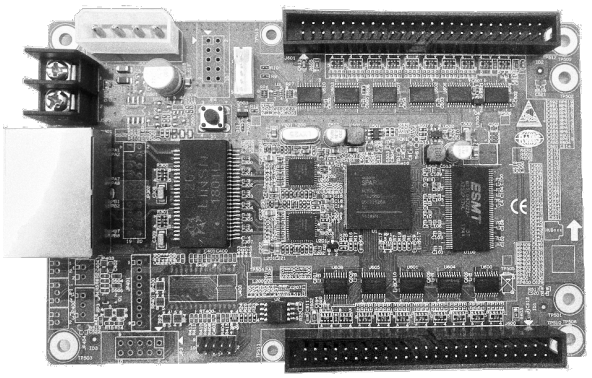
CHUBBY75

github.com/q3k/chubby75

Turn this boring old LED panel controller into an **FPGA devboard!**



Ever needed a **cheap FPGA board** to just throw into a project somewhere? Are you bothered by the fact that the most GPIO you usually get is a measly Arduino header? Look no further!

Chubby75 is an effort to reverse engineer an LED panel controller board (identified RV901T, available on Aliexpress for around **$20**), that just happens to contain:

- a **Spartan 6** LX15 **FPGA**
- 2x **Gigabit Ethernet** with PHYs
- 8MBytes of **SDRAM**
- over **70 5V GPIOs**

As seen on IRC!

We provide extensive documentation to turn this board into an FPGA development board for education and research. And, given enough effort, you might even be able to write a proper open source stack for controlling LED panels!

We also provide support for **Migen/MiSoC/LiteX**, so you can define your digital logic in Python. To **blink an LED** run the following Python code in the Chubby75 git checkout:

```
from migen import *
class Top(Module):
  def __init__(self, platform):
    # Single clock domain from external
    # oscillator.
    osc = platform.request('clk25')
    self.clock_domains.cd_sys = \
      ClockDomain()
    self.comb += self.cd_sys.clk.eq(osc)
    # Blink that LED.
    led = platform.request('user_led')
    counter = Signal(max=25000000)
    self.sync += \
      If(counter == 0,
        counter.eq(25000000),
        led.eq(~led),
      ).Else(
        counter.eq(counter-1))
      )

# Instantiate and build for RV901T.
from platform import Platform
p = Platform()
t = Top(p)
p.build(t)
# Program over JTAG with xc3sprog and a
# Xilinx Platform Cable.
import migen.build.xilinx.programmer \
  as prgs
prog = prgs.XC3SProg('xpc')
prog.load_bitstream('build/top.bit')
```

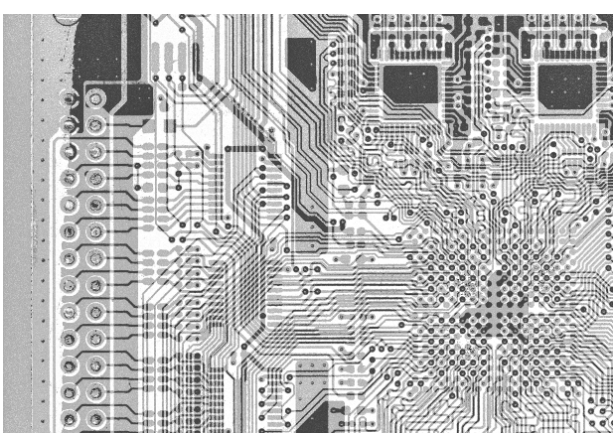
Don't forget! Using LiteX allows you to quickly integrate support for **Ethernet** (via LiteEth), and **SDRAM** (via LiteDRAM). And, if you want a soft core, this FPGA will easily fit a Lattice LM32 and a bunch of picorv32 **RISC-V** cores! In the repository, you'll find a working example of SDRAM + LM32 running C code.

Right now you will still need Xilinx's ISE suite to develop for this board. However, there are efforts to bring an open source toolchain to Spartan 6 FPGAs, so keep your eyes peeled!

You might be wondering - how do you **document a 4 layer PCB** and get a full pinout of all the connectors?

We started by finding **JTAG** on the board. Thankfully, it's marked on the silkscreen, so we just had to scrape the soldermask off and solder to it. With that, we could start running our own bitstreams on the board. But how do we even know where a clock or an LED is?

We ended up taking a **brute force approach**. One board was fully **depopulated, sanded**, and **photos** were taken of every layer. This allowed us to understand some things about how the PHYs and SDRAM are connected, and how to control the I/O buffers on the board. We post processed the photos of the layers in **GIMP** and then layered them in **Inkscape**, so that we could trace and label things as we discovered them.



Once we had a general idea of how things worked, we wrote a little piece of Migen code to take all unknown pads of the FPGA and make them output their identifier as a repeated ASCII string via UART. Using this, we could just **probe** the two large connectors on board with a USB to UART adapter to immediately know what pin of the FPGA drove what pin of the connector.

Chubby75 is a collaboration brought to you by:
Niklas Fauth, Jan Henrik, q3k, carrotIndustries, enjoydigital, jeanthom, informatic and many others.