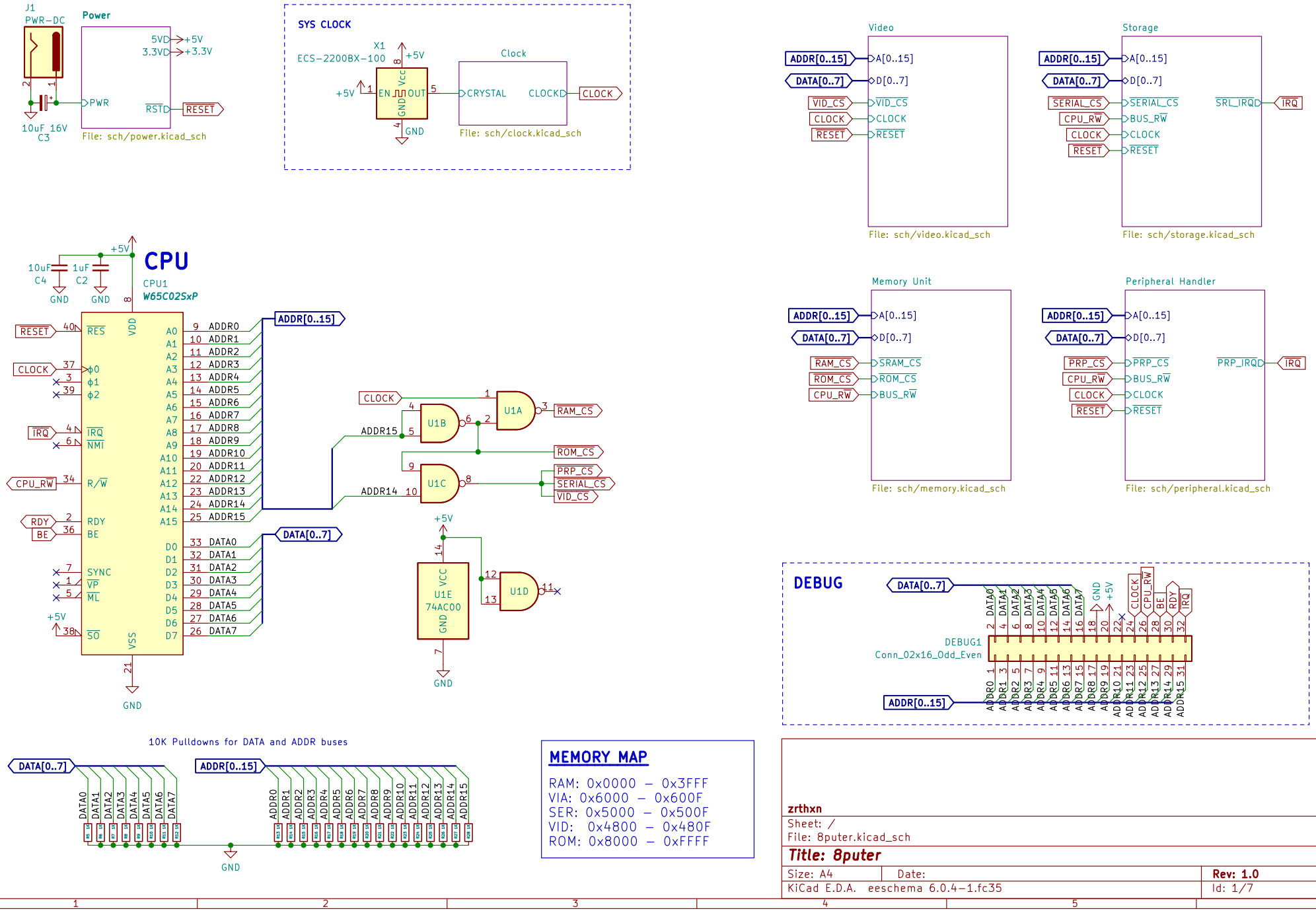
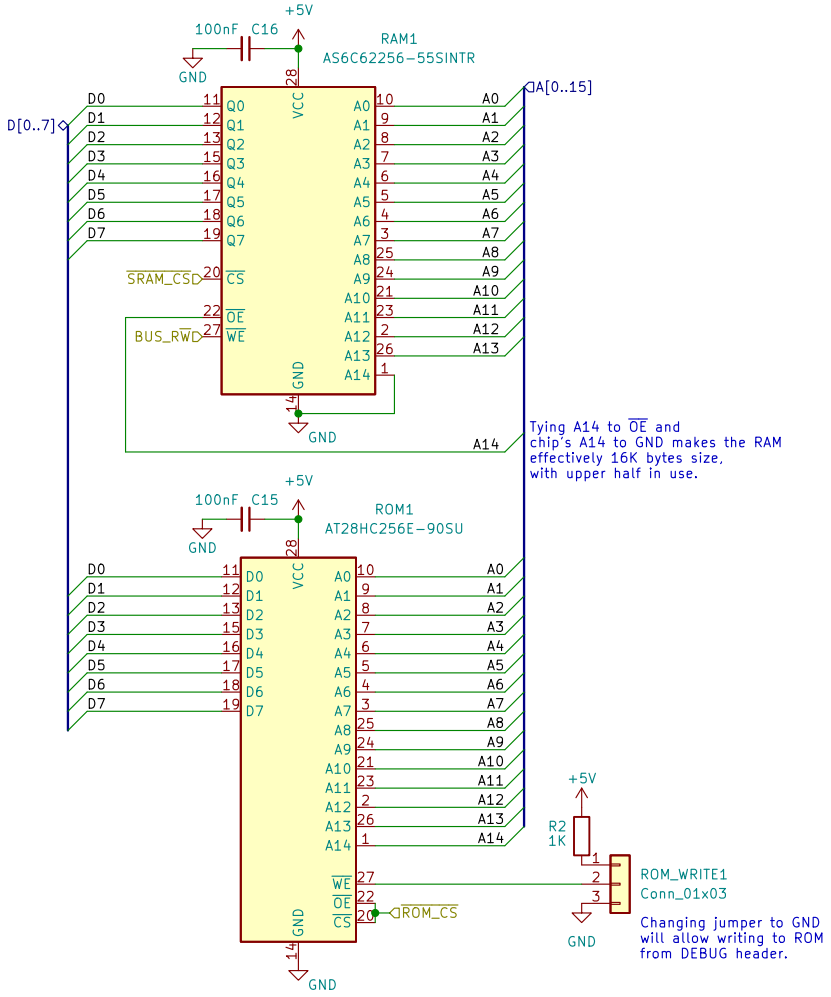

J1
PWR-DC
Power
5VD
+5V
3.3VD
+3.3V
PWR
RST
RESET
10uF 16V
C3
File: sch/power.kicad_sch
SYS CLOCK
X1
ECS-2200BX-100
+5V
Vcc
EN
OUT
GND
Clock
CRYSTAL
CLOCKD
CLOCK
File: sch/clock.kicad_sch
Video
ADDR[0..15]
DATA[0..7]
VID_CS
CLOCK
RESET
A[0..15]
D[0..7]
File: sch/video.kicad_sch
Storage
SERIAL_CS
CPU_RW
BUS_RW
SRL_IRQ
IRQ
File: sch/storage.kicad_sch
CPU
CPU1
W65C02SxP
10uF
C4
1uF
C2
RES
VDD
ϕ0
ϕ1
ϕ2
IRQ
NMI
R/W
RDY
BE
SYNC
VP
ML
SO
VSS
ADDR0
ADDR1
ADDR2
ADDR3
ADDR4
ADDR5
ADDR6
ADDR7
ADDR8
ADDR9
ADDR10
ADDR11
ADDR12
ADDR13
ADDR14
ADDR15
DATA0
DATA1
DATA2
DATA3
DATA4
DATA5
DATA6
DATA7
U1A
U1B
U1C
U1D
U1E
74AC00
RAM_CS
ROM_CS
PRP_CS
SERIAL_CS
VID_CS
Memory Unit
SRAM_CS
File: sch/memory.kicad_sch
Peripheral Handler
PRP_IRQ
File: sch/peripheral.kicad_sch
DEBUG
DEBUG1
Conn_02x16_Odd_Even
10K Pulldowns for DATA and ADDR buses
MEMORY MAP
RAM: 0x0000 – 0x3FFF
VIA: 0x6000 – 0x600F
SER: 0x5000 – 0x500F
VID: 0x4800 – 0x480F
ROM: 0x8000 – 0xFFFF
zrthxn
Sheet: /
File: 8puter.kicad_sch
Title: 8puter
Size: A4
Date:
Rev: 1.0
KiCad E.D.A. eeschema 6.0.4-1.fc35
Id: 1/7

System Memory

RAM: 0x0000 – 0x3FFF
ROM: 0x8000 – 0xFFFF



zrthxn		
Sheet: /Memory Unit/		
File: memory.kicad_sch		
Title: 8puter		
Size: A4	Date:	**Rev: 1.0**
KiCad E.D.A. eeschema 6.0.4-1.fc35		Id: 2/7

Peripheral Interface

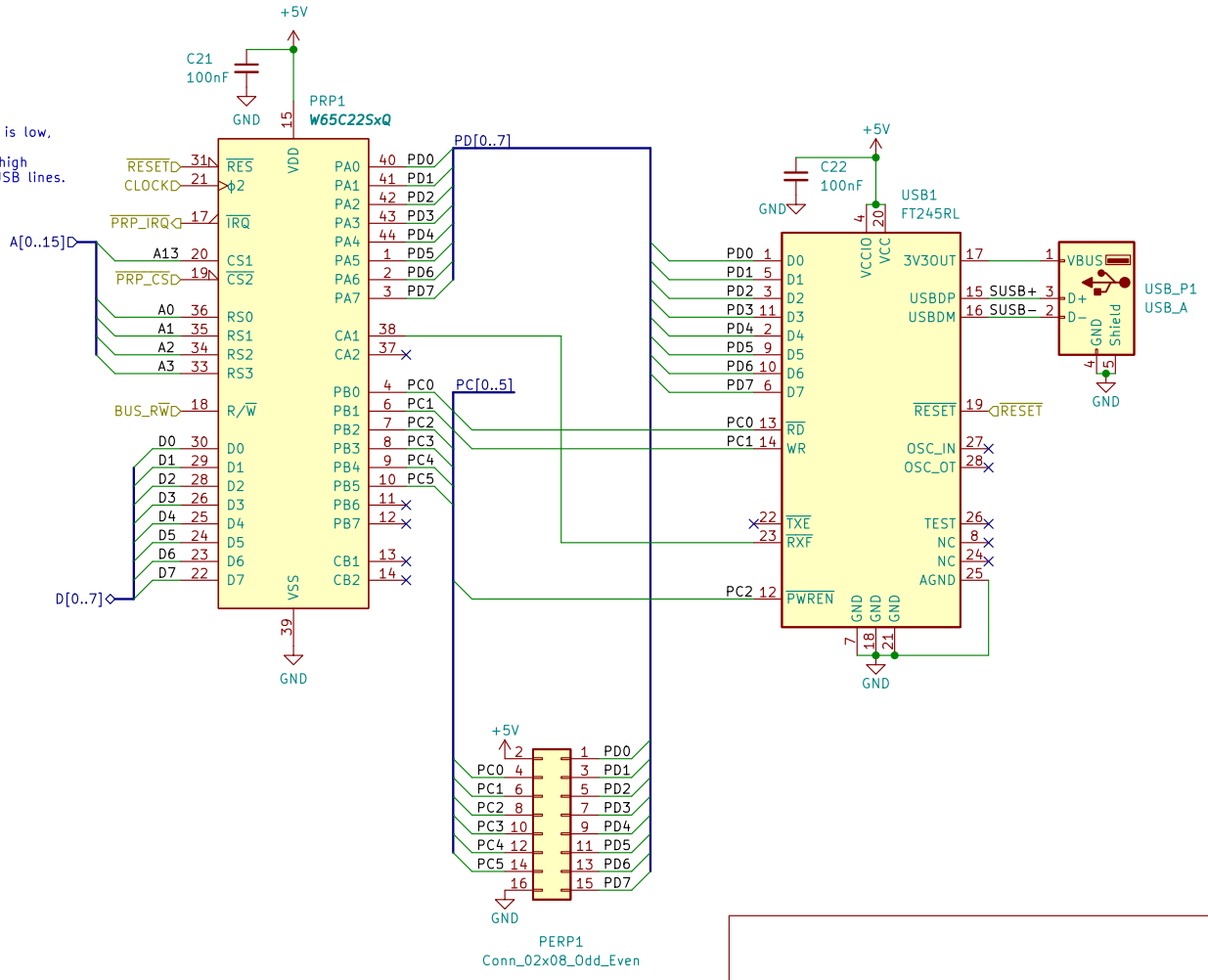
0x6000 – 0x600F

READ FROM USB

Controller's $\overline{RD}$ (PC0) is set low.
USB data is shifting onto PORT A.
Controller pulls $\overline{RXF}$ low, PORT A is latched
onto DATA bus and IRQ is set.

WRITE TO USB

When the VIA's both CS are active and BUS_$\overline{RW}$ is low,
it latches DATA bus onto PORT A.
On the next clock edge, controller's WR is set high
and it starts transmitting from PORT A to the USB lines.
Then WR is set low again.

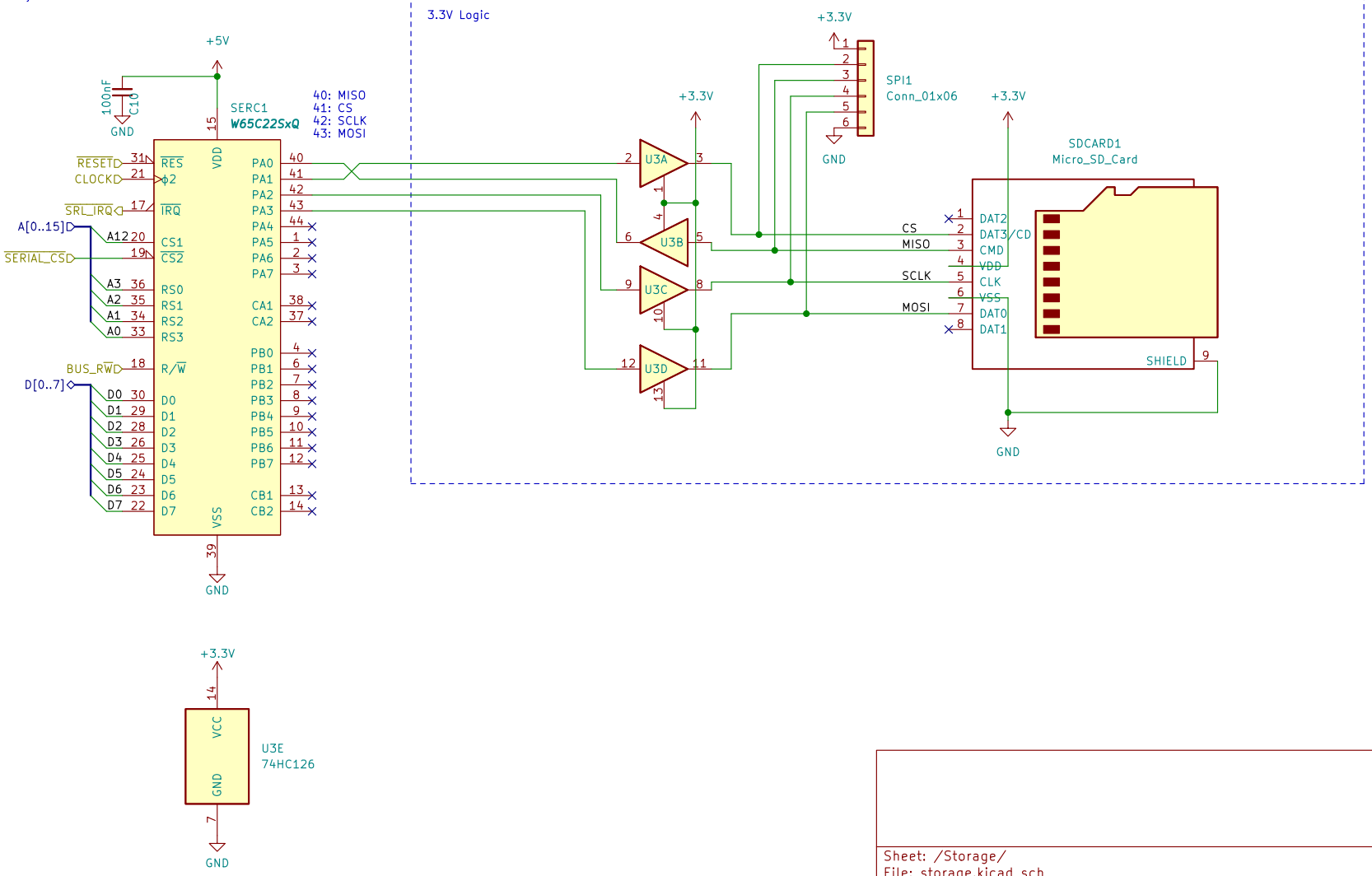


zrthxn		
Sheet: /Peripheral Handler/		
File: peripheral.kicad_sch		
Title: 8puter		
Size: A4	Date:	**Rev: 1.0**
KiCad E.D.A. eeschema 6.0.4-1.fc35		Id: 3/7

SPI SD Card Interface

0x5000 – 0x500F

Serial SPI Interface will
be used to interact primarily
with SD cards. There is also a
SPI Header to allow connecting other
devices but only ONE at a time.



Sheet: /Storage/		
File: storage.kicad_sch		
Title:		
Size: A4	Date:	**Rev:**
KiCad E.D.A. eeschema 6.0.4-1.fc35		Id: 4/7

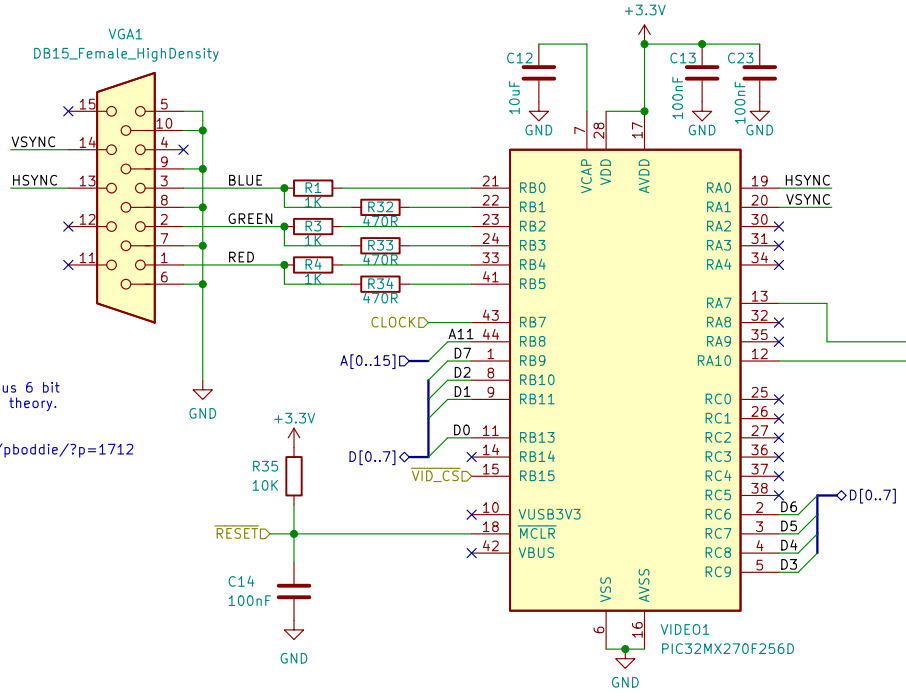
Video Controller

0x4800 – 0x480F

Video will be generated using a microcontroller which is fast enough to generate the VGA timing signals.

CPU will send a single byte to the MCU which can be be a char code or index of glyph, and the MCU just generates the video signal.

This setup is limited in generating graphics but it avoids having to keep a large framebuffer.



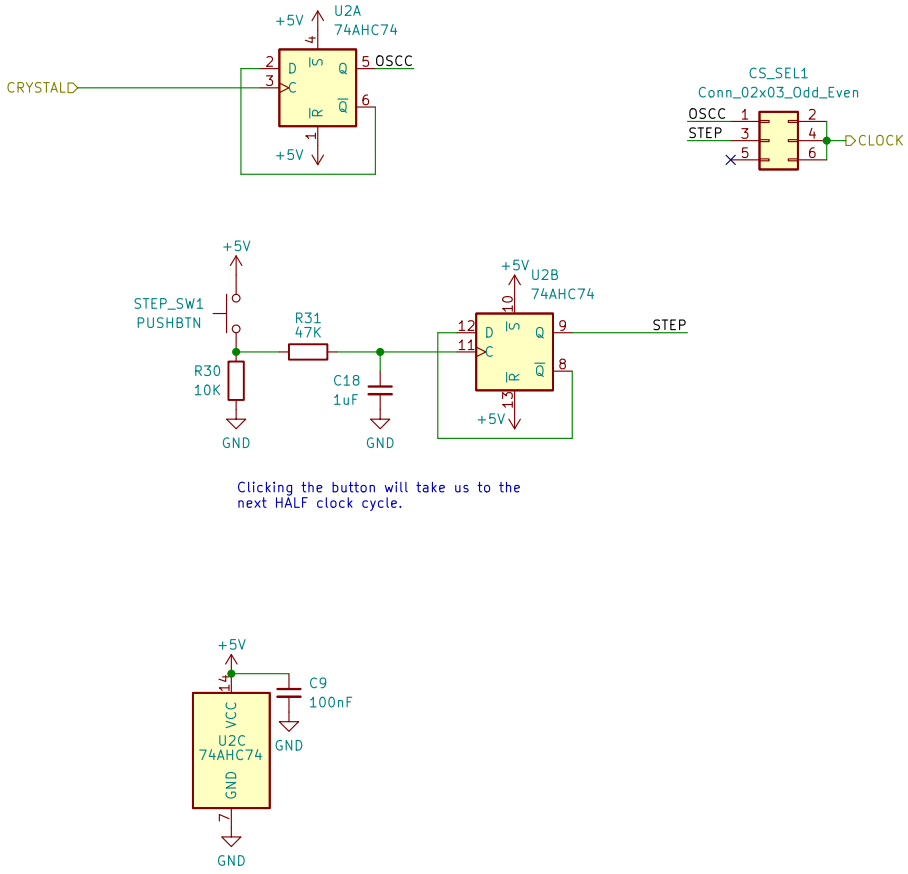
Two bits are used for each color, this gives us 6 bit color and 32 colors in theory.

Ideas from this article
https://blogs.fsfe.org/pboddie/?p=1712

Programming can be done with PICkit3 programmer made by Microchip. They provide an IDE called MPLAB X which take a .hex file and flashes it into memory.

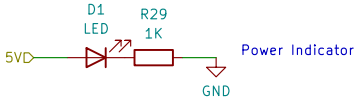
zrthxn		
Sheet: /Video/		
File: video.kicad_sch		
Title: 8puter		
Size: A4	Date:	**Rev: 1.0**
KiCad E.D.A. eeschema 6.0.4-1.fc35		Id: 5/7

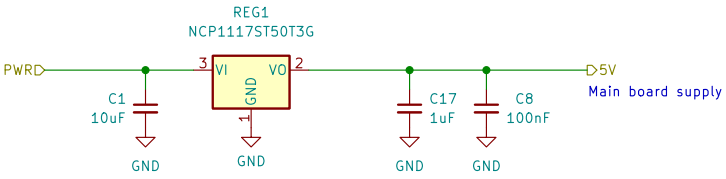
Clock Source Select

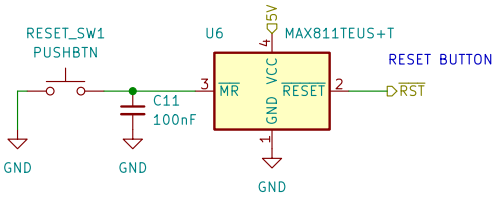


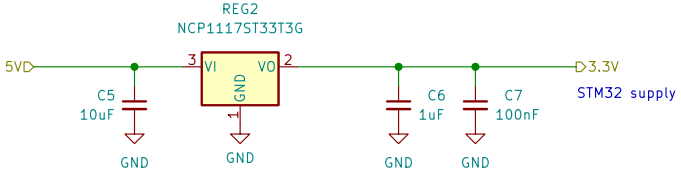
zrthxn		
Sheet: /Clock/		
File: clock.kicad_sch		
Title: 8puter		
Size: A4	Date:	**Rev: 1.0**
KiCad E.D.A. eeschema 6.0.4-1.fc35		Id: 6/7

Power Delivery









zrthxn		
Sheet: /Power/		
File: power.kicad_sch		
Title: 8puter		
Size: A4	Date:	**Rev: 1.0**
KiCad E.D.A. eeschema 6.0.4-1.fc35		Id: 7/7