User Manual – Bring Up Plan

Overview

This is a growing document which includes portions of and builds off a previous senior design team's work, sdmay23-28. This document includes a basic Nucleo test plan, and expands to fit our specific design.

Purpose

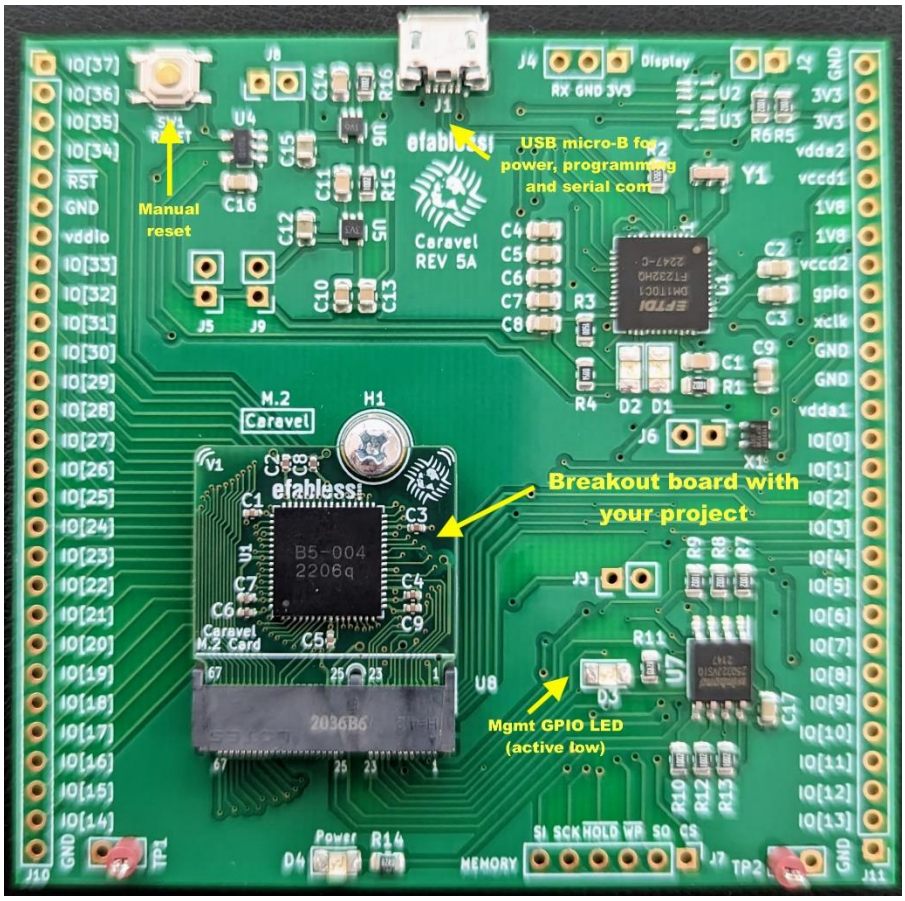
The purpose of this document is to guide the students on how to test Efabless project from our team. This document provides firmware examples, flash programming and diagnostic tools for testing Open MPW and chipIgnite projects using Caravel. It also provides schematics, layout and gerber files for PCB evaluation and breakout boards.

Development Board Description:

The larger green board below is the Caravel Development Board, which contains:

- Reset button
- USB to micro-B programming port
- 38 GPIO ports
- External Reset
- External power connections
- M.2 Edge connector for Caravel Breakout Board
- Jumper connectors to determine power sources

The Caravel Development Board contains two 28 pin connectors on both sides of the board, which including 38 GPIO pins, power connections, and an external clock connection. These GPIO pins can be used to interface with our Backdoor SPI interface.

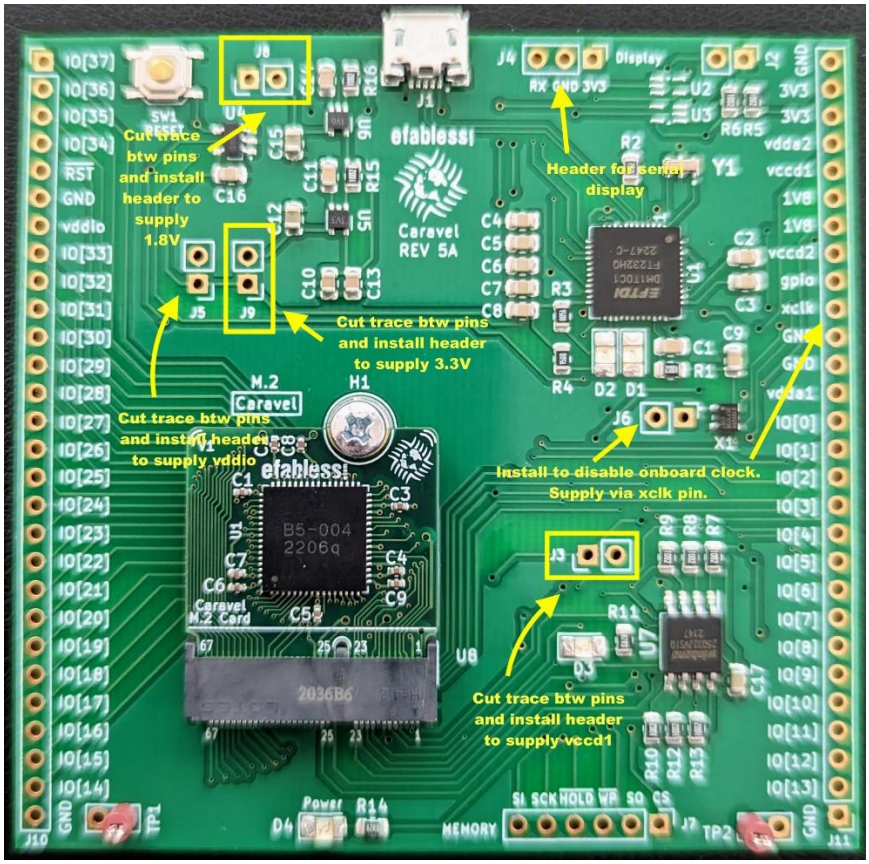


Efabless Development Board and Breakout Board[1]

I/O Connections

Notice:

- The clock is driven by X1 with a frequency of 10MHz. To drive the clock with custom frequency, disable X1 through installing J6 and use the external pin for xclk
- The voltage regulator U5 and U6 supply 1.8V and 3.3V through J8 and J9. The traces have to be cut if they are supplied externally.
- vccd1 is connected to 1.8V through J3. The trace has to be cut if it is supplied externally
- vddio is connected to 3.3V through J5. The trace has to be cut if it is supplied externally



Efabless Development Board Jumper Labels[1]

Hardware Test

Test Procedure:

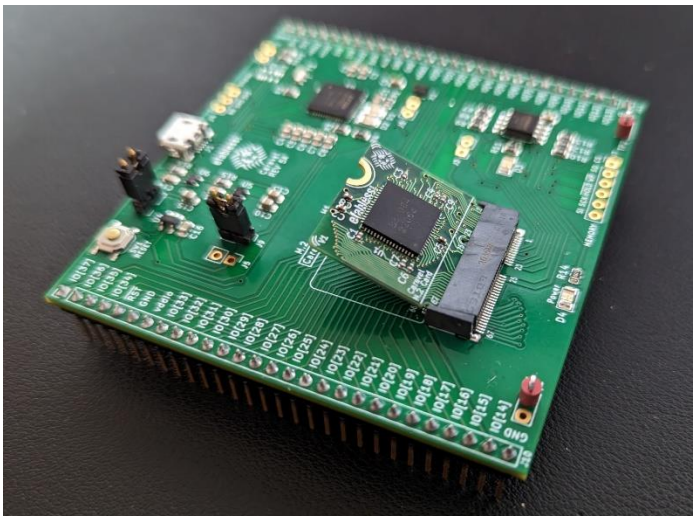
1. Set multimeter to Continuity mode and ensure there are no short circuits between 3.3V and GND

2. Set multimeter to Continuity mode and ensure there are no short circuits between 1.8V and GND
3. Insert USB to Micro-B connector into both the users laptop and the J1 connector on the top of the Evaluation board to supply power
4. Set multimeter to DC Voltage mode and verify J8 reads 1.8V between J8 and GND
5. Set multimeter to DC Voltage mode and verify J9 reads 3.3V between J9 and GND

Software Setup

Setup Procedure:

1. Install the Caravel Breakout Board onto the Evaluation Board
    a. Remove the screw (M1) to add/remove the breakout board module
    b. Insert the Caravel Breakout Board into a ZIF connector at a 45 degree angle
    c. Press the Caravel Breakout Board down so that it is flush with the Development Board, and turn the screw (M1) down
2. Connect the Master SPI IO pins to J11
    a. SPI Master Clock: IO[5]
    b. Slave Select Reset: IO[6]
    c. Master Out Slave In (MOSI): IO[7]
    d. Master In Slave Out (MISO): IO[8]
3. Connect the DSP Accelerator IO pins to J11
    a. DSP Weight ACK: IO[9]
    b. DSP Data ACK: IO[10]
    c. DSP Convolution ACK: IO[11]
4. Insert USB to Micro-B connector into both the users laptop and the J1 connector on the top of the Evaluation board



Efabless Development Board Installation[1]

Software Test

We have not yet received a Nucleo to test with, so these instructions will require some debugging with reference to the Nucleo board and its interface to the management SoC.

Use https://github.com/efabless/caravel_board/tree/main/firmware/chipignite#readme to program the board. This repository contains makefiles and instructions for interfacing with the board.

1. Use the template above to run the basic blink test. Follow the README instructions, running `make flash` in the blink folder. Make sure the GPIOs blink as expected from that program before continuing.
2. Copy the blink program and rename it to c_test. Copy the c_test.c file from verilog/dv/c_test in the project repository. This file contains example code for every module and examples of using every module from the management SoC. You will need to edit the makefile and remove isr.c from the list of source files as c_test provides its own interrupt handler.
3. Use the Nucleo programming guide to use the clkmux with the EXTCLK input pin to try clocking the wishbone bus externally. Verify using a simple program in C on the management SoC.
4. Use the Nucleo programming guide to interface with the user project via SPI to the wishbone test, then the DSP module.
5. Verify the clock frequency of the user project, and increase until the DSP module fails to function to determine the physical maximum clock frequency.
6. Reference the Design Document and experiment with any other experimental questions you may have.

Module Definitions

For a full description of how each evaluated module should function, please reference Appendix 8.4.4. For a detailed description of the Backdoor SPI interface, please reference Appendix 8.4.7.

Reference

[1] Efabless. (n.d.). Efabless/caravel_board. GitHub. Retrieved April 17, 2023, from https://github.com/efabless/caravel_board