SCUBA-2 Block Specification

Cyclic Redundancy Check

February 11, 2004

Table of Contents

1. Block Overview

1.1 Block Location and Block Interface Within System

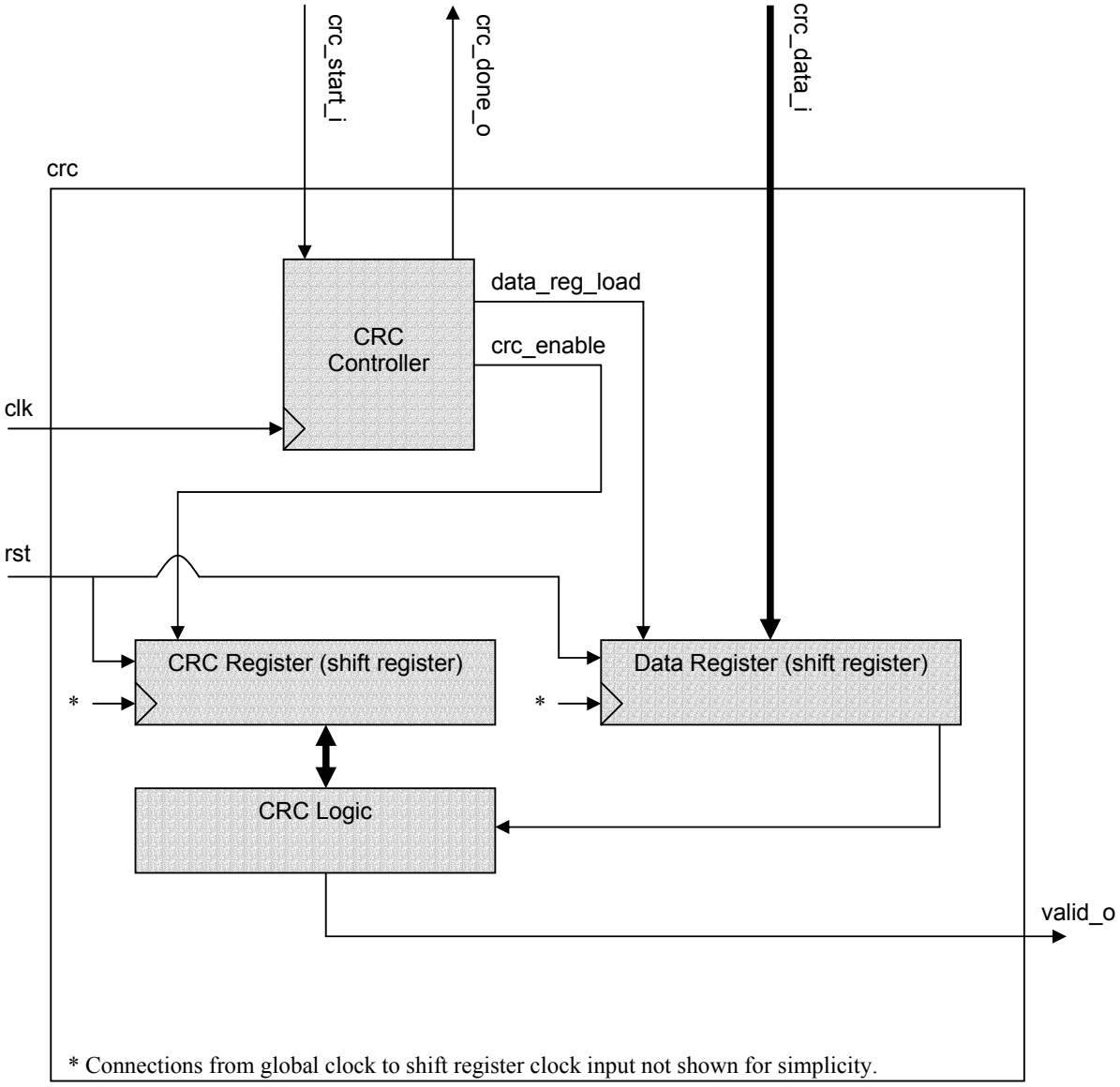
The cyclic redundancy check (CRC) is incorporated into the following modules:

- card_id
- temperature

1.2 Block Functionality / Features

- CRC polynomial: $x^8 + x^5 + x^4 + 1$
- CRC computed serially (LSB first), one bit per clock cycle.

1.3 Block Dataflow



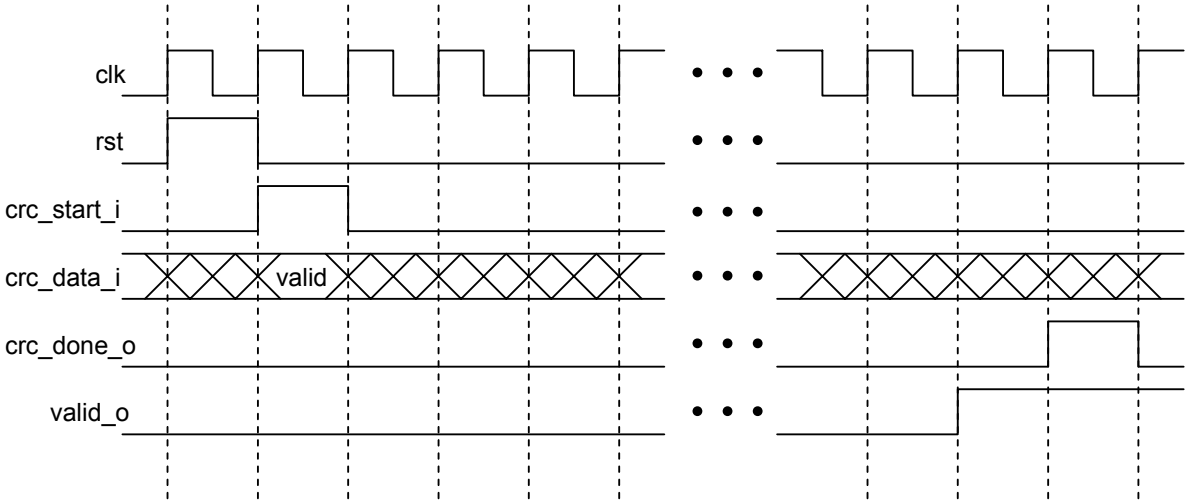
2. Block Interfaces

2.1 Interface Signal Description

Table 1: Interface Signals

Signal	Description	Direction
Global Signals		
clk	Global clock signal	in
rst	Global asynchronous active-high reset	in
CRC Controls		
crc_start_i	Active-high start signal. Tells CRC FSM to begin processing.	in
crc_done_o	Active-high done signal. Indicates that the CRC has been computed and/or the CRC has been validated.	out
CRC Data		
crc_data_i	CRC input data.	in
valid_o	Active-high valid flag. Indicates that the CRC is valid.	out

2.2 Interface Protocol and Timing



3. High-Level Description

3.1 State Machine Description

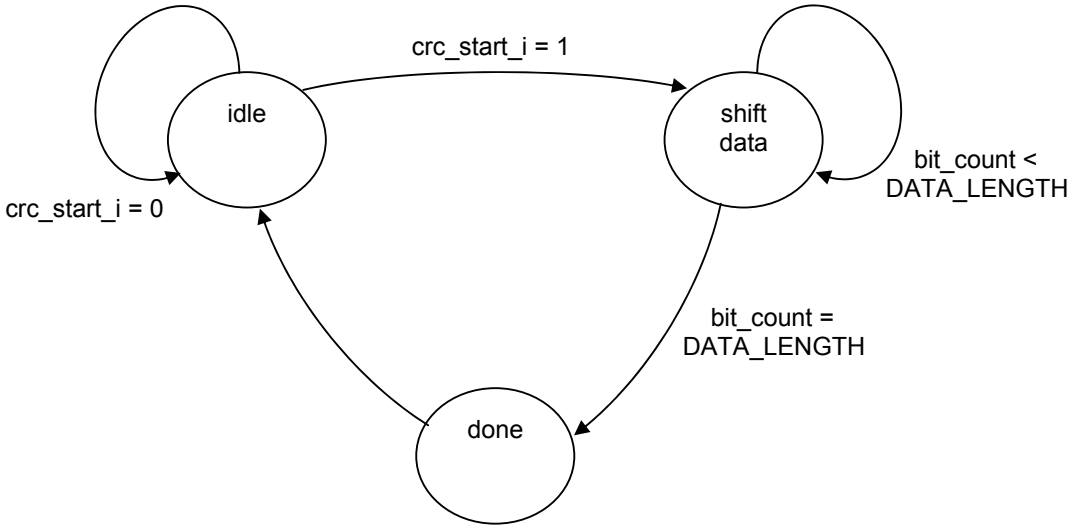


Table 2: State Machine Outputs

State	Signal	Value
IDLE	data_reg_load	1
	crc_enable	0
	crc_done_o	0
SHIFT_DATA	data_reg_load	0
	crc_enable	1
	crc_done_o	0
DONE	data_reg_load	0
	crc_enable	0
	crc_done_o	1

Table 3: Description of State Machine Outputs

Signal	Description
data_reg_load	Stores the input data in a data register.
crc_enable	Enables shifting out of data from data register.
crc_done_o	Indicates that CRC has been computed.

4. Files of the Block

4.1 Source Code

4.1.1 crc.vhd

This file contains the CRC implementation. It is compiled into the “components” library.

4.2 Header Code

Not applicable for this block.