# Reverse engineering the Commodore 128

Johan Grip, VCF West 2022

# Or: How to find all the rabbit holes and jump into them

#### Who am I?

- Computer and electronics nerd
- Self taught
- Working in enterprise IT since forever

https://c128.se

https://github.com/jgrip/

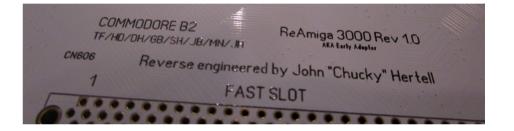
### Background

- I still have my first computer
- It was broken when I unpacked it
- The repair snowballed a bit...



#### Inspirations

John "Chucky" Hertell



• Rob Taylor



#### Goals

- Platform preservation
- Integrate factory fixes
- Small quality of life improvements

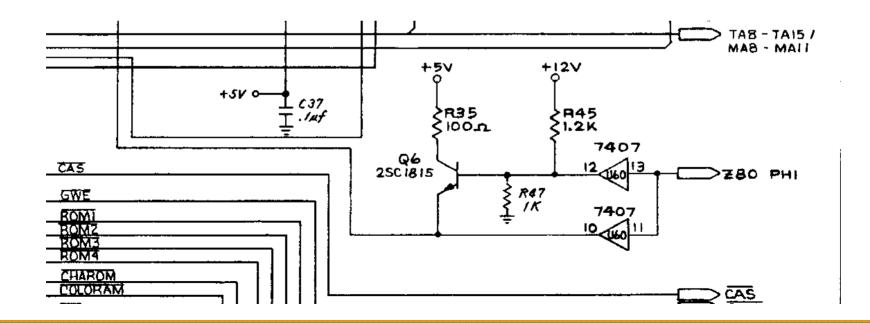
#### Software

- Sprint Layout
- Open source preferred
- KiCad is getting real good

Ki File View Tools Preferences Help oiect Files C128.kicad pro Schematic Editor Edit the project schematic C128-backups Legacy Symbol Editor C128-rescue.kicad sym Edit global and/or project schematic symbol libraries C128.kicad\_pcb C **PCB Editor** Edit the project PCB design C128.kicad\_sym Footprint Editor Edit global and/or project PCB footprint libraries **Gerber Viewer** Preview Gerber files **Image Converter** Convert bitmap images to schematic symbols or PCB footprints Calculator Tools Show tools for calculating resistance, current capacity, etc. A. **Drawing Sheet Editor** Edit drawing sheet borders and title blocks for use in schematics and PCB designs Plugin and Content Manager Manage downloadable packages from KiCad and 3rd party repositories

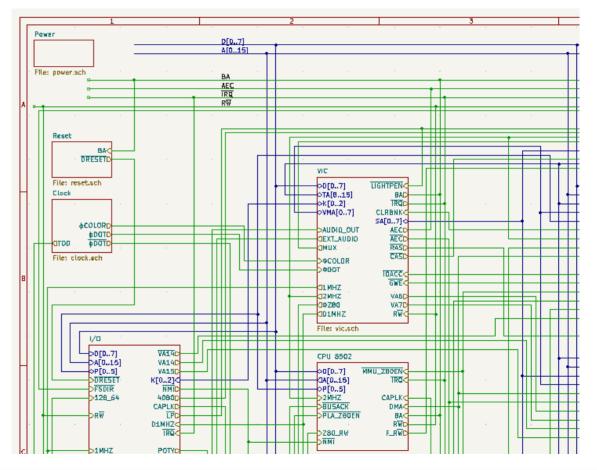
#### Schematic capture

- C= schematic very accurate
- Did not attempt to preserve layout in KiCad



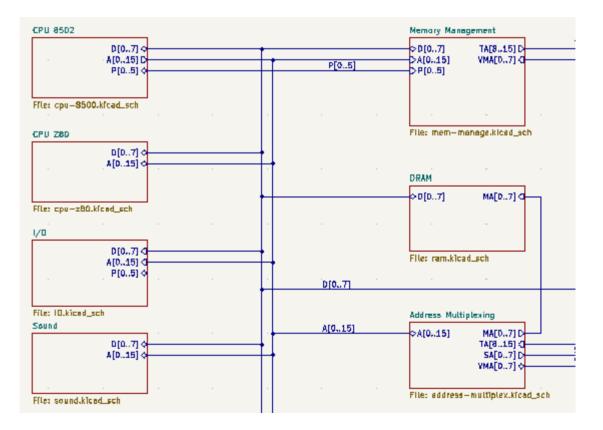
#### Schematic capture

• First attempt



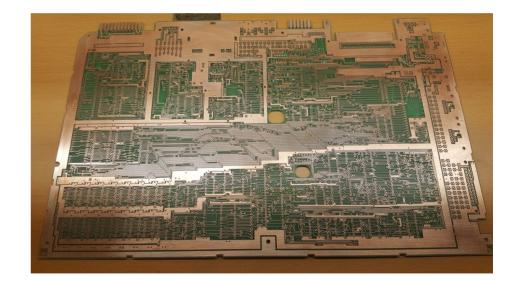
#### Schematic capture

Better now



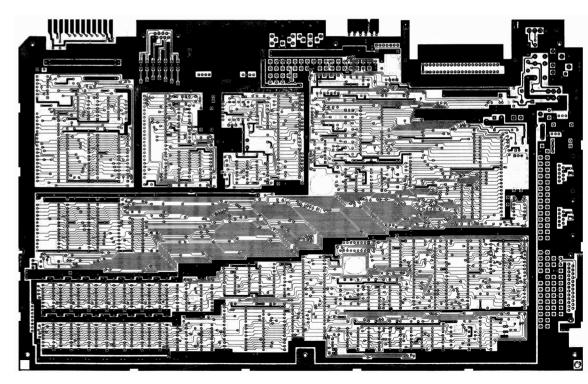
#### Prepare PCB

- Desolder all the things
- Sanded down solder mask
- Flatbed scanner



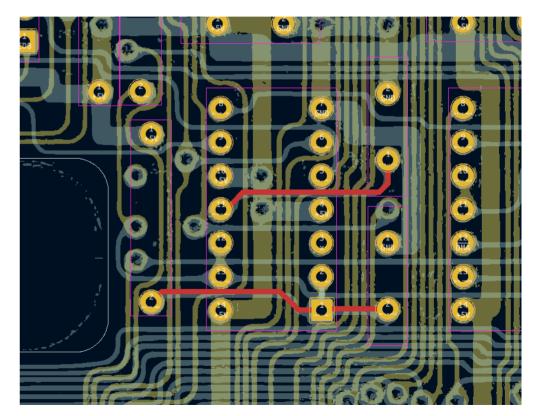
#### Prepare for KiCad

- Turn monochrome
- Align rotation
- Convert to component
- Add to PCB and align

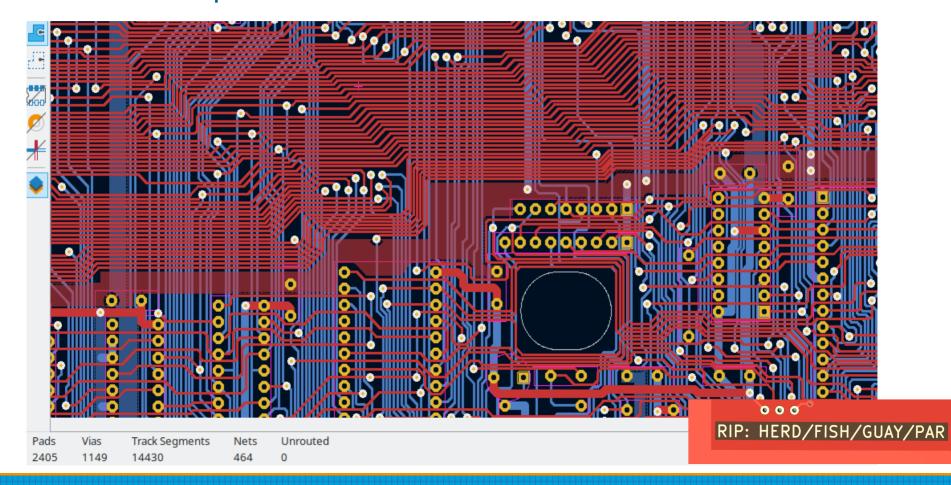


#### KiCad PCB

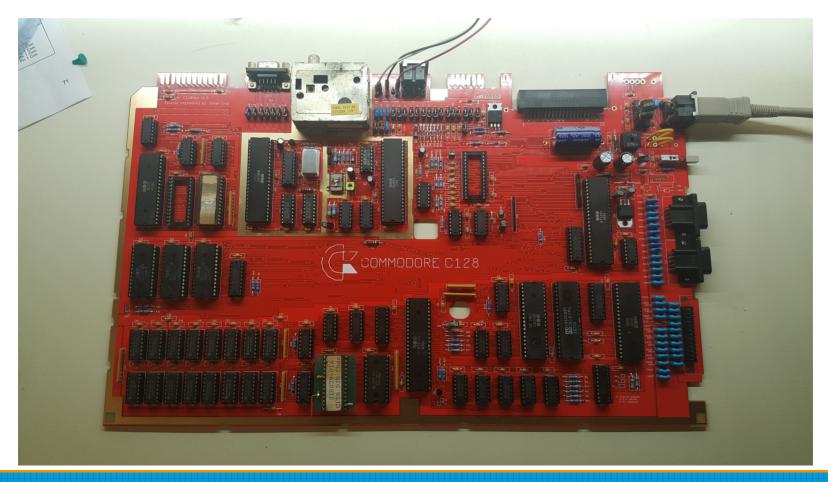
- Import scan as component
- Place components
- Draw traces
- Bring patience



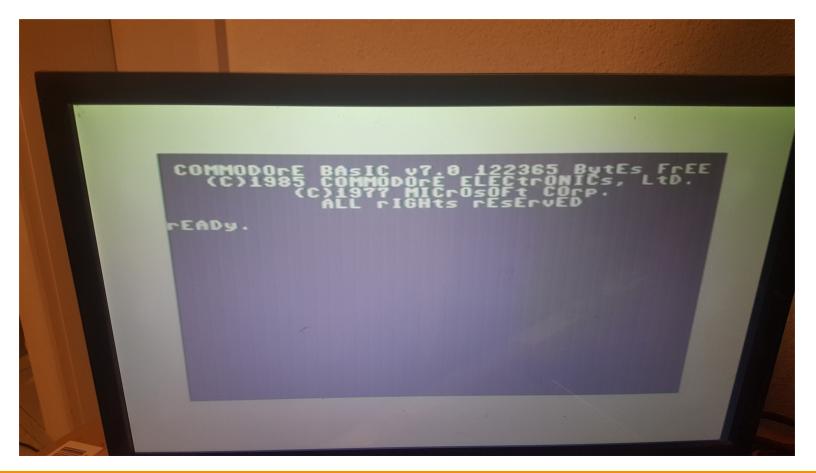
#### Lots of patience...



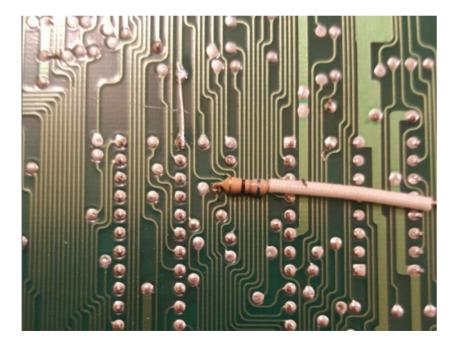
#### First prototype board



#### Worked on first try...







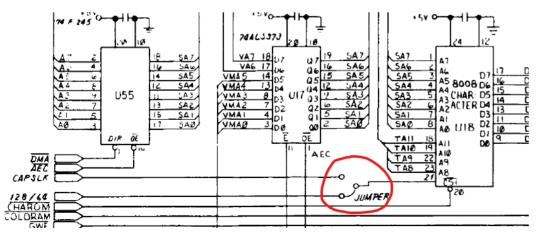


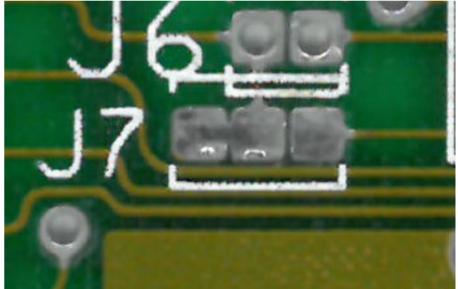
#### Fix: J6

- Used to select size of Kernal ROM
- Should be A15
- Was connected to A14

#### Fix: J7

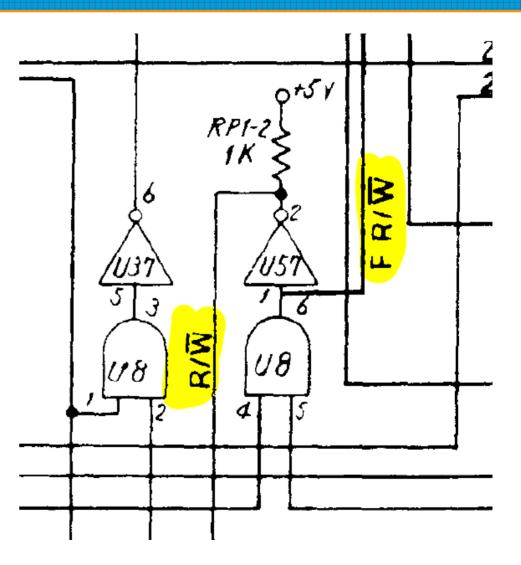
- Use to select source of font selection
- Not functional
- Bodge wire on international





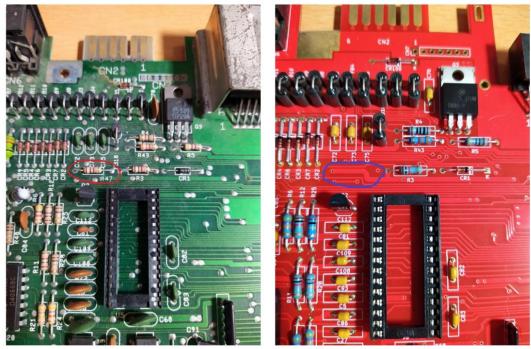
### VDC R/W

- Buffer gate on R/W
- VDC first
- Introduced on rev.7



# One bug

- Missing pullup
- Dotclock on expansion



R47 Commodore

R47 Neo

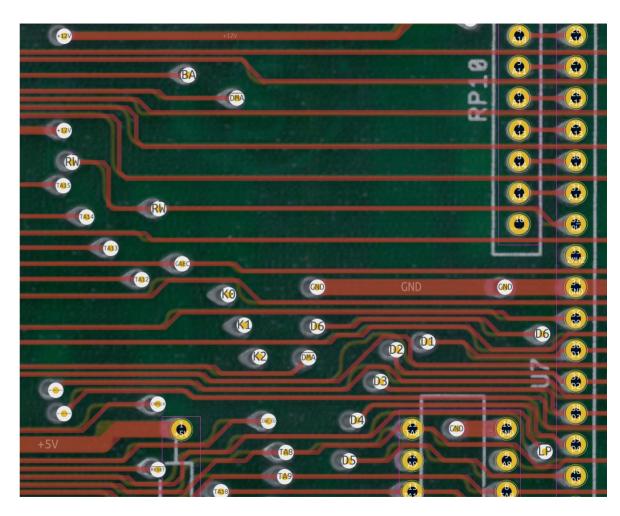
#### Rev.4 soon

- Lots of quality of life
- Pin headers
- RF modulator
- Minimal jailbars



#### KiCad 7

- Early next year
- Bitmap backgrounds
- In nightlies now



#### Other PCB work

- C128CR board
- On github



#### Interesting curiosities

C128D CES prototype



• Drean 128



### Keyboard

- Mechanical kbd PCB
- Alps/MX hybrid
- Key latches
- On github

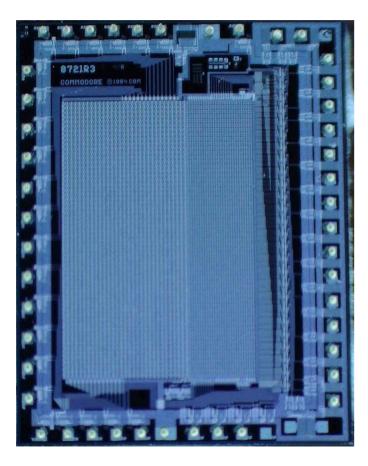


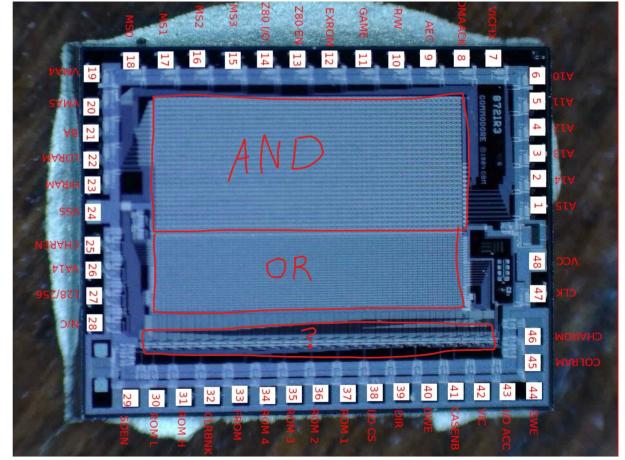
# So, I bought a microscope

- AmScope SM-4T for working on surface mount
- Got curious about ICs
- Nasty Chemicals
- Chips a la Antoine



#### First die shot



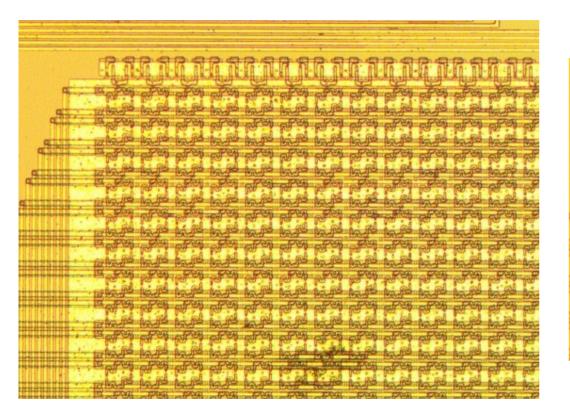


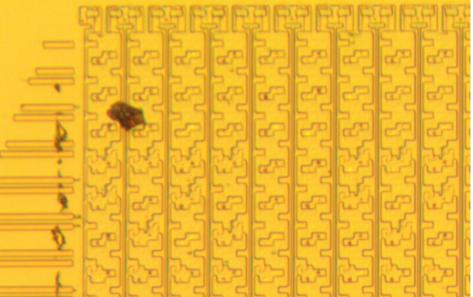
# So, I bought a microscope

- AmScope ME580-T
- Home brew motorization
- RPi HQ Camera

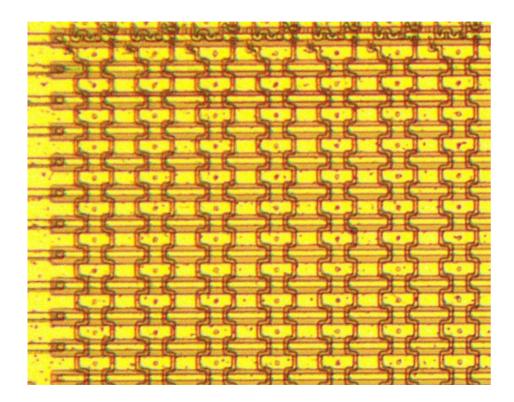


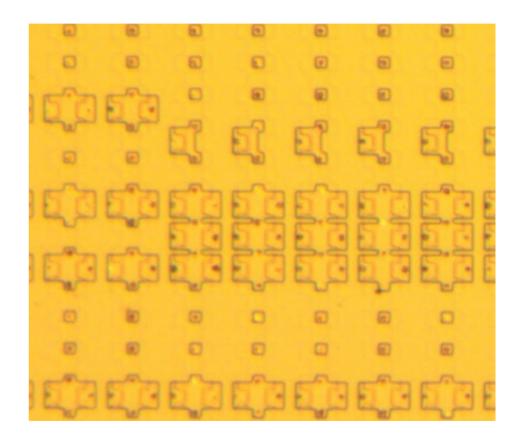
#### AND matrix



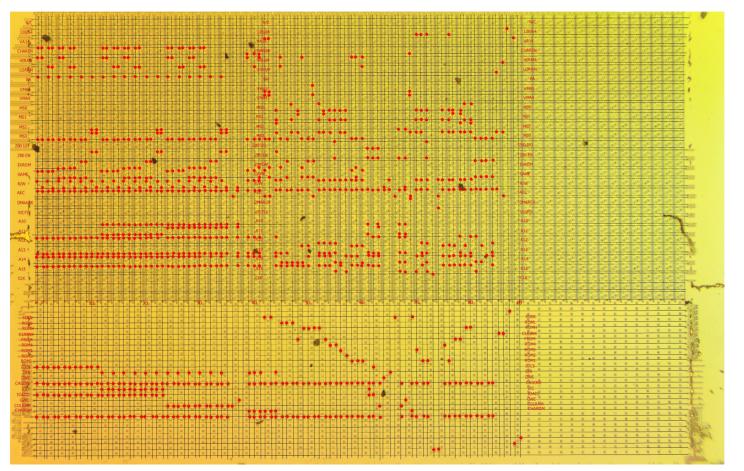


#### OR matrix



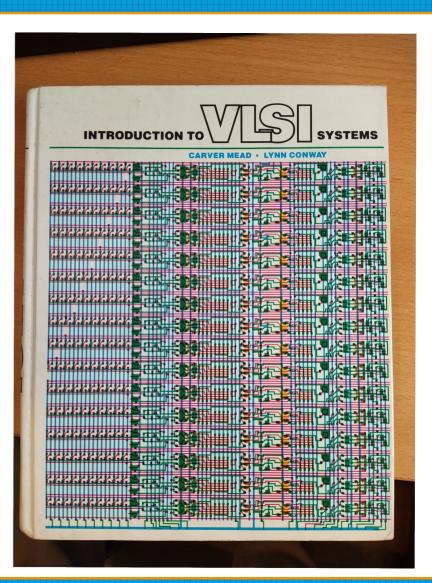


#### Full decode



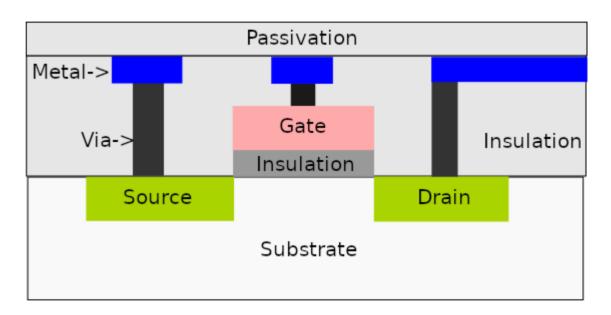
#### The book

- Conway and Mead
- NMOS

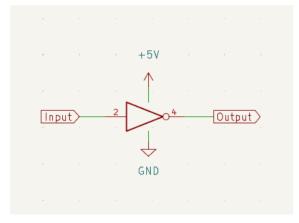


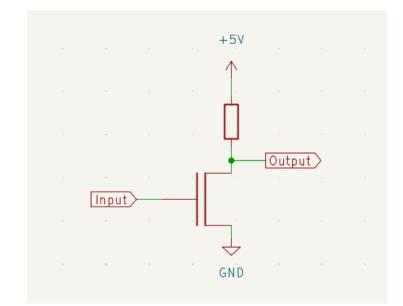
#### MOS/CSG NMOS

- Single metal layer
- Feature size
- NMOS/HMOS/HMOS-II

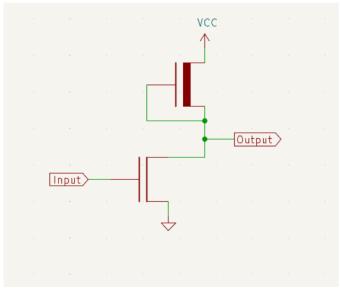


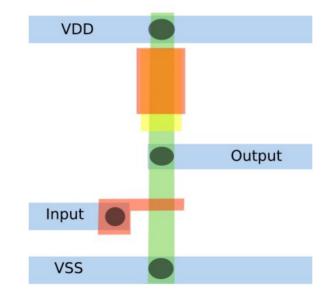
# NMOS logic inverter

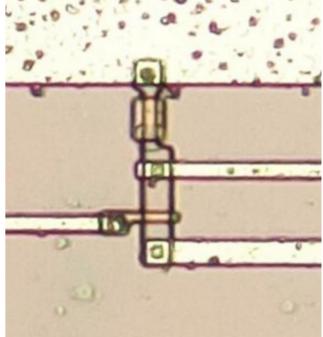




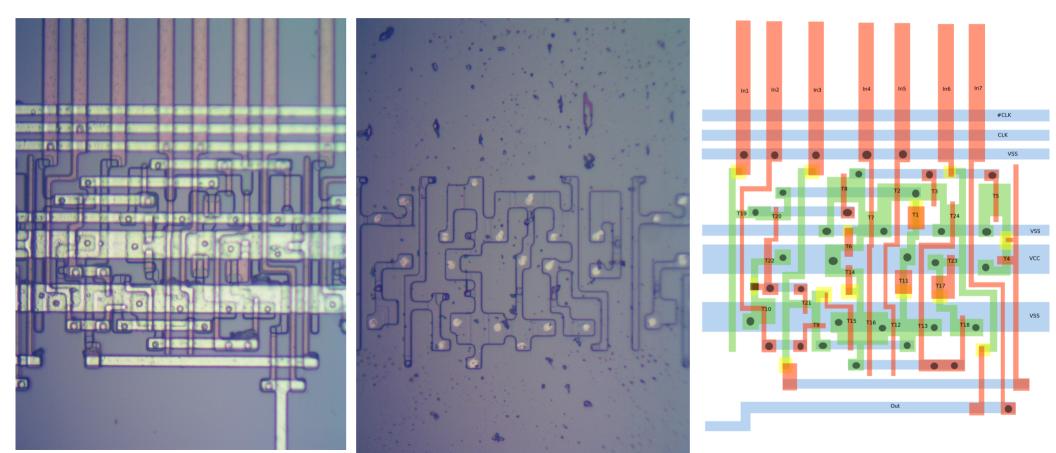
# NMOS logic inverter



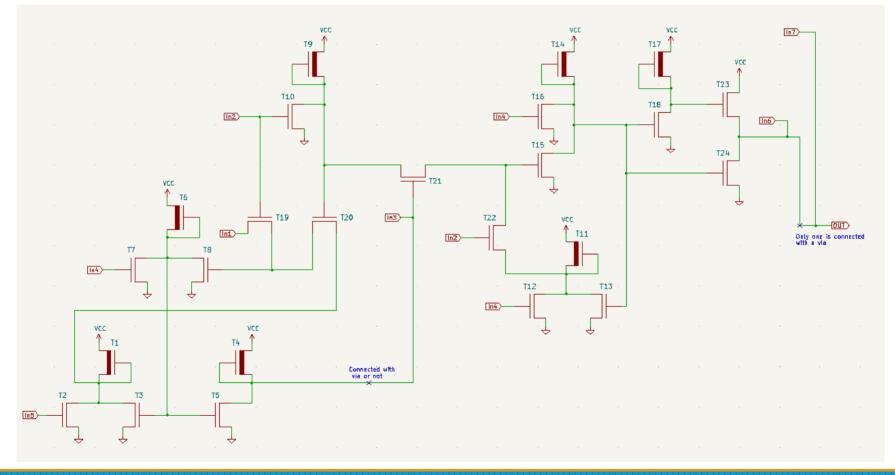




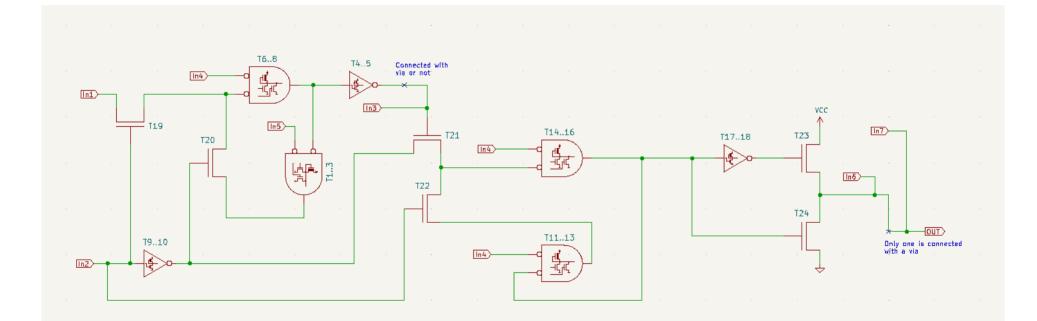
# Outputs



#### Outputs



#### Outputs

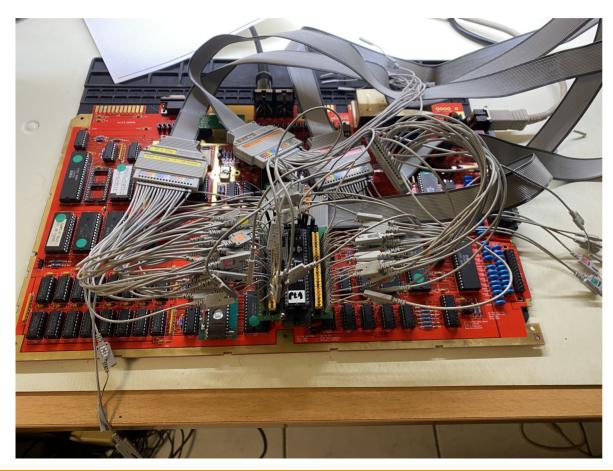


# Verilog

2.04	assign poz - : mss a exion a : yane a aco a ais a : ai+,
255	assign p83 = !ms3 & exrom & !game & aec & a14;
256	assign p84 = !ms3 & exrom & !game & aec & !a12 & !a13 & a14 & a15;
257	
258	assign p85 = !loram & ms3 & aec;
259	assign p86 = !hiram & ms3 & !aec;
260	
261	/* outputs */
262	
263	assign sden = p42    p43    p66    p69;
264	<pre>assign rom1 = p45    p46    p47;</pre>
265	assign romh = p49    p50    p51    p52    p79    p80;
266	<pre>assign clrbnk = p85    p86;</pre>
267	assign from = p48    p53    p77    p78;
268	<pre>assign rom4 = p54    p55    p75;</pre>
269	assign rom3 = p56    p70;
270	assign rom2 = p57;
074	accide rem1 - nE0 11 nE0 11 nE0 11 n71 11 n72 11 n76;

#### https://github.com/jgrip/c128-verilog

### Some debugging later





# So, I bought a microscope

- Olympus BH2 w/ UMA
- Home built motirization
- DSLR camera



#### Future plans

- Finish re-implementing the MMU
- Reverse engineer the VDC
- Built a complete C128 with only new parts

Questions?