Summary - Matchstiq-Z1 I2C

Name	matchstiq_z1_i2c
Worker Type	Device
Version	v1.5
Release Date	4/2019
Component Library	ocpi.assets.platforms.matchstiq_z1.devices
Workers	matchstiq_z1_i2c.hdl
Tested Platforms	Matchstiq-Z1(PL)

Worker Implementation Details

The Matchstiq-Z1 I2C device worker uses the subdevice construct to implement the I2C bus for the Matchstiq-Z1 platform. Matchstiq-Z1 I2C supports 5 device workers:

1. Si5338
2. Matchstiq-Z1 AVR
3. Pca9534
4. Pca9535
5. Tmp100

Matchstiq-Z1 I2C uses the i2c primitive library which is based upon the OpenCores I2C controller. This revision of the device worker supports 8 bit and 16 bit I2C accesses.

Block Diagrams

Top level

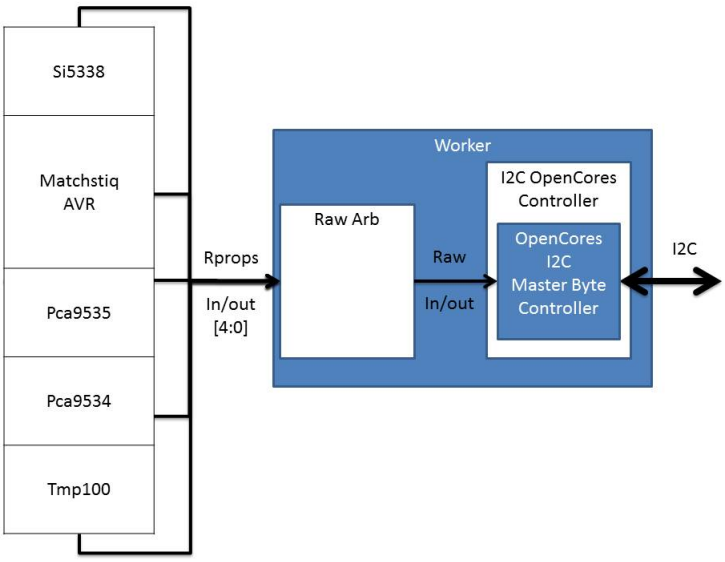

Figure 1: I2C Connection Block Diagram

State Machine

Source Dependencies

matchstiq_z1_i2c.hdl

- assets/hdl/platforms/matchstiq_z1/devices/matchstiq_z1_i2c.hdl/matchstiq_z1_i2c.vhd
- assets/hdl/primitives/i2c/i2c_pkg.vhd
  - assets/hdl/primitives/i2c/i2c_opencores_ctrl.vhd
  - assets/hdl/primitives/i2c/i2c_master_byte_ctrl.v
  - assets/hdl/primitives/i2c/i2c_master_bit_ctrl.v
  - assets/hdl/primitives/i2c/timescale.v
  - assets/hdl/primitives/i2c/i2c_master_defines.v
- core/hdl/primitives/ocpi/raw_arb.vhd

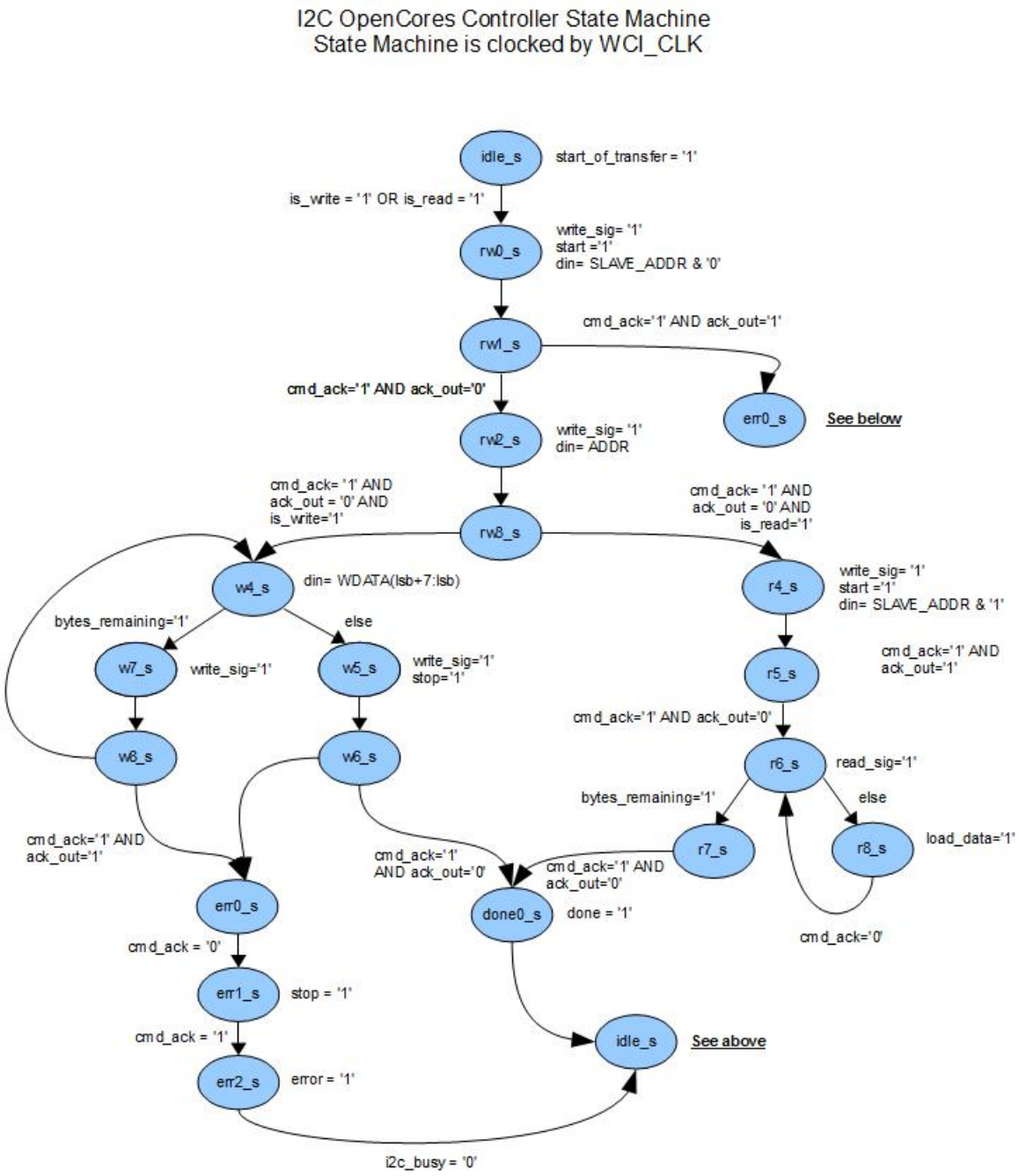

Figure 2: I2C OpenCores Controller State Machince

Component Spec Properties

Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
NUSERS_p	-	-	-	Readable, Parameter	-	5	Number of supported devices
SLAVE_ADDRESS_p	UChar	-	NUSERS_p	Readable, Parameter	-	-	Array of I2C Slave Addresses
CLK_FREQ_p	Float	-	-	Readable, Parameter	-	100e6	Input clock rate which is divided down to create I2C clock

Worker Interfaces

matchstiq_z1_i2c.hdl

Type	Name	DataWidth	Advanced	Usage
RawProp	rprops	-	Count=NUSERS_p Optional=true	Raw properties connections for master devices Index 0: matchstiq_z1_avr Index 1: si5338 Index 2: tmp100 Index 3: pca9534 Index 4: pca9535

Signals

Name	Type	Width	Description
SDA	Inout	1	I2C Data
SCL	Inout	1	I2C Clock

Control Timing and Signals

The Matchstiq-Z1 I2C HDL device worker uses the clock from the Control Plane and standard Control Plane signals.

Worker Configuration Parameters

matchstiq_z1_i2c.hdl

Table 1: Table of Worker Configurations for worker: matchstiq_z1_i2c

Configuration	SLAVE_ADDRESS_p	CLK_FREQ_p	ocpi_endian	NUSERS_p	ocpi_debug
0	69113723332	100000000.0	little	5	false
1	69113723332	50000000.0	little	5	false

Performance and Resource Utilization

matchstiq_z1_i2c.hdl

Table 2: Resource Utilization Table for worker "matchstiq_z1_i2c"

Configuration	OCPI Target	Tool	Version	Device	Registers (Typ)	LUTs (Typ)	Fmax (MHz) (Typ)	Memory/Special Functions
0	zynq	Vivado	2017.1	xc7z020clg484-1	104	328	N/A	N/A
1	zynq	Vivado	2017.1	xc7z020clg484-1	104	328	N/A	N/A

Test and Verification

Testing of the Matchstiq-Z1 I2C device worker consists of a C++ test bench that use the Application Control Interface API to command the UUT.

Hardware

The testbench for this worker checks the functionality of the I2C devices and generates an output file with the received input data.

Building the test assembly requires that the *matchstiq_z1* platform has been built. Details on how to build the *matchstiq_z1* platform can be found in the Matchstiq-Z1 platform document. To build the testbench's assembly and ACI, follow the instructions that are provided by running ***make show*** within this test's directory.

Connect a signal generator to the input "RX" channel. Configured the signal generator to produce tone at a frequency of 2.140001 GHz and amplitude -55 dBm.

Execute and validate the output of the test by continuing to follow the instructions provided by running ***make show*** within this test's directory.

An example of the terminal output is provide below:

```
% ./target-xilinx13_3/testbench
Application XML used for testbench: ./hw_testbench_app_file.xml
Start of Testbench
Set Sampling Clock to 200 kHz (100 kSps):
PCA9535: Starting Test
PCA9535: Testing filter bandwidth:
PCA9535: Set unfiltered
PCA9535: Set filter bandwidth to 300 to 700 MHz
PCA9535: Set filter bandwidth to 625 to 1080 MHz
PCA9535: Set filter bandwidth to 1000 to 2100 MHz
PCA9535: Set filter bandwidth to 1700 to 2500 MHz
PCA9535: Set filter bandwidth to 2200 to 3800 MHz
PCA9535: Set filter bandwidth to unfiltered
PCA9535: Testing Lime RX input:
PCA9535: Set Lime RX input to 2
PCA9535: Set Lime RX input to 3
PCA9535: Testing Pre-lime LNA:
PCA9535: Setting Pre-lime LNA off
PCA9535: Setting Pre-lime LNA on
PCA9535: End of Test
Matchstiq-Z1 AVR: Starting Test
Matchstiq-Z1 AVR: Testing attenuator:
Matchstiq-Z1 AVR: Reset attenuator to 0:
Matchstiq-Z1 AVR: Testing LED:
Matchstiq-Z1 AVR: Set LED off
Matchstiq-Z1 AVR: Set LED green
Matchstiq-Z1 AVR: Set LED red
Matchstiq-Z1 AVR: Set LED orange
Matchstiq-Z1 AVR: Testing Serial Number:
Matchstiq-Z1 AVR: Serial number is: 6188
Matchstiq-Z1 AVR: Testing WARP voltage register:
Matchstiq-Z1 AVR: Set WARP voltage to 2048
Matchstiq-Z1 AVR: End of Test
TMP100: Starting Test
TMP100: Testing temperature:
TMP100: Temperature is: 42 degrees C
TMP100: End of Test
```

Additionally, an output file is produced odata/testbench_rx.out which can be plotted. Figure 1 shows the expected result for the received data. These results should be inspected manually as the testbench does not verify these trends.

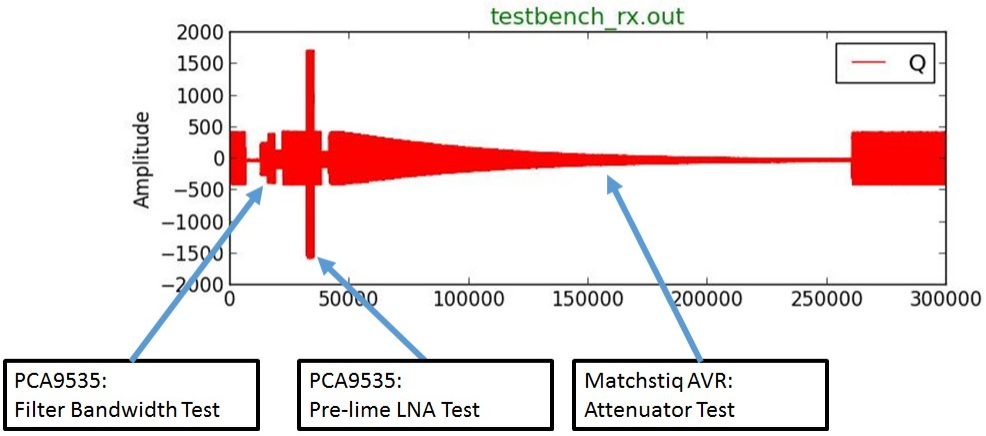

Figure 3: Expected Results

References

1) The Matchstiq-Z1 Software Development Manual (provided by Epiq with the Platform Development Kit)