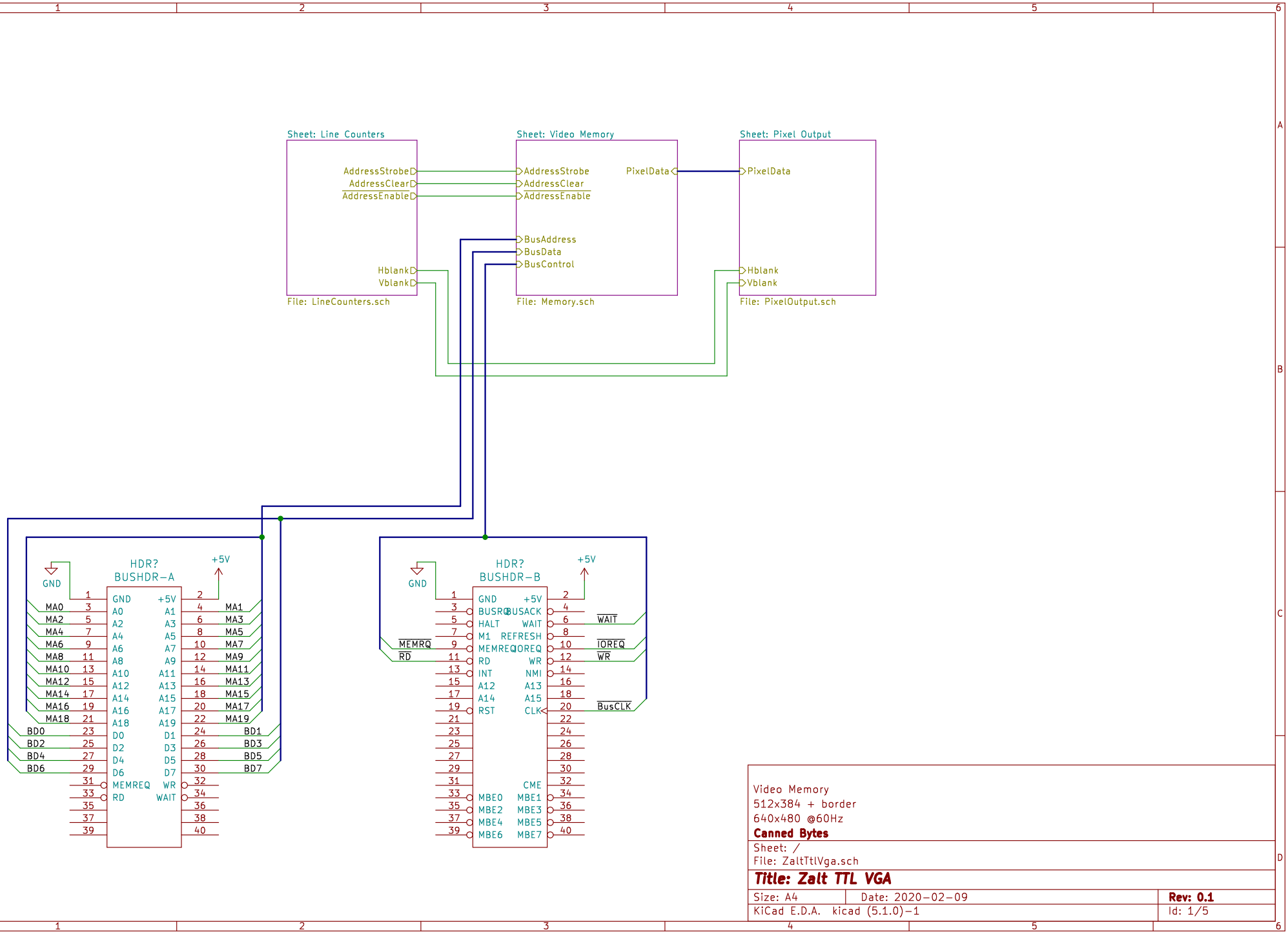
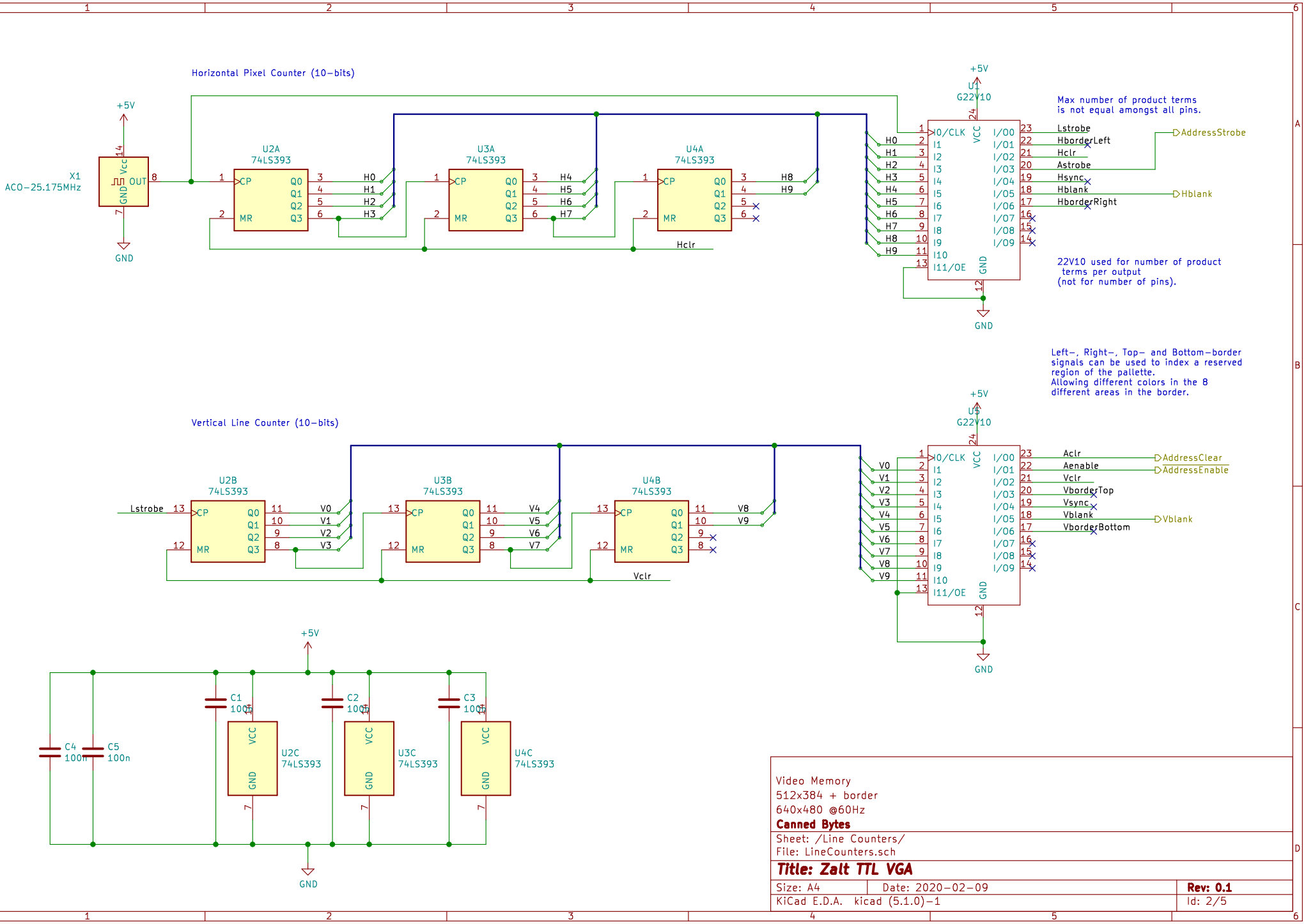


Video Memory
512x384 + border
640x480 @60Hz
Canned Bytes

Sheet: /
File: ZaltTtlVga.sch

Title: Zalt TTL VGA

Size: A4	Date: 2020-02-09	**Rev: 0.1**
KiCad E.D.A. kicad (5.1.0)-1		Id: 1/5



Video Memory
512x384 + border
640x480 @60Hz
Canned Bytes

Sheet: /Line Counters/
File: LineCounters.sch

Title: Zalt TTL VGA

Size: A4	Date: 2020-02-09	**Rev: 0.1**
KiCad E.D.A. kicad (5.1.0)-1		Id: 2/5

We could have registers here that load in a start address for each frame. That way a scrolling function can be supported in hardware. The software would load a value of the address of the start of the next frame based on the direction of scrolling. Left or right would simply increment or decrement the start address. Scrolling up and down would be adding/subtracting a complete horizontal line worth of bytes.

Video Memory Pixel Address Counter (18-bits)

U6 G22V10

U7 G22V10

U8A 74LS139

U9 UM61512A

U10 UM61512A

U11 74LS245

U12 74LS374

U13 74LS245

U14 74LS245

AddressStrobe

AddressClear

This jumper is to disable the VGA VRAM access (removed). This allows the SystemController to easily perform tests and access VRAM without synchronization.

JP? Jumper

AddressEnable

R? 10k

BusAddress

BusControl

BusData

Sheet: Bus Decoder

AddressEnable

BusAddress

BusControl

BusDriverEnable

ControlRegisterEnable

File: BusDecoder.sch

PixelData

Z80 interfaces through a 4k Memory Page.
The location (bit address) of this memory page is determined by the Bus Decoder.

The Control Register provides the upper 6 bits A12..A17 of the VRAM address and can only be changed through an IO write.
The IO address (A0-A7) is also determined by the Bus Decoder.

U8C 74LS139

C6 100n

C7 100n

C8 100n

C12 100n

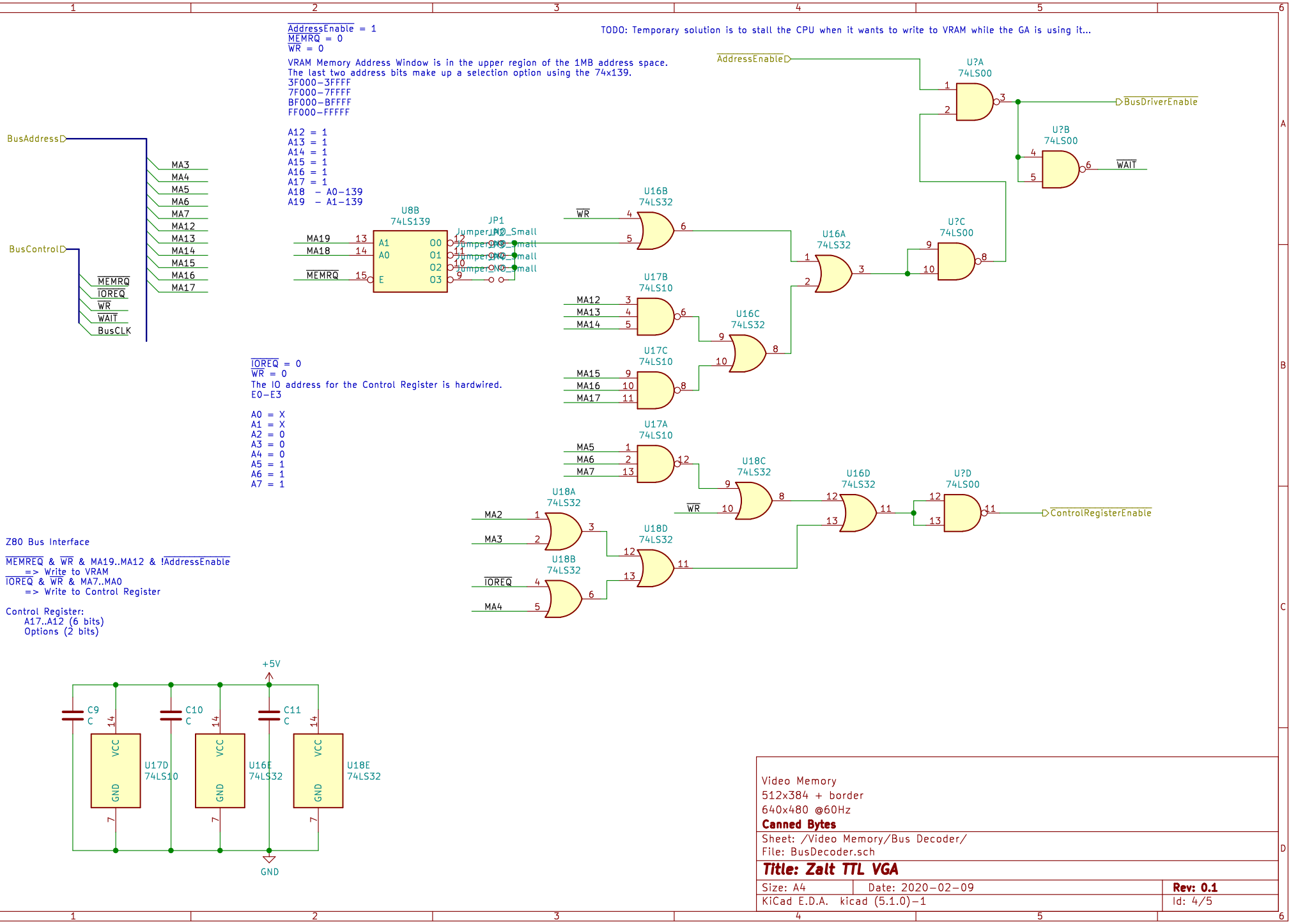
C13 100n

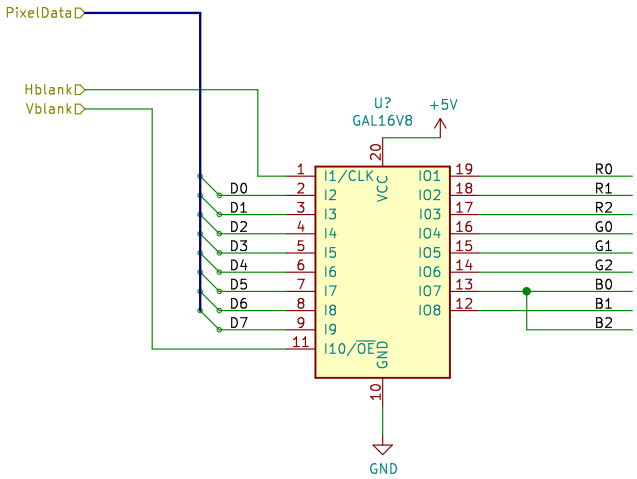
Video Memory
512x384 + border
640x480 @60Hz
Canned Bytes

Sheet: /Video Memory/
File: Memory.sch

Title: Zalt TTL VGA

Size: A3	Date: 2020-02-09	**Rev: 0.1**
KiCad E.D.A. kicad (5.1.0)-1		Id: 3/5


AddressEnable = 1
MEMRQ = 0
WR = 0
VRAM Memory Address Window is in the upper region of the 1MB address space.
The last two address bits make up a selection option using the 74x139.
3F000-3FFFF
7F000-7FFFF
BF000-BFFFF
FF000-FFFFF
A12 = 1
A13 = 1
A14 = 1
A15 = 1
A16 = 1
A17 = 1
A18 - A0-139
A19 - A1-139
TODO: Temporary solution is to stall the CPU when it wants to write to VRAM while the GA is using it...
AddressEnable
BusAddress
BusControl
MA3
MA4
MA5
MA6
MA7
MA12
MA13
MA14
MA15
MA16
MA17
MEMRQ
IOREQ
WR
WAIT
BusCLK
U8B
74LS139
JP1
Jumper_NO_Small
U?A
74LS00
BusDriverEnable
U?B
74LS00
WAIT
U16B
74LS32
U16A
74LS32
U?C
74LS00
U17B
74LS10
U16C
74LS32
U17C
74LS10
IOREQ = 0
WR = 0
The IO address for the Control Register is hardwired.
E0-E3
A0 = X
A1 = X
A2 = 0
A3 = 0
A4 = 0
A5 = 1
A6 = 1
A7 = 1
U17A
74LS10
U18C
74LS32
U16D
74LS32
U?D
74LS00
ControlRegisterEnable
U18A
74LS32
U18D
74LS32
U18B
74LS32
Z80 Bus Interface
MEMREQ & WR & MA19..MA12 & !AddressEnable
=> Write to VRAM
IOREQ & WR & MA7..MA0
=> Write to Control Register
Control Register:
A17..A12 (6 bits)
Options (2 bits)
+5V
C9
C
C10
C
C11
C
U17D
74LS10
U16E
74LS32
U18E
74LS32
VCC
GND
GND
Video Memory
512x384 + border
640x480 @60Hz
Canned Bytes
Sheet: /Video Memory/Bus Decoder/
File: BusDecoder.sch
Title: Zalt TTL VGA
Size: A4
Date: 2020-02-09
Rev: 0.1
KiCad E.D.A. kicad (5.1.0)-1
Id: 4/5



Pixel Output
512x384 + border
640x480 @60Hz
Canned Bytes

Sheet: /Pixel Output/
File: PixelOutput.sch

Title: Zalt TTL VGA

Size: A4	Date: 2020-02-09	**Rev: 0.1**
KiCad E.D.A. kicad (5.1.0)-1		Id: 5/5