# Board Checkout Firmware

## Introduction

The basic Board Checkout Firmware is designed to verify functionality of a board without any dependencies other than a functioning FPGA.

Implemented modules:

* Clock counting
* I2C bus scanning
* SPI flash device identification
* LED verification

In planning:

* JTAG scan (for soft JTAG)
* GTP verification (using loopback)
* UART verification (loopback)

## Usage

The only things in the RTL that really need to be changed are the following lines in *board\_checkout.v*:

// Board checkout parameters. These specify the number of things this board has to check out.  
`DEFINE\_IF(NUM\_DIFF\_CLOCKS, 0)  
`DEFINE\_IF(NUM\_SE\_CLOCKS, 1)  
`DEFINE\_IF(NUM\_I2C, 1)  
`DEFINE\_IF(NUM\_SPI\_FLASH, 1)  
`DEFINE\_IF(NUM\_LED, 4)

These parameters define the number of differential clocks (here 0), single-ended clocks (here 1), I2C busses (here 1), SPI flash devices (1), and LEDs (4).

Additional necessary ports (enables, etc.) can be put in the top module where indicated.

The constraints file obviously needs to be created to update pin locations. Note that the following I/Os are always vectors/arrays and must be constrained as such (even if there’s only 1):

* Differential clocks (DIFFCLK\_P[n:0]/DIFFCLK\_N[n:0])
* Single-ended clocks (CLK[n:0])
* I2C busses (SDA[n:0], SCL[n:0])
* LEDs (LED[n:0])
* SPI flash chip selects (SPI\_CS\_B[n:0])

Also note that if the SPI flash bus does NOT use the CCLK pin, it must be connected by user (it’s named SPI\_SCLK in the RTL).

## Testing

Once that’s done and the firmware is built, a VIO should exist for each clock, I2C bus, and another for the SPI Flash testing.

## Clock Verification

The Clock VIOs contain the *approximate* clock frequency divided by 1000. This frequency is determined by the internal ring oscillator, and so can be off by as much as 50% - but in typical (room temperature) testing, it’ll likely be ~10-20% accurate.

## SPI Flash Verification

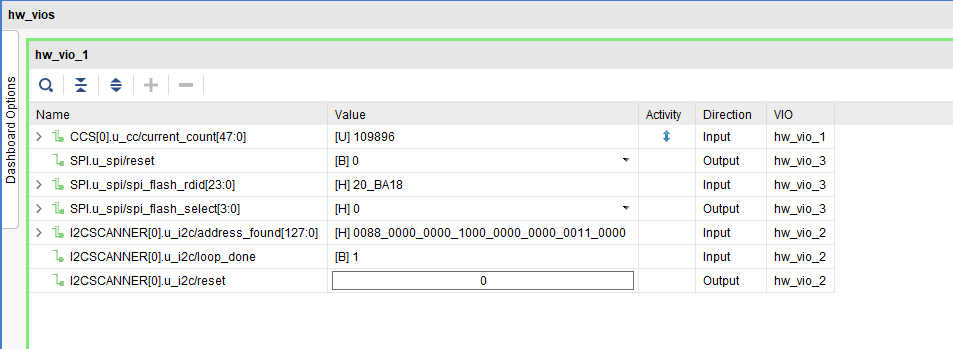
The SPI Flash VIO contains the 3-byte response to the RDID instruction. An additional VIO output can reset the controller used to read this value, reading it again, and another output selects which of the chip selects to use. Note that if there is only 1 chip select, changing this value will cause the RDID to fail (since nothing responds). After changing spi\_flash\_select, the controller must be reset to reissue RDID.

## I2C Bus Verification

The I2C Bus VIO contains a 128-bit vector which corresponds to all of the 7-bit I2C addresses that did not NAK when a zero-byte write was attempted to them. That is, the device address was written to (with the last bit low) and then a STOP was issued.

There is also a single-bit input indicating that the scan loop has completed (so the values are accurate) and an additional single-bit output which resets the scan loop.

# Example

This image shows the VIO Dashboard with all probes enabled for an Arty-A7 board with several I2C devices attached to it.



Here we can see that the clock is active with a frequency of ~110 MHz (which is about 10% too high: its true value is 100 MHz – again, this is due to the imprecise ring-oscillator in the FPGA, not the clock itself). The SPI Flash attached has a RDID of 20 BA 18, and several I2C devices are present.