I2C BERT

- Author: Darryl Miles
- Description: I2C Bit Error Rate Test
- Language: SpinalHDL

How it works

This text will be updated nearer the scheduled TT05 redistribution time (early 2024) along with the project github README.md and gh-pages documentation. Please regenerate your documentation.

Issue synchronous reset, ensure interface inputs are set to zero. Power-on-reset configuration is possible via the input pins, see documentation.

This design is an I2C peripheral that implements an 8-bit ALU over I2C. The purpose of the ALU is to allow pattern testing to occur and read back the accumulated result.

There are a few clocking modes, the default uses SCL pin as per the standard.

Connection to I2C interface: * uio[2] = SDA (should be direct to RP2040 pin with capable mode) * uio[3] = SCL (shouid be direct to RP2040 pin with capable mode)

When in open-drain mode the standard pull-up resistor is in the order of 4k7 to 10k and no more than 400pF capacitance on lines. Higher speeds my require attention to those metrics for your setup. The project is peripheral only and does not drive SCL. So open-drain or push-pull can be used by the controller no matter the mode setup in this project.

Power-on-reset configuration (set all zero for standard mode): * ui_in[1] sets CLOCKMUX to use divider * ui_in[2] sets PUSHPULL I2C bus mode (by default open-drain is in use) * ui_in[3] activates DIV12 divider setup on reset (default is 10Mhz for 10Khz) * {uio_in[7:0], ui_in[7:4]} is the DIV12 value to use

The design is based around a high-speed clock, at default speed of 10MHz with

Other than the default divider setup for CLOCKMUX mode there is no restriction upon the system clock used, other than trying to operate at low ratios of system-clock:SCL. The design has been simulated from "3:1" upto 1000000:1. Maybe lower than 3:1 is possible.

How to test

RP2040 code is expected to be provided to conduct testing based on simulation expectations.

IO

#	Input	Output	Bidirectional
0	i2cSampleDivisor bit0	segment a	none
1	i2cSampleDivisor bit1	segment b	none
2	none	segment c	I2C SCL (bidi)
3	none	segment d	I2C SDA (bidi)
4	none	segment e	none
5	none	segment f	none
6	none	segment g	none
7	none	dot	powerOnSense (out)