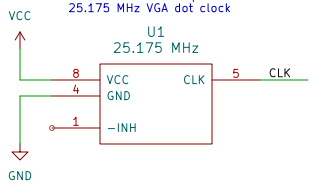
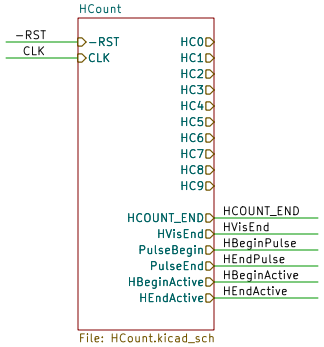
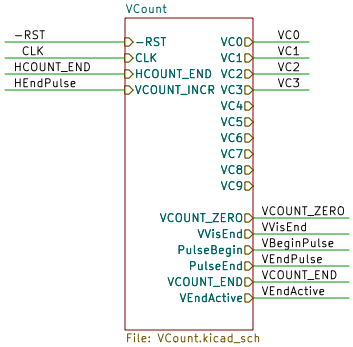
Oscillator module to produce
25.175 MHz VGA dot clock



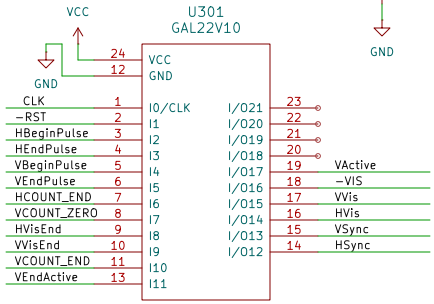
Horizontal count

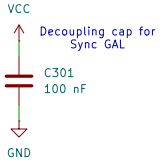


Vertical count

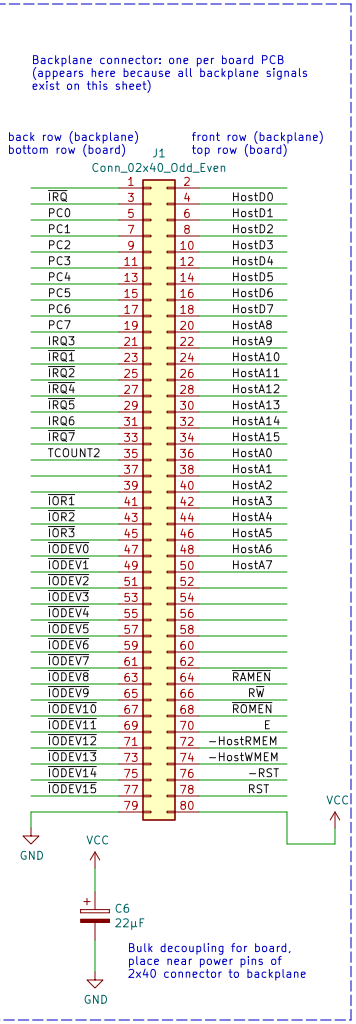


Sync gen and visibility signals:
Logic is in Sync.pld

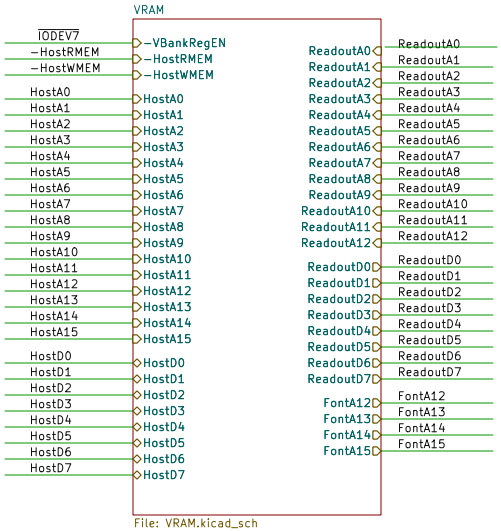




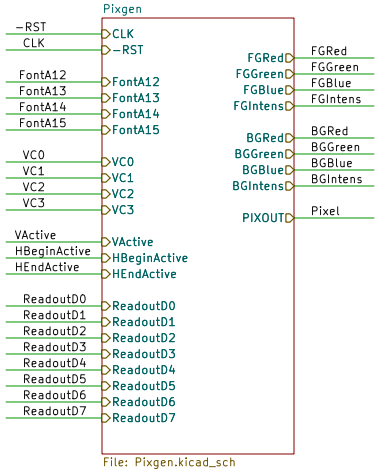
Backplane connector: one per board PCB
(appears here because all backplane signals
exist on this sheet)



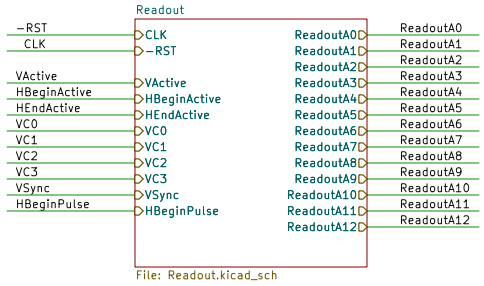
VRAM (video memory for
character and attribute data)



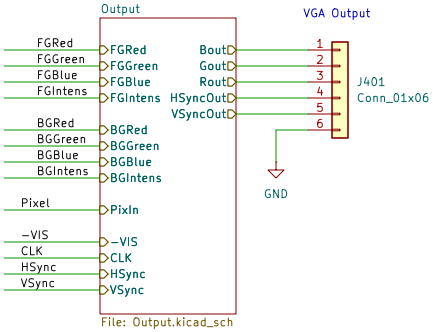
Pixel generation (using data from VRAM,
generate sequence of pixel fg/bg values
along with fg/bg color information
to send to Output module for display)



Readout module (generate addresses
for reading character and attribute
data from VRAM)



Video output
(to VGA monitor)

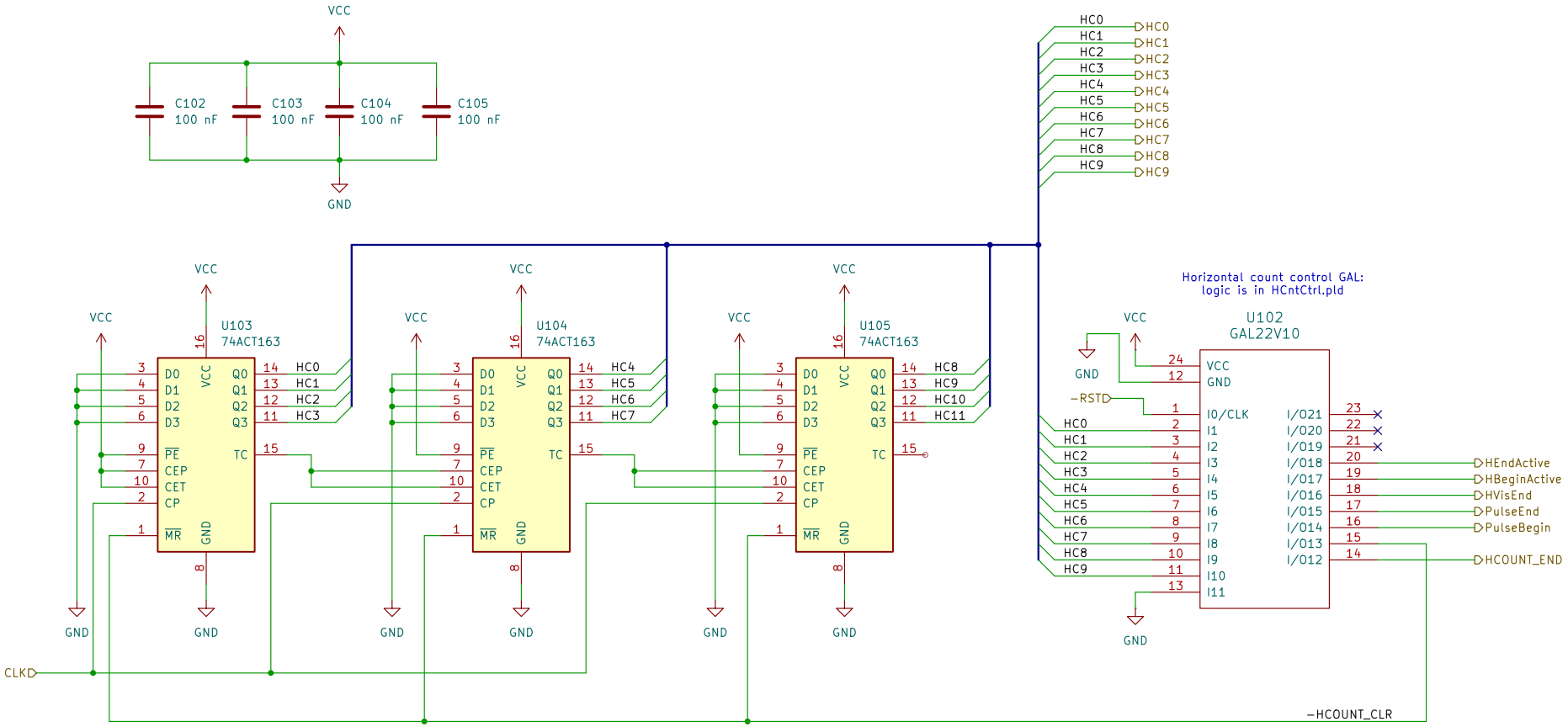


daveho hacks

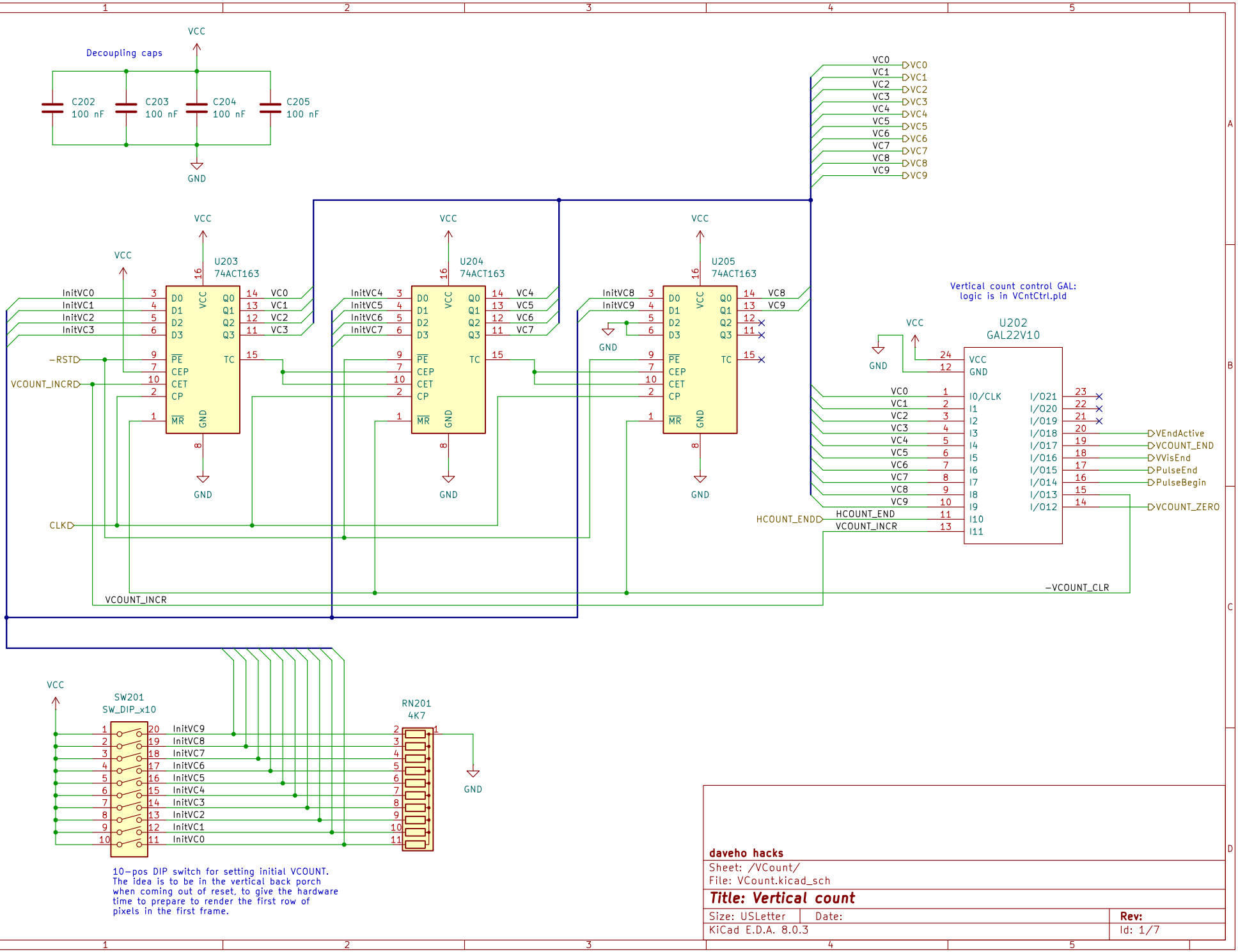
Sheet: /
File: HW_VGA.kicad_sch

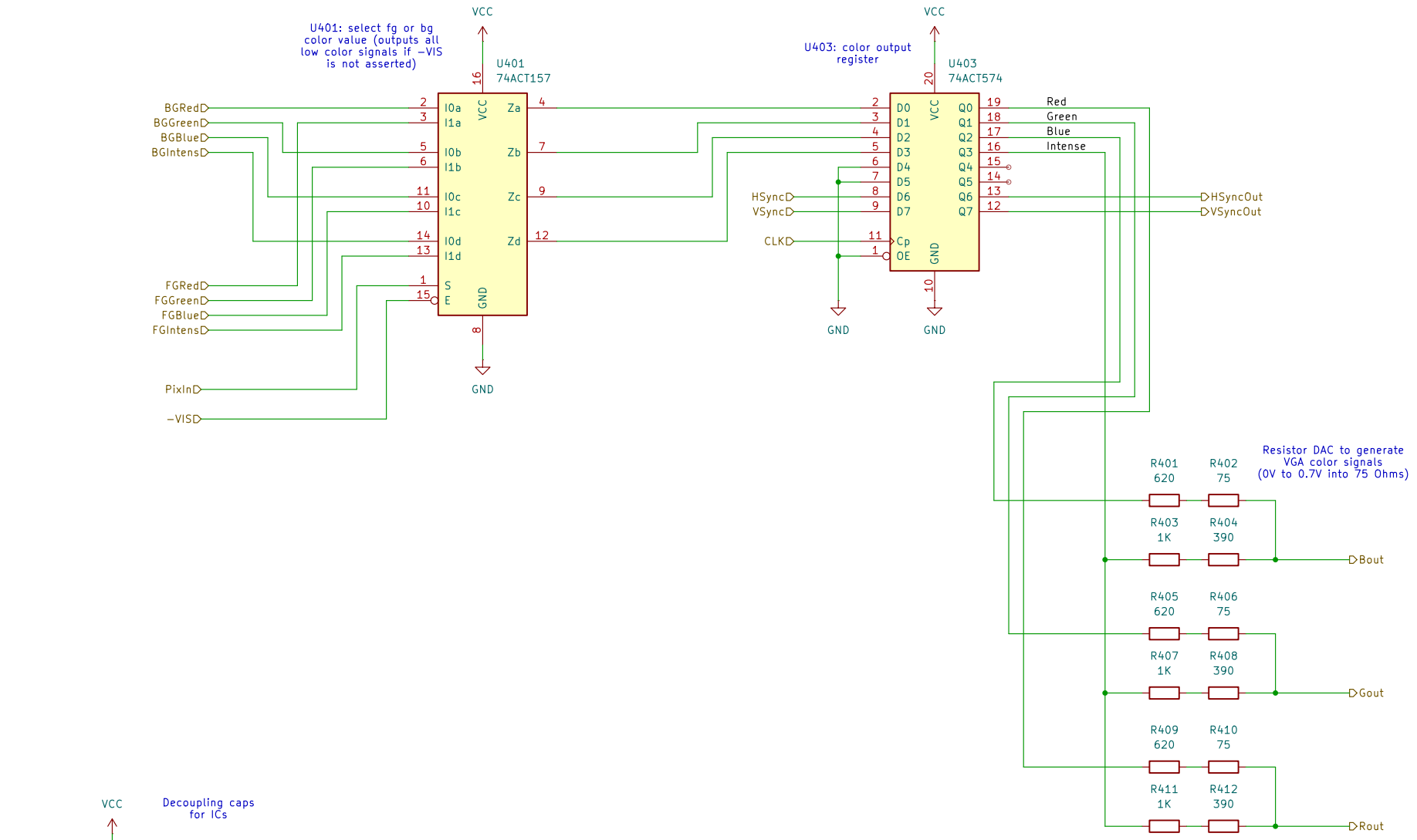
Title: Hardware VGA

Size: User	Date:	Rev:
KiCad E.D.A. 8.0.3		Id: 1/7

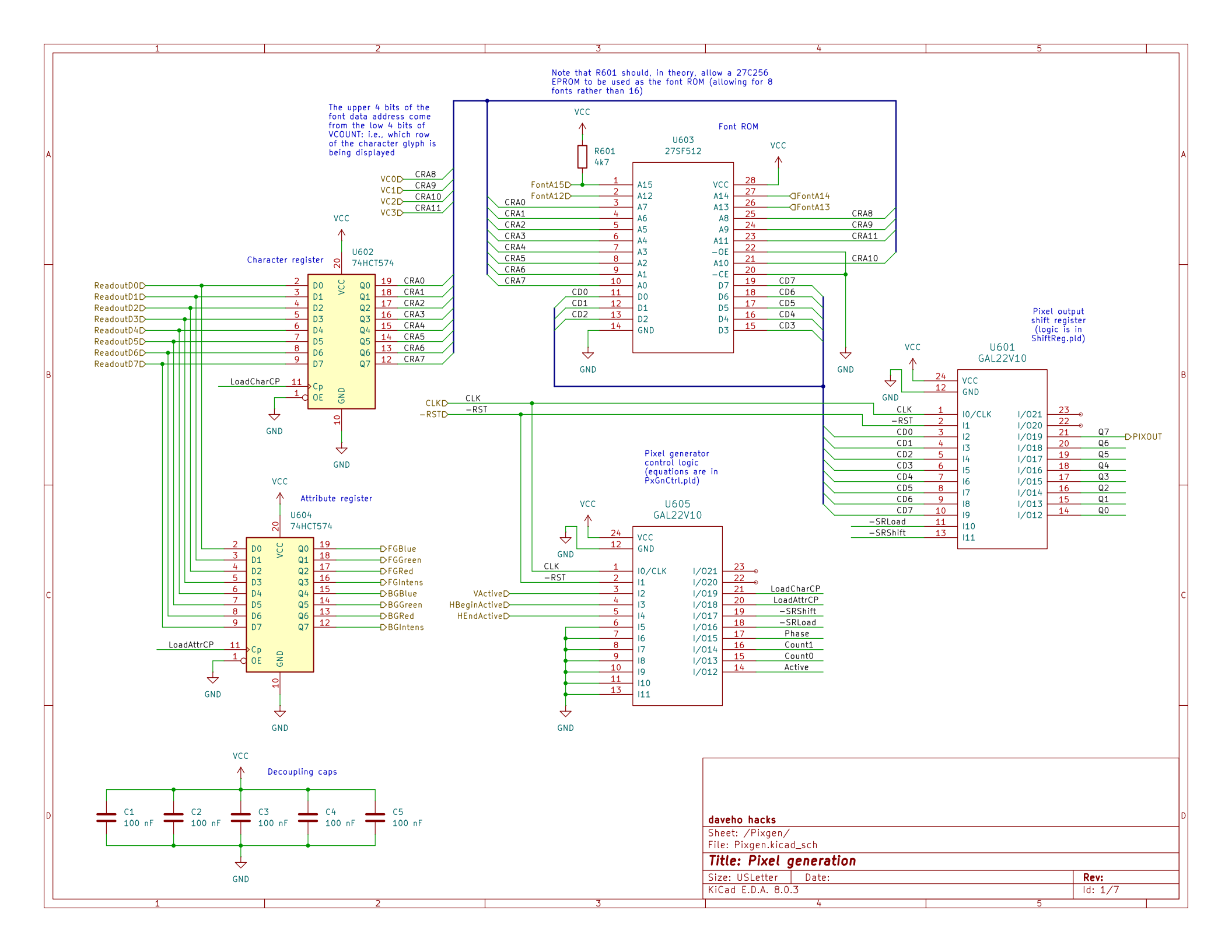


daveho hacks	
Sheet: /HCount/	
File: HCount.kicad_sch	
Title: Horizontal count	
Size: USLetter | Date:	**Rev:**
KiCad E.D.A. 8.0.3	Id: 1/7

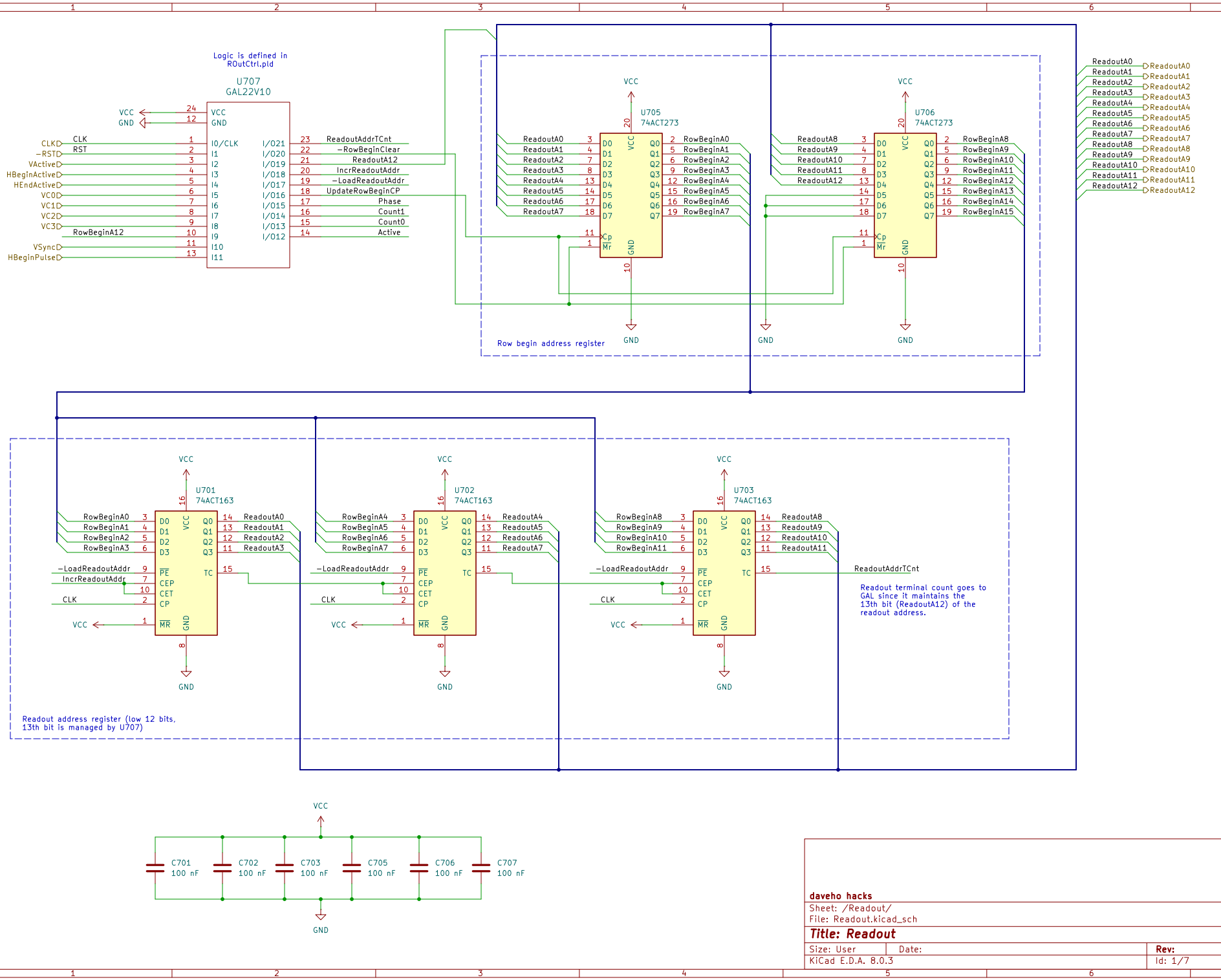

Decoupling caps
C202 100 nF
C203 100 nF
C204 100 nF
C205 100 nF
VCC
GND
U203 74ACT163
U204 74ACT163
U205 74ACT163
InitVC0
InitVC1
InitVC2
InitVC3
InitVC4
InitVC5
InitVC6
InitVC7
InitVC8
InitVC9
VC0
VC1
VC2
VC3
VC4
VC5
VC6
VC7
VC8
VC9
−RSTD
VCOUNT_INCRD
CLKD
VCOUNT_INCR
Vertical count control GAL: logic is in VCntCtrl.pld
U202 GAL22V10
HCOUNT_ENDD
HCOUNT_END
VCOUNT_INCR
VEndActive
VCOUNT_END
VVisEnd
PulseEnd
PulseBegin
VCOUNT_ZERO
−VCOUNT_CLR
SW201 SW_DIP_x10
RN201 4K7
10−pos DIP switch for setting initial VCOUNT.
The idea is to be in the vertical back porch
when coming out of reset, to give the hardware
time to prepare to render the first row of
pixels in the first frame.
daveho hacks
Sheet: /VCount/
File: VCount.kicad_sch
Title: Vertical count
Size: USLetter
Date:
Rev:
KiCad E.D.A. 8.0.3
Id: 1/7



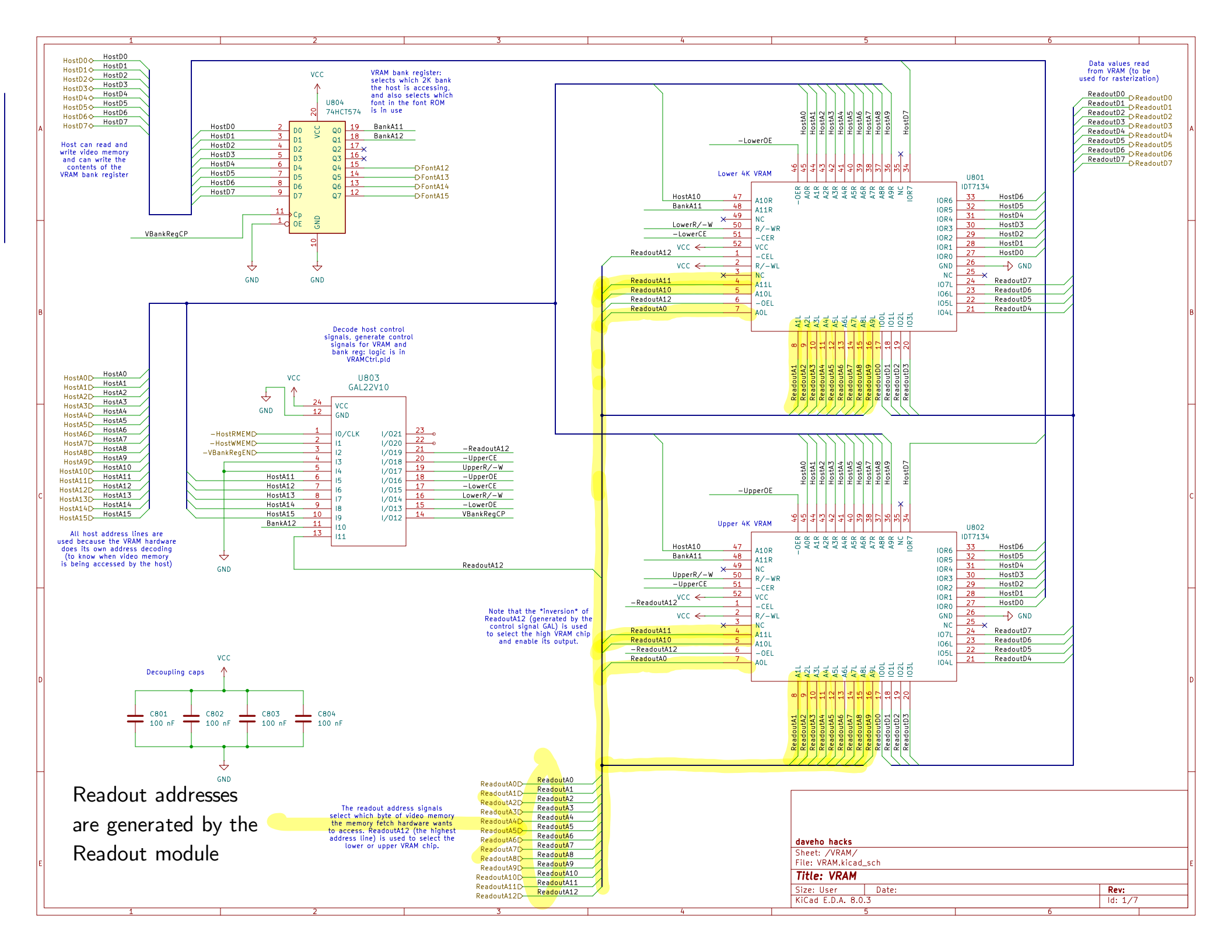
daveho hacks	
Sheet: /Output/	
File: Output.kicad_sch	
Title: Output	
Size: USLetter | Date:	**Rev:**
KiCad E.D.A. 8.0.3	Id: 4/7



daveho hacks	
Sheet: /Pixgen/	
File: Pixgen.kicad_sch	
Title: Pixel generation	
Size: USLetter | Date:	**Rev:**
KiCad E.D.A. 8.0.3	Id: 1/7



daveho hacks	
Sheet: /Readout/	
File: Readout.kicad_sch	
Title: Readout	
Size: User	Date:
KiCad E.D.A. 8.0.3	Rev:
	Id: 1/7


Host can read and write video memory and can write the contents of the VRAM bank register
VRAM bank register: selects which 2K bank the host is accessing, and also selects which font in the font ROM is in use
U804
74HCT574
VBankRegCP
BankA11
BankA12
FontA12
FontA13
FontA14
FontA15
Data values read from VRAM (to be used for rasterization)
Lower 4K VRAM
Upper 4K VRAM
U801
IDT7134
U802
IDT7134
-LowerOE
-UpperOE
LowerR/-W
UpperR/-W
-LowerCE
-UpperCE
Decode host control signals, generate control signals for VRAM and bank reg; logic is in VRAMCtrl.pld
U803
GAL22V10
-HostRMEM
-HostWMEM
-VBankRegEN
-ReadoutA12
-UpperCE
UpperR/-W
-UpperOE
-LowerCE
LowerR/-W
-LowerOE
VBankRegCP
All host address lines are used because the VRAM hardware does its own address decoding (to know when video memory is being accessed by the host)
Note that the *inversion* of ReadoutA12 (generated by the control signal GAL) is used to select the high VRAM chip and enable its output.
Decoupling caps
C801 100 nF
C802 100 nF
C803 100 nF
C804 100 nF
The readout address signals select which byte of video memory the memory fetch hardware wants to access. ReadoutA12 (the highest address line) is used to select the lower or upper VRAM chip.
Readout addresses are generated by the Readout module
daveho hacks
Sheet: /VRAM/
File: VRAM.kicad_sch
Title: VRAM
Size: User
Date:
Rev:
KiCad E.D.A. 8.0.3
Id: 1/7