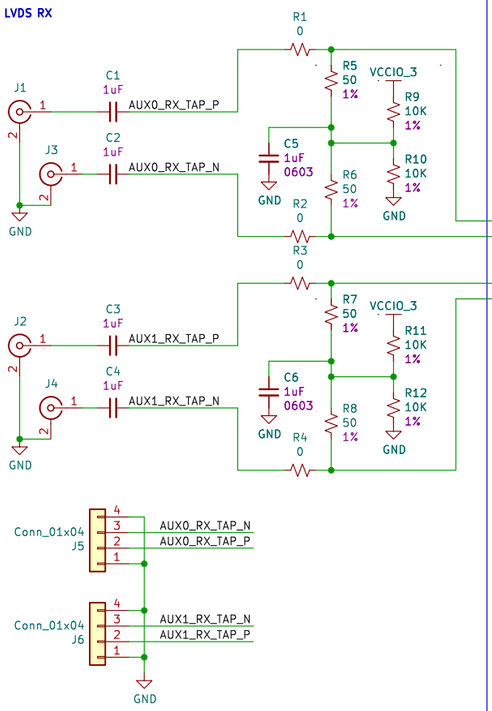
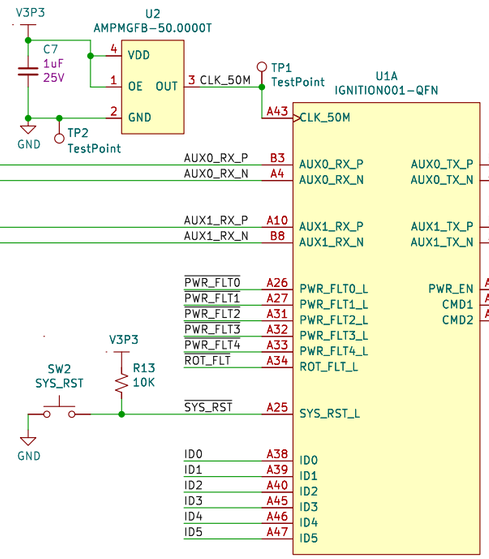
Ignition Target

LVDS RX



The LVDS transmitter pair is implemented as per Lattice FPGA-TN-1253 using PIO pin pairs in Bank 3. The ICE40 inputs expect a proper LVDS signal so the inputs are biased after the AC-coupling cap. Resistor and capacitor values can be tuned where required.

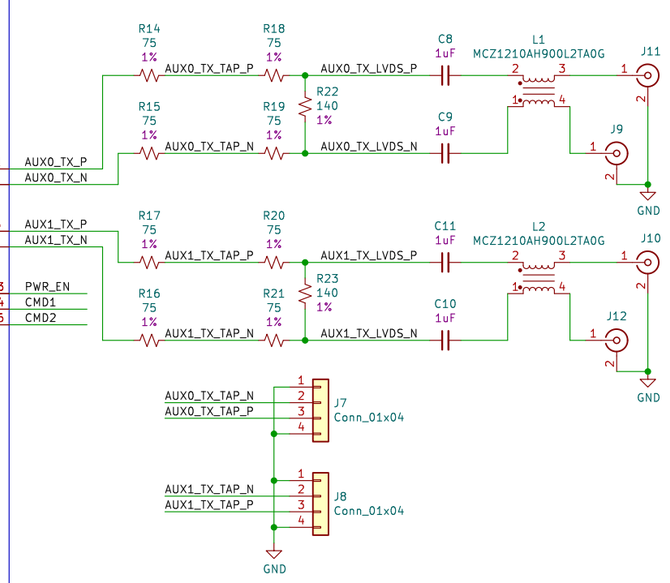
An optional header footprint/via is placed near the SMA connector to allow for alternate prototyping, using either single-ended or differential signaling. The zero ohm resistor footprint can be used for termination or slew limiting resistors if desired.



TODO:

- Check TX/RX cap values againt bit rate

LVDS TX



The LVDS transmitter pair is implemented as per Lattice FPGA-TN-1253 using PIO pin pairs in Bank 3. The resistor values above were derived using the equations on p. 4 and the following assumptions:

Z0 = 50 ohm
VCCIO = 2.5V
V_OD = 0.35V
R_OUTPUT = 30 ohm

R_P = 2 * ((Z0 * VCCIO) / (VCCIO - (2 * V_OD)))
    = 2 * (165 / 1.8)
    = 139 ohm

R_S = ((Z0 * R_P / 2) / ((R_P / 2) - Z0) - R_OUTPUT
    = (3472 / 19) - 30
    = 149 ohm

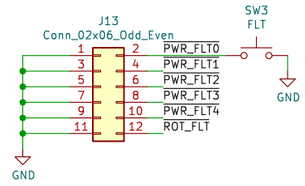
The series resistor is broken into two pieces of 75 ohm each. The intend here is that one pin of a 100 mil header/ footprint is inserted between the two resistors. If done using a tight layout this via should add minimal disruption at the edge rates of these transmitters.

Inserting the via would allow for IO pin to be reused for alternative prototyping by not fitting the second series resistor, parallel resistor and SMA connector, while using the first resistor footprint as slew limiting resistor or for series termination.

One possible application of this alternative scheme is to allow the Ignition protocol to be carried using single ended LVCMOS signaling at 3.3V between this broad and an ECP5 dev board without requiring SMA connectors for the link partner. This would simplify initial prototyping work.

The AC couping capacitors near the SMA connector are optional in case one wants to experiment. The layout should allow for some copper nearby connected to ground so experiments with a choke are possible. If not in use 0 ohm resistors should be fitted.

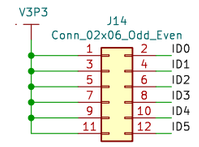
Fault Pin Straps



Fault pin straps allow setting fixed fault values. The switch attached to PWR_FLT0 provides a convenient way to cause fault interrupts during integration tests. The header allows reuse of these pins for alternative prototyping.

The FPGA should use programmable pull-up/pull-down on these poins if required.

ID Straps



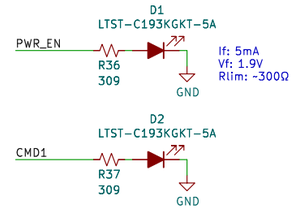
ID straps allow setting the desired value. The header allows reuse of these pins for alternative prototyping.

See RFD 142 for currently allocated ID values. The FPGA should use programmable pull-up/pull-down on these pins if required.

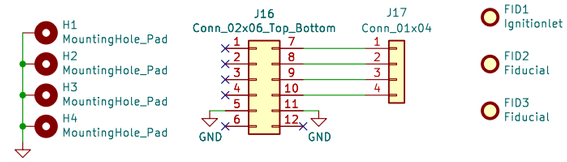
CMD Bits Header

Command bits are exposed on a header for easy probing and/or alternative prototyping.

CMDx LEDs



Mechanical



The board assumes the width of two standard 0.80" PMOD boards. J16 is intended for additional mechanical stability, but is connected to J17 as a Type 1 interface, providing GPIOs for prototyping which for example can be wired to CDONE or FLASH_RESET when connected to an appropriate host board.

Power

Power

File: ignitionlet-power.kicad_sch

Config

Programming & Configuration

File: ignitionlet-config.kicad_sch

Oxide Computer Co.

Sheet: /
File: ignitionlet.kicad_sch

Title: Ignition Target

Size: A3	Date: 2021-07-21	**Rev: 2**
KiCad E.D.A. kicad 7.0.0		Id: 1/3

Fake Bus Bar

J101
Barrel_Jack_MountingPin

J102
Conn_01x02

VIN_12V

PWR_FLAG

GND

The Fake Bus Bar can take 5–12V from the barrel jack or J2 can be used to injecting 12V from the Gimletlet using short Dupont wires. This allows both boards to be powered by a single supply.

I_hold: 100mA
I_trip: 300mA

V_max: 30V
I_max: 10A

U101
LT3013EFE#PBF

C101 10uF 25V
C102 10uF 25V
R102 100K 1%
R103 1K 1%
R104 147K 1%
C106 10uF
TP101
C108 10uF 10V
C111 10uF 10V
C113 10uF 10V
V1P2

Vout nom: 1.2V
Imax: 250mA

PGOOD_V1P2

The number of input/output capacitors was copied from the Gimlet implementation, which assumes a 54V input voltage. Given a smaller difference between VIN of this board and the output less may be required. The footprints are here to adjust/fix if needed.

U102
LT3013EFE#PBF

V3P3
C103 10uF 25V
C104 10uF 25V
R101 100K 1%
R105 147K 1%
R106 88K7 1%
C107 10uF
TP102
C109 10uF 10V
C112 10uF 10V
C114 10uF 10V

Vout nom: 3.3V
Imax: 250mA

PGOOD_V1P2
PGOOD_V3P3

TP104 TP105

Keystone 5015s for easier probe access.

U103
TPS7A0525PDBVR

C105 1uF 25V
C110 1uF 25V
TP103
V2P5

Vout nom: 2.5V
Imax: 200mA

PGOOD_V3P3

U1B
IGNITION001-QFN

V1P2
C115 1uF 25V
C116 1uF 25V
C118 1uF 25V
C121 1uF 25V

VCC_1V2 (A7, A15, A28, B28)

V3P3
C117 1uF 25V
C119 1uF 25V
C122 1uF 25V

VCCIO_0_3V3 (A42)
VCCIO_1_3V3 (B25)
VCCIO_2_3V3 (A17)

JP3 allows VCCIO_3 to be configurable as 2.5V/3.3V.

If configured as 3.3V the LVDS TX resistors need to be adjusted for compliant diff swing.

J103
Conn_01x03

2.5V: pins 1–2
3.3V: pins 2–3

VCCIO_3
V2P5

C120 1uF 25V
C123 1uF 25V

PWR_FLAG

VCCIO_3_2V5 (B6)
VPP_2V5 (A36)
VPP_FAST (A37)
PAD (A6)
GND (A18, A30, B33)

Sheet: /Power/		
File: ignitionlet-power.kicad_sch		
Title: Power		
Size: A3	Date: 2021-07-21	**Rev: 2**
KiCad E.D.A. kicad 7.0.0		Id: 2/3

PMOD 2A Peripheral (optional)

Intended to allow a Gimletlet to program either SPI flash or the FPGA.

J201
PMOD_2A_PERIPHERAL

Ignition Program/Debug Header

Intended to mate with Adafruit FT232H + adapter to allow software compatibility with both Lattice and open source programming tools.

The pinout is compatible with a straight PMOD 2A connector to allow for alternative adapters if desired.

Whatever connects to this header is expected to bring slew limiting resistors.

J202
IGNITION_TARGET_HDR

Header/Jumpers/Test Points

Intended for hacking and/or forced reset/write protect.

SW201
FPGA_RST

SW202
FLASH_RST

J203
Conn_02x05_Odd_Even

Fit one of two SPI flash options.

U5
MT25QL128ABB8E12-0AUT

U6
AT25PE80-SSHN-B

U8
SN74AUP1G08DBVR

U7
SN74CBTLV3257RSV

U1C
IGNITION001-QFN

FPGA SPI Mode MUX

The SDI/SDO signals of the FPGA SPI port switch direction depending on whether the FPGA acts as controller or peripheral. U7 handles this reversal of the signals allowing the FPGA to be programmed by reading from the SPI flash, or by another controller. The logic is as follows:

FPGA_MODE	1A	2A
H: SPI Master (default)	SPI_MOSI	SPI_MISO
L: SPI Slave	SPI_MISO	SPI_MOSI

ColdBoot Header/Test Points

If the SPI flash is programmed with an appropriate ColdBoot applet and the FPGA boots in SPI Master mode, these headers can be used to select one of four bitstreams as defined in the applet.

See Lattice FPGA-TN-02001-3.2 for details.

J204
Conn_01x03

J205
Conn_01x03

Operating Modes

– FPGA as SPI Master (default)

The default operating mode for this board is with the FPGA acting as SPI master. Without anything driving SPI_SS this signal is pulled high. On init (after PoR or asserting CRESET) the FPGA will sample this pin. With the pin pulled high it will resume its init sequence as SPI master. Consequently it will then assert SPI_SS and drive SPI_SCK, allowing it to read a bitstream from SPI flash and enter the user application.

– Program the SPI flash from IGNITION_TARGET_HDR

The second mode is to program the SPI flash via the IGNITION_TARGET_HDR using an FTDI USB programmer. The programmer will assert both SPI_RESET and SPI_SS, causing the FPGA to go/stay in reset while selecting the SPI flash as peripheral. The programmer then writes to SPI flash as normal.

– Program the FPGA SRAM from IGNITION_TARGET_HDR

The FTDI programmer can program the FPGA SRAM (or NVCM) directly if the user installs a jumper on the FPGA_MODE signal. This will cause the SPI flash to remain in reset and ignore any SPI traffic while mux beteen the header and FPGA SPI port will switch the SDO/SDI lines appropriately. The programmer then asserts SPI_SS while toggling SPI_RESET. This causes the FPGA to (re-)initialize, sampling the SS pin, and initialize as SPI peripheral instead of master when it finds this signal low. The programmer then writes to the FPGA as per Lattice FPGA-TN-02001-3.2.

Holding the flash in reset can potentially be automated by connecting FLASH_RESET to an additional GPIO pin on the programming adapter and modifying the (open source) programming software to assert this pin during the programming cycle. This may already be supported in software, but is not tested.

– Program the FPGA SRAM from PMOD_2A

In the same way as stated above the PMOD header can be used to program the FPGA SRAM from an attached Gimletlet using software running on the SP. This mode assumes the SPI flash is disabled during programming, either using the described FLASH_RESET jumper, or by connecting the FLASH_RESET signal to an SP GPIO using a Dupont wire, allowing flash reset to happen under software control.

Sheet: /Config/
File: ignitionlet-config.kicad_sch

Title: Programming & Configuration

Size: A3	Date: 2021-07-21	**Rev: 2**
KiCad E.D.A. kicad 7.0.0		Id: 3/3