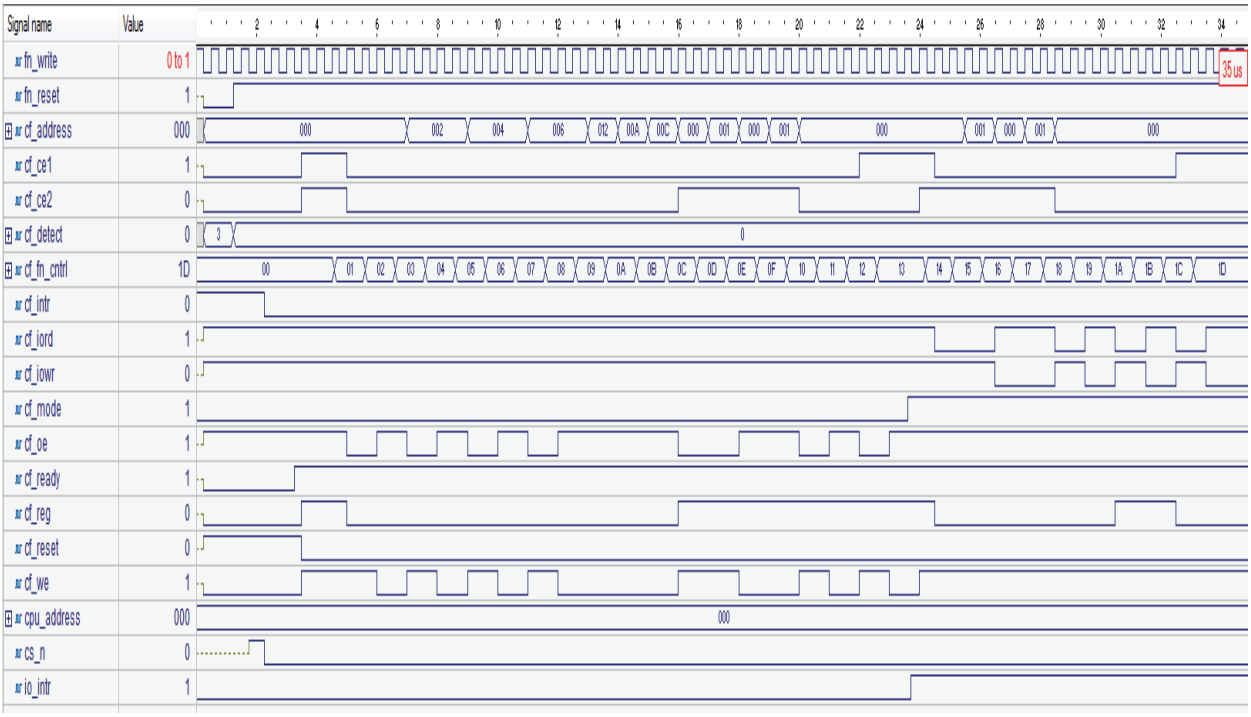
On the rising edge of the fn_write, the FSM samples cf_fn_cntrl[4:0] pins. Each operation in both the modes is encoded with 5 bits specified by the controller using the cf_fn_cntrl pins. The FSM on every rising edge of fn_write decodes the cf_fn_cntrl line and generates the necessary control signals to carry out the requested operation successfully. The control signals generated by the interface during memory operation are cf_we and cf_oe, while during I/O operations the control signals generated are cf_iord and cf_iowr. The other control signals generated by CF+ Interface are CF enable signals, cf_ce1 and cf_ce2, and memory enable signal cf_reg, are common for both modes. The logic levels of these generated signals are defined as per the CF+ standard.

The host data bus [15:0] and card data bus[15:0] are connected to each other without going through the FPGA.

The fn_reset signal is generated by the host. When receiving this, iCE40 generates the reset signal to the CF+ card. The fn_reset triggers a high on cf_reset(hardware reset) as well as the control signals necessary to configure the configuration registers for CF+ reset functionality.

Simulation Waveforms

Figure 3. Simulation Waveforms



Implementation

This design is implemented in VHDL. When using this design in a different device, density, speed or grade, performance and utilization may vary.

Table 2. Performance and Resource Utilization

Family	Language	Utilization (LUTs)	f_{MAX} (MHz)	I/Os	Architectural Resources
iCE40[1]	VHDL	53	>50	43	(18/160)PLBs

1. Performance and utilization characteristics are generated using iCE40LP1K-CM121 with iCEcube2 design software.

References

- iCE40 Family Handbook

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com

Revision History

Date	Version	Change Summary
April 2013	01.0	Initial release.