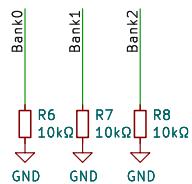


The bus connector has a shared open-drain active-high RDY signal. If all bus devices are ready then they allow the line to remain high. If any bus device is not ready then it drives the line low. When RDY is driven low, the CPU Phi1 clock stops and the CPU disconnects from the bus, placing the lines in a high-Z mode.

If no bus devices are connected then the CPU is always ready.



Peripheral devices on separate boards connect to the system bus here.

Sheet: /System Bus Connector/  
File: System Bus Connector.sch

**Title: System Bus Connector**

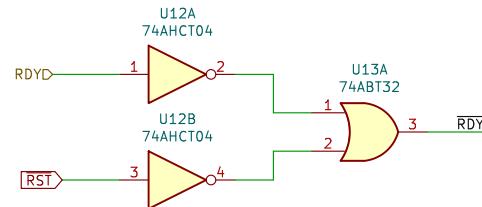
Size: USLetter | Date: 2021-06-22

KiCad E.D.A. kicad 7.0.8

**Rev: B (cb021f9a)**

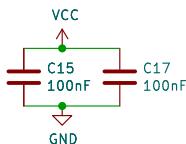
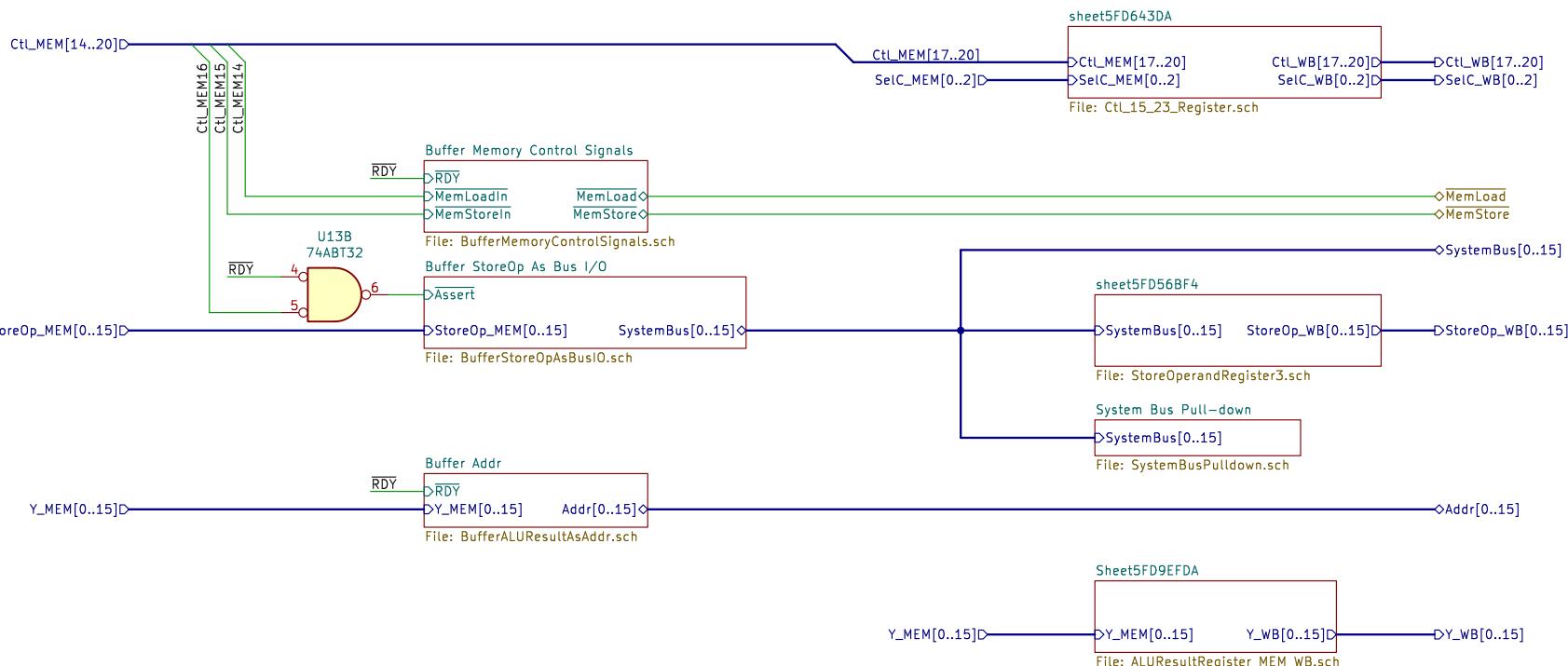
Id: 2/39

1 2 3 4 5



The RDY signal is an open-collector signal shared between all bus devices. When a bus device takes this signal high, the CPU releases the system bus I/O lines, address lines, and control signals, putting them into a high-Z state. This allows peripheral devices to drive the bus when needed.

The Reset cycle also forces the CPU to release the bus. This prevents erroneous I/O operations on reset as a result of interstage registers being initialized to arbitrary values on power-on.



The MEM stage interfaces with memory and memory-mapped peripherals. These devices connect to the main board via a connector described in another sheet. Devices on the bus may take the open-collector ~RDY~ signal high to force the CPU to place bus lines into tristate and halt the Phi1 clock.

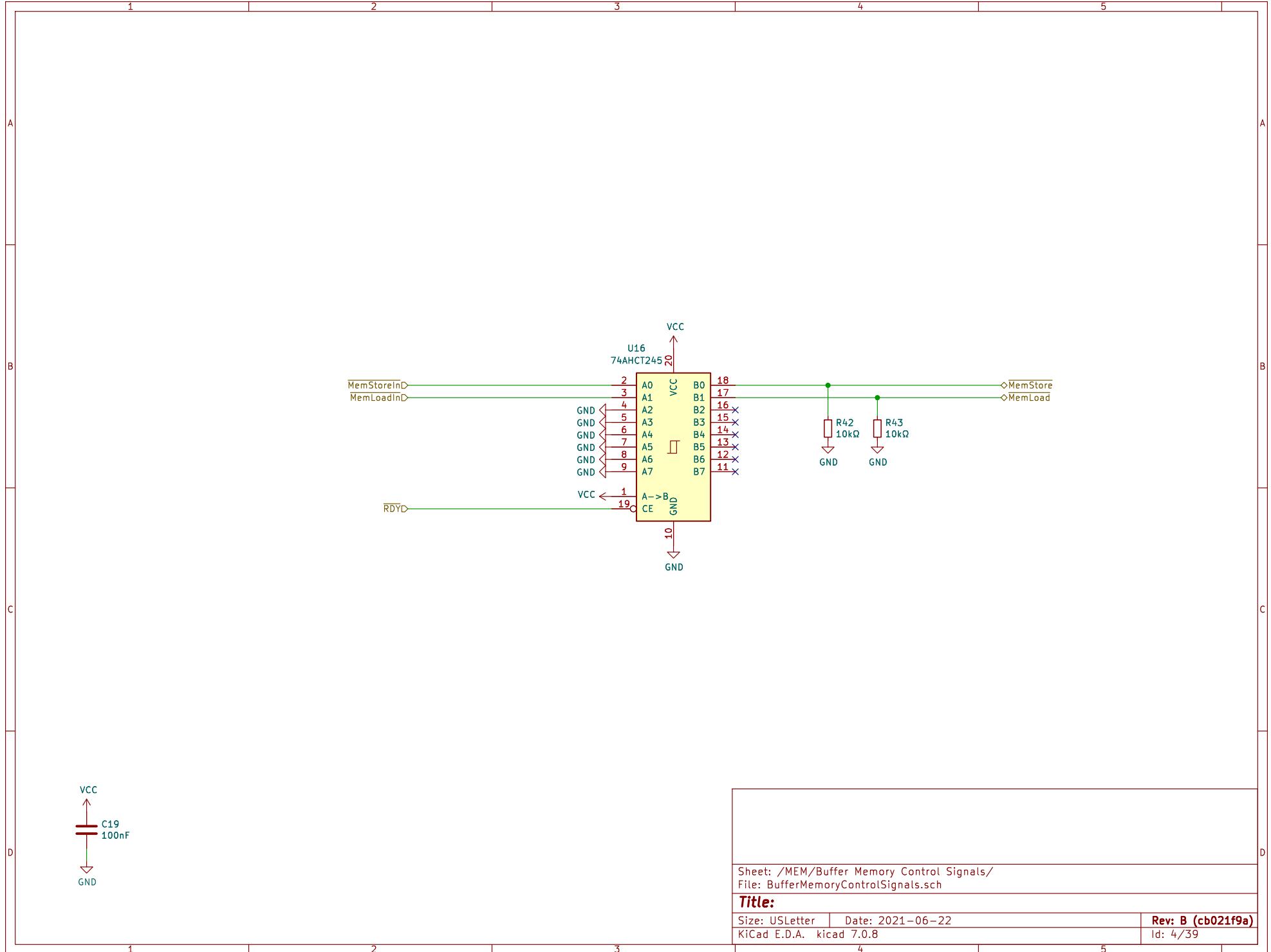
Sheet: /MEM/  
File: MEM.sch

**Title: MEM**

Size: USLetter | Date: 2021-06-22  
KiCad E.D.A. kicad 7.0.8

Rev: B (cb021f9a)  
Id: 3/39

1 2 3 4 5



A

A

B

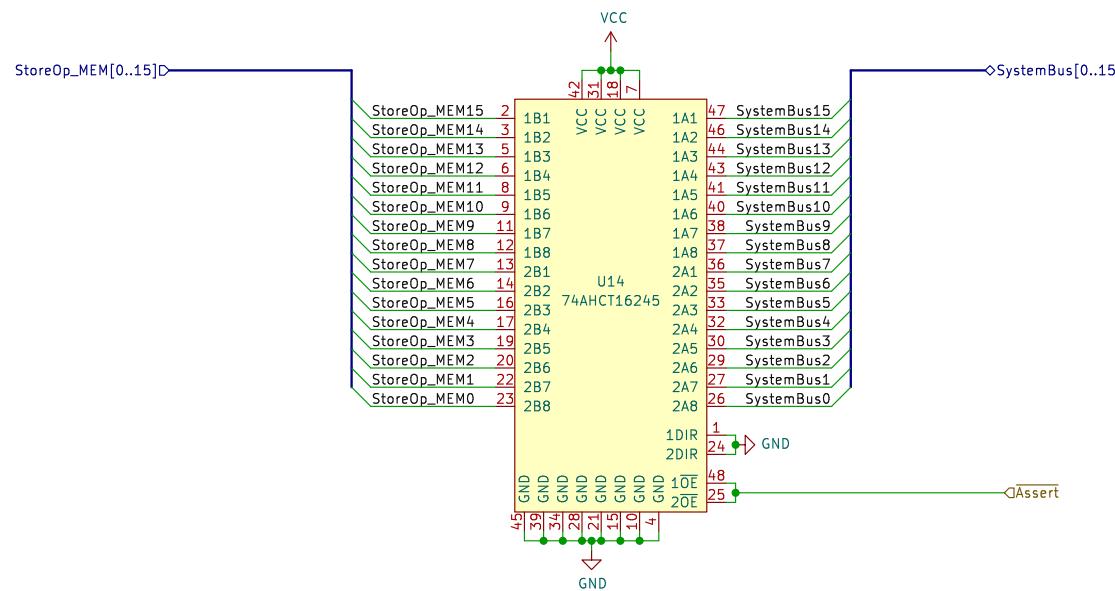
B

C

C

D

D



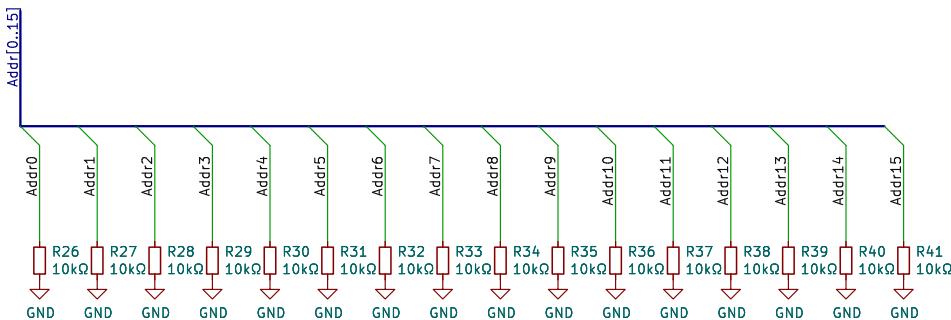
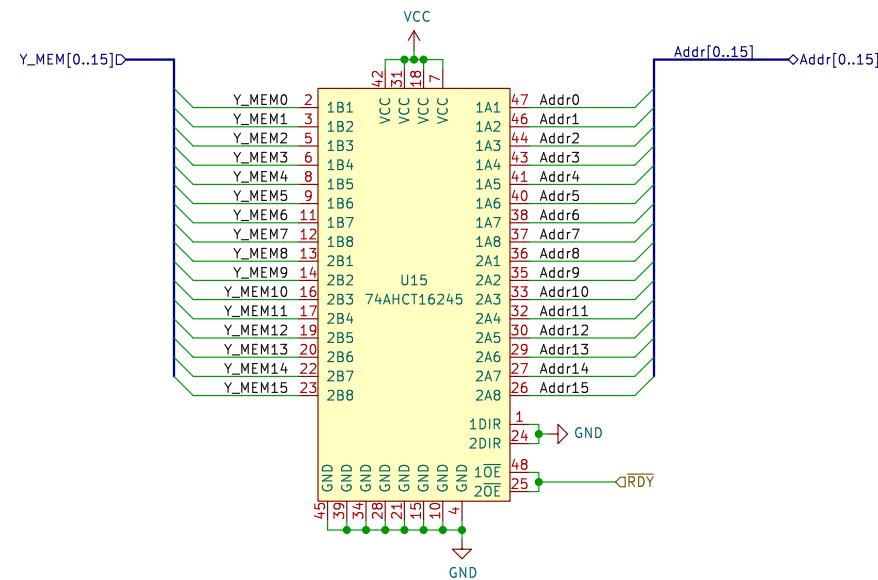
The store operand can be disconnected from the system memory bus via tristate outputs. This allows peripheral devices to assert their own value to the bus during a Load.

Sheet: /MEM/Buffer StoreOp As Bus I/O/  
File: BufferStoreOpAsBusIO.sch

### Title: MEM: Buffer StoreOp

Size: USLetter | Date: 2021-06-22  
KiCad E.D.A. kicad 7.0.8

Rev: B (cb021f9a)  
Id: 5/39



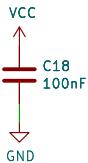
Buffer the effective address before it leaves the main board.

Sheet: /MEM/Buffer Addr/  
File: BufferALUREsultAsAddr.sch

**Title: MEM: Buffer ALUResult**

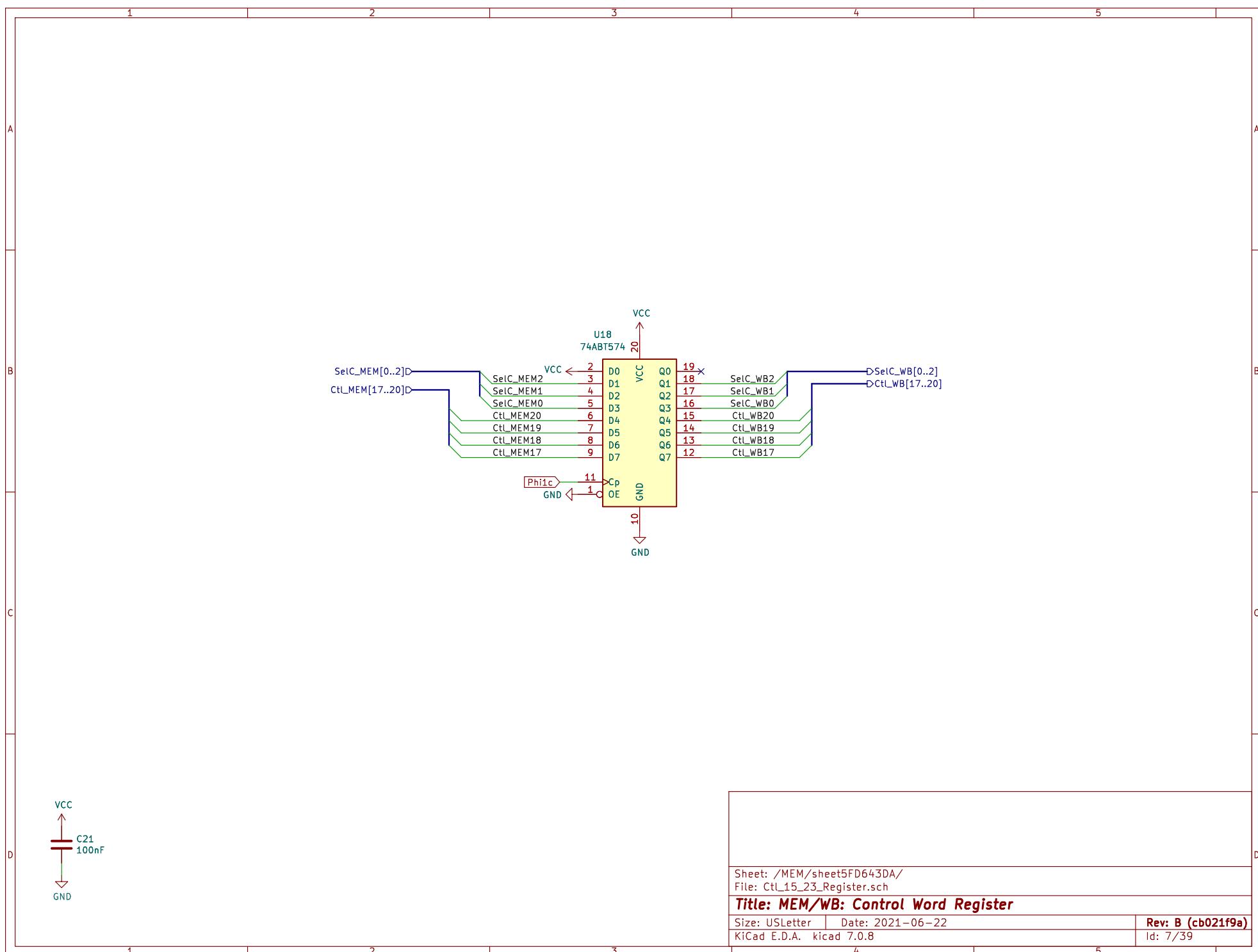
Size: USLetter Date: 2021-06-22

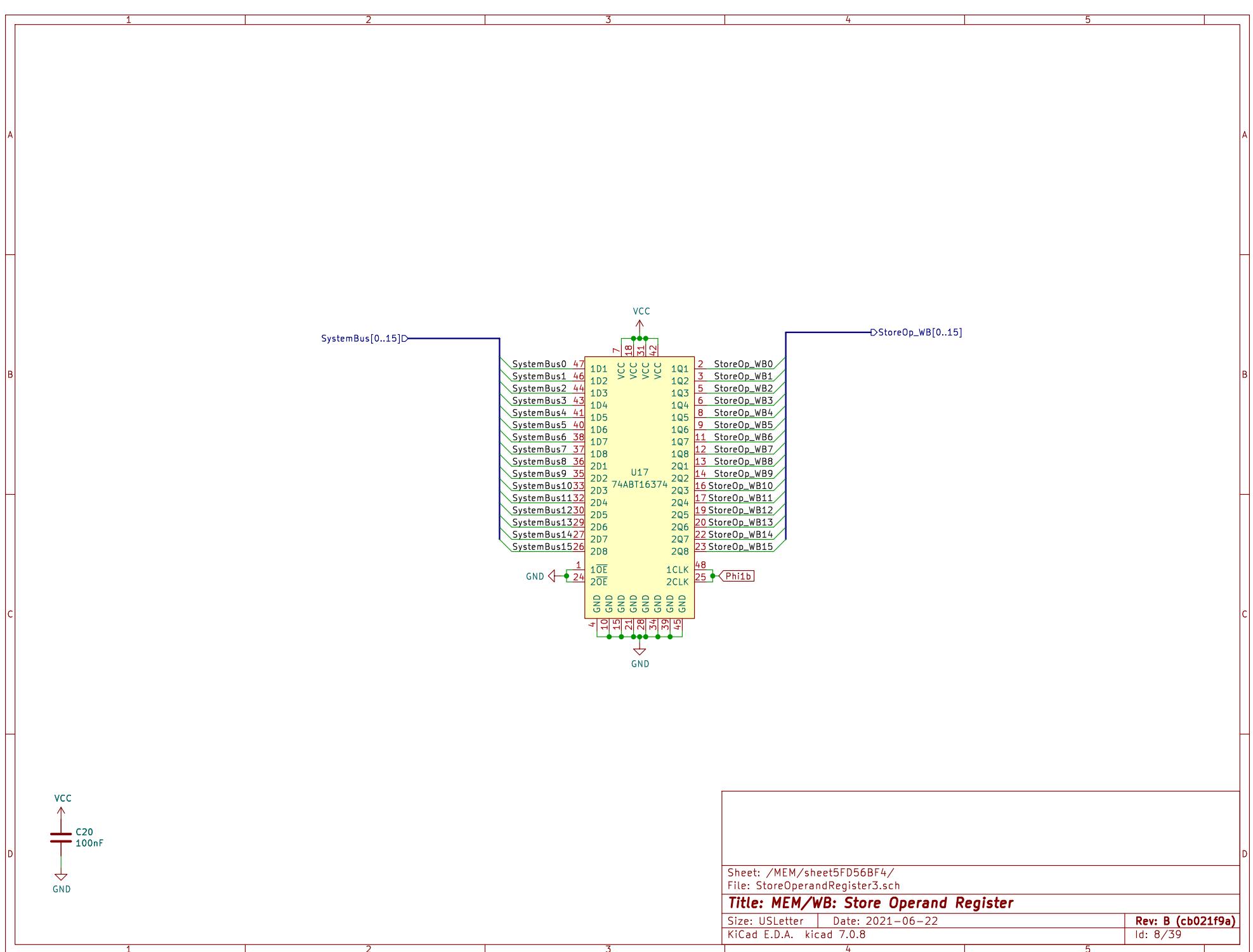
KiCad E.D.A. kicad 7.0.8



Rev: B (cb021f9a)

Id: 6/39





A

A

B

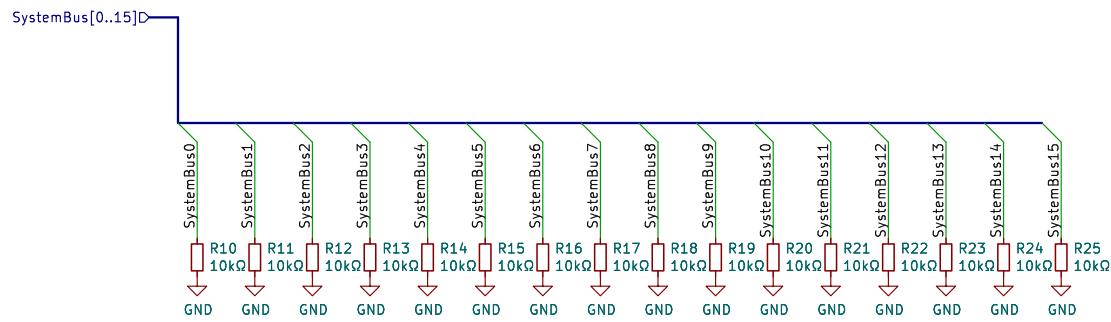
B

C

C

D

D



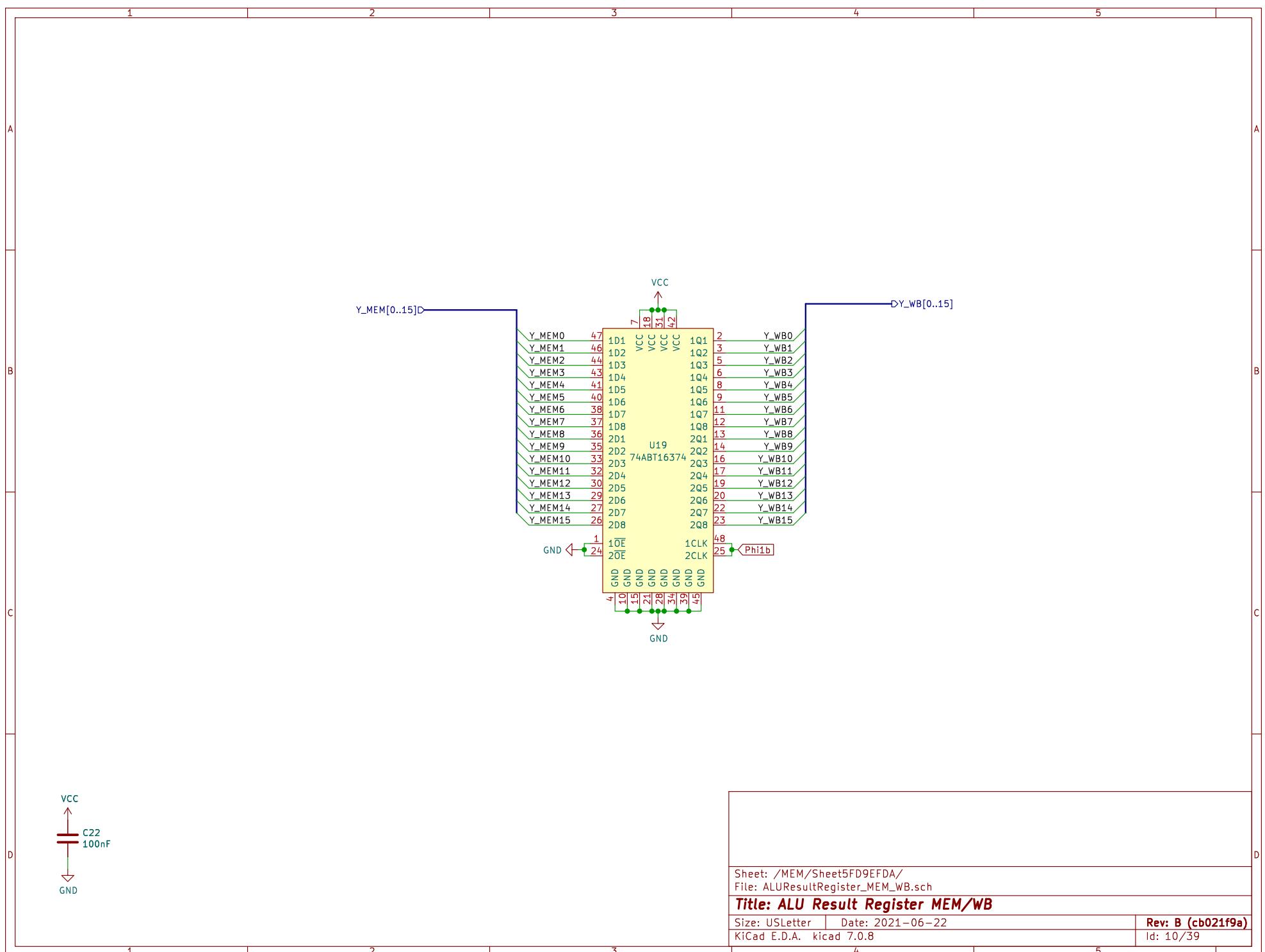
The bus needs pull-down resistors since these lines may otherwise float. sometimes.

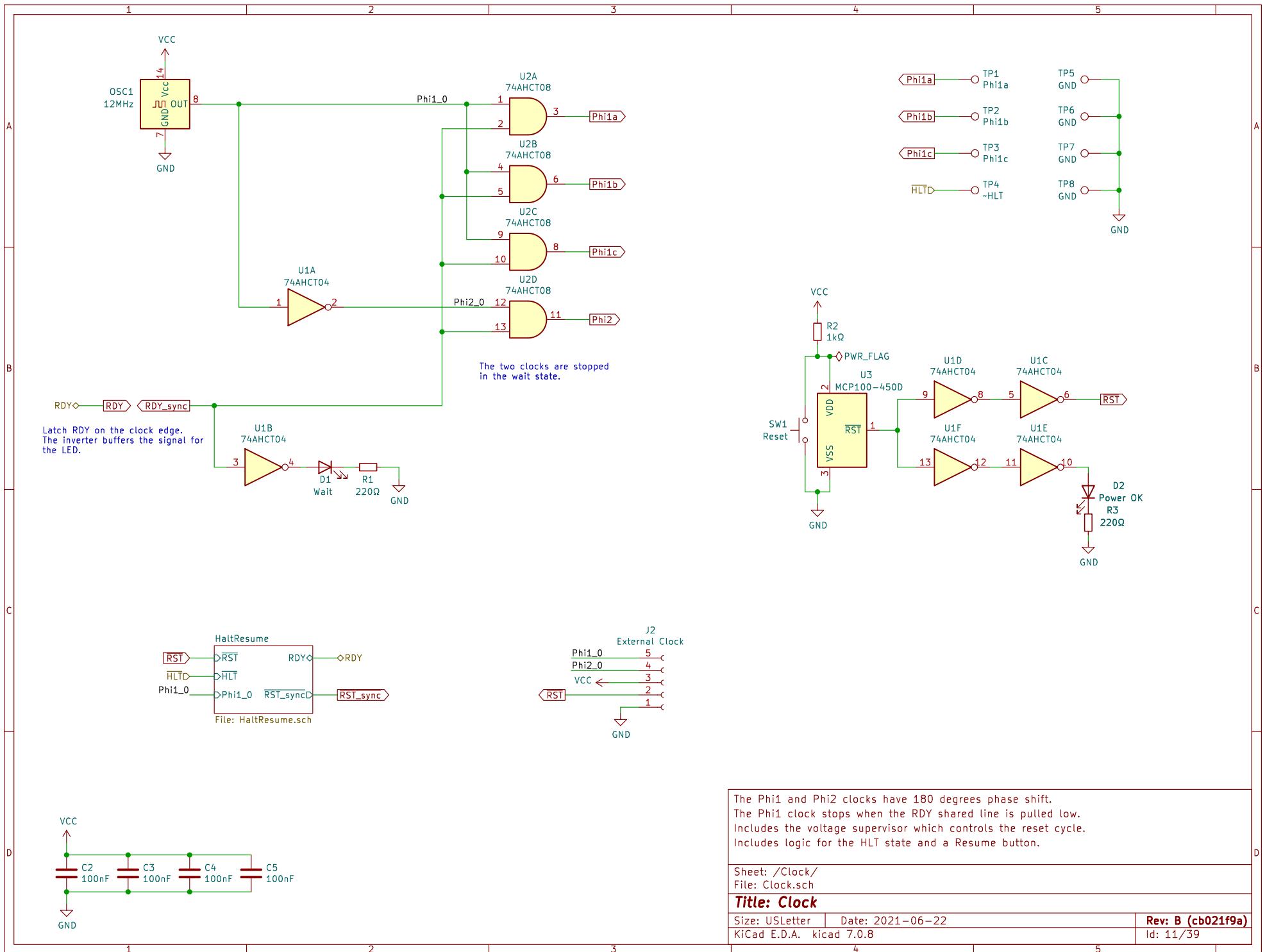
Sheet: /MEM/System Bus Pull-down/  
File: SystemBusPulldown.sch

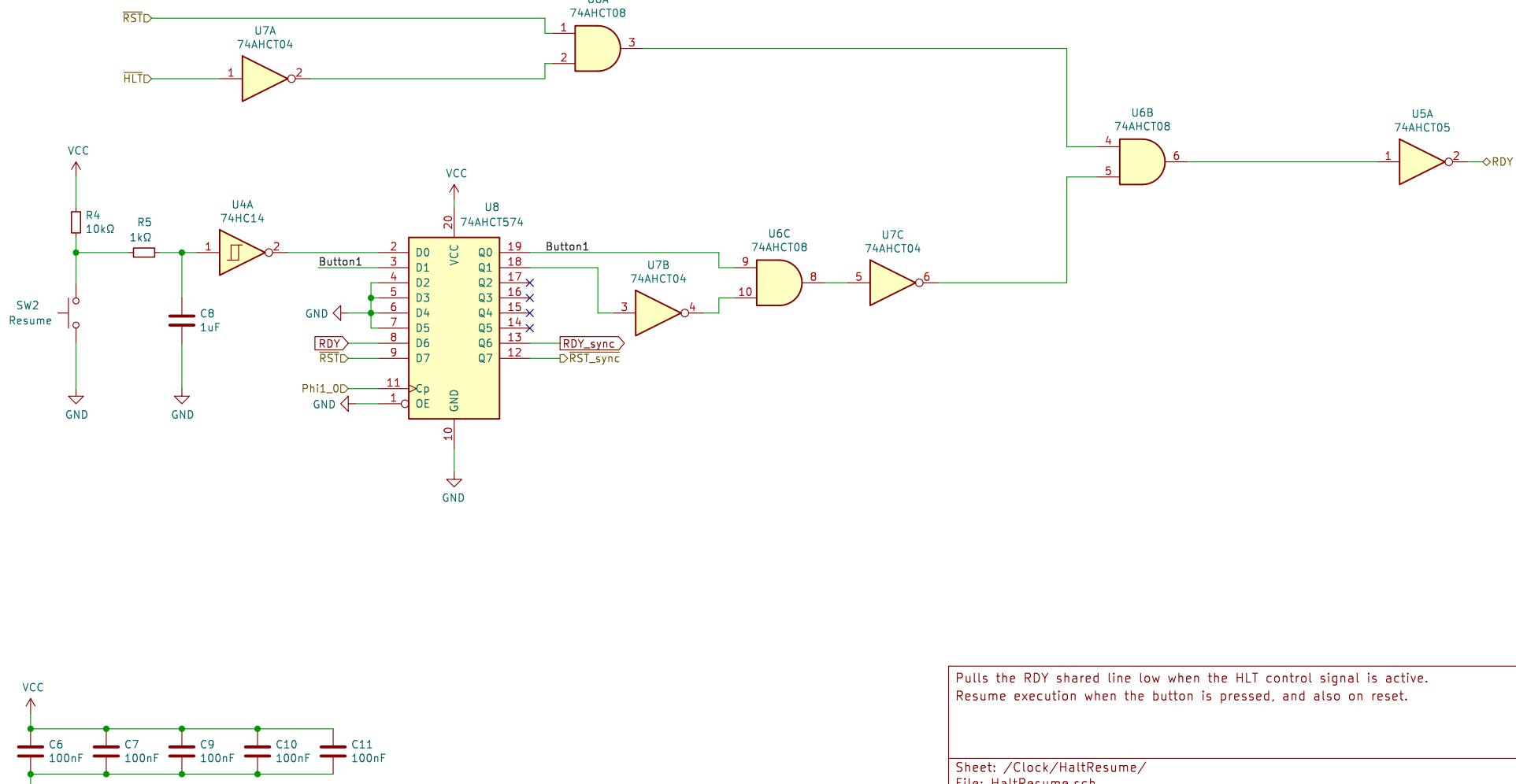
### Title: MEM: System Bus Pull-down Resistors

Size: USLetter Date: 2021-06-22  
KiCad E.D.A. kicad 7.0.8

Rev: B (cb021f9a)  
Id: 9/39







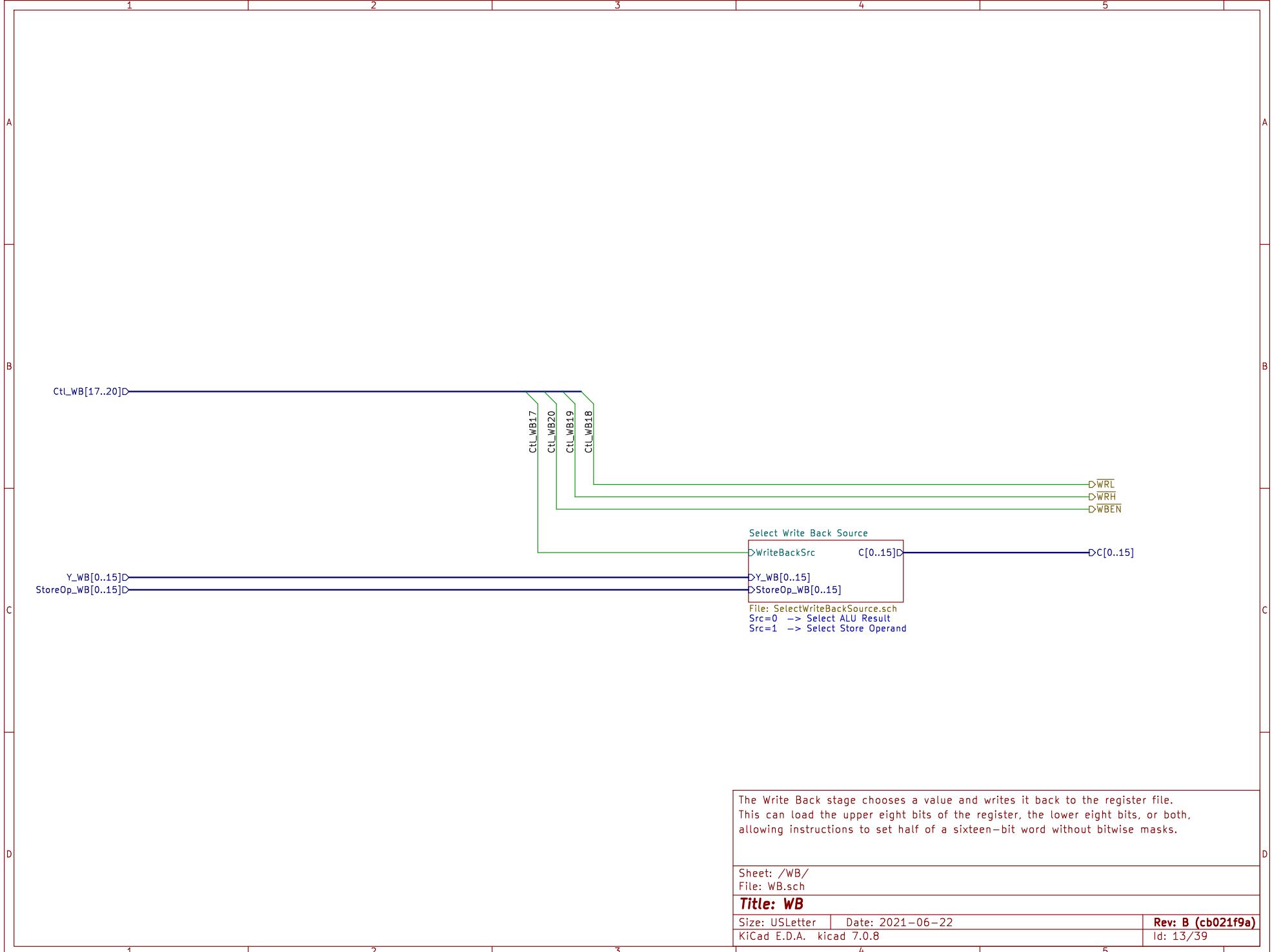
Pulls the RDY shared line low when the HLT control signal is active.  
Resume execution when the button is pressed, and also on reset.

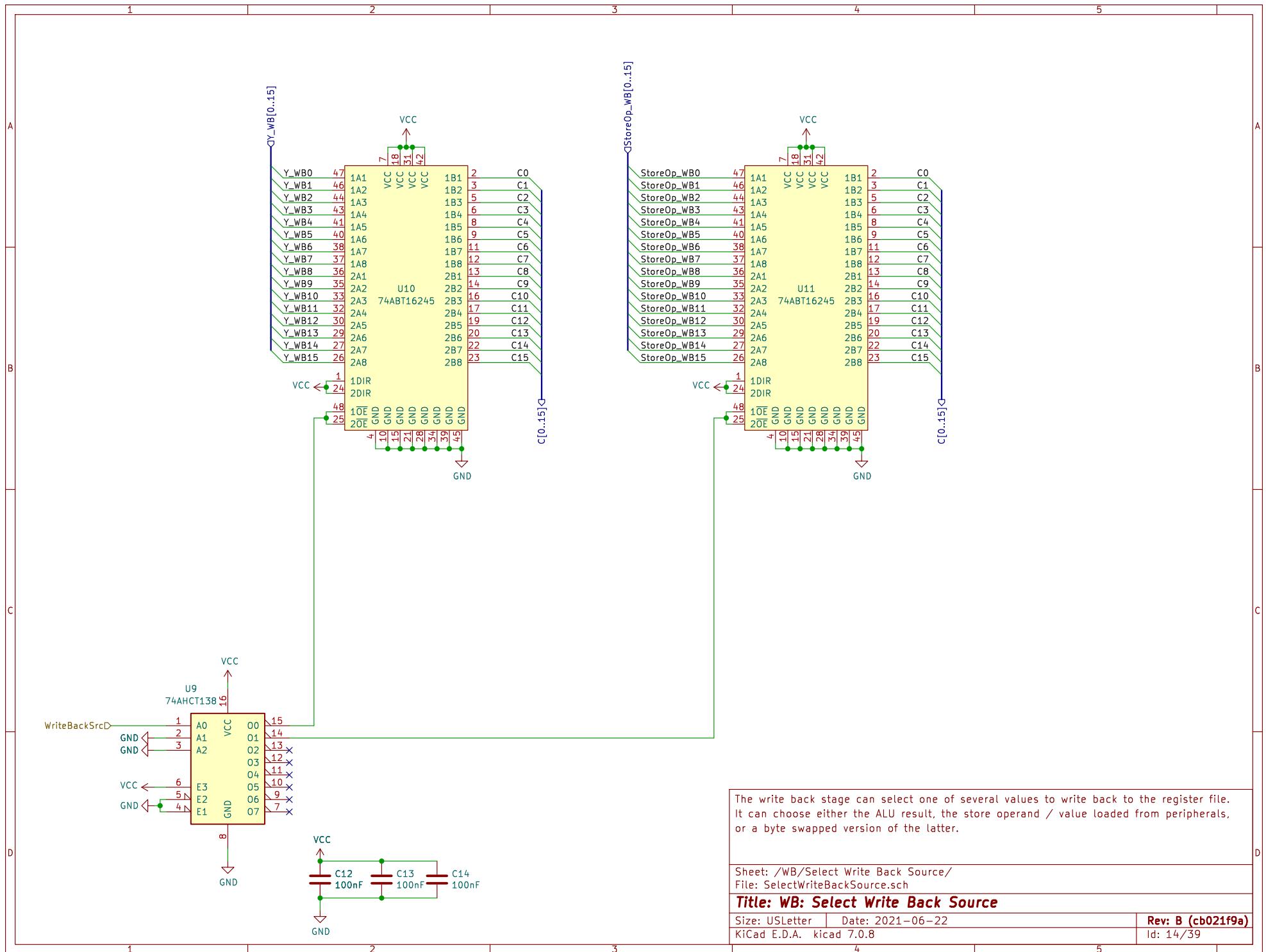
Sheet: /Clock/HaltResume/  
File: HaltResume.sch

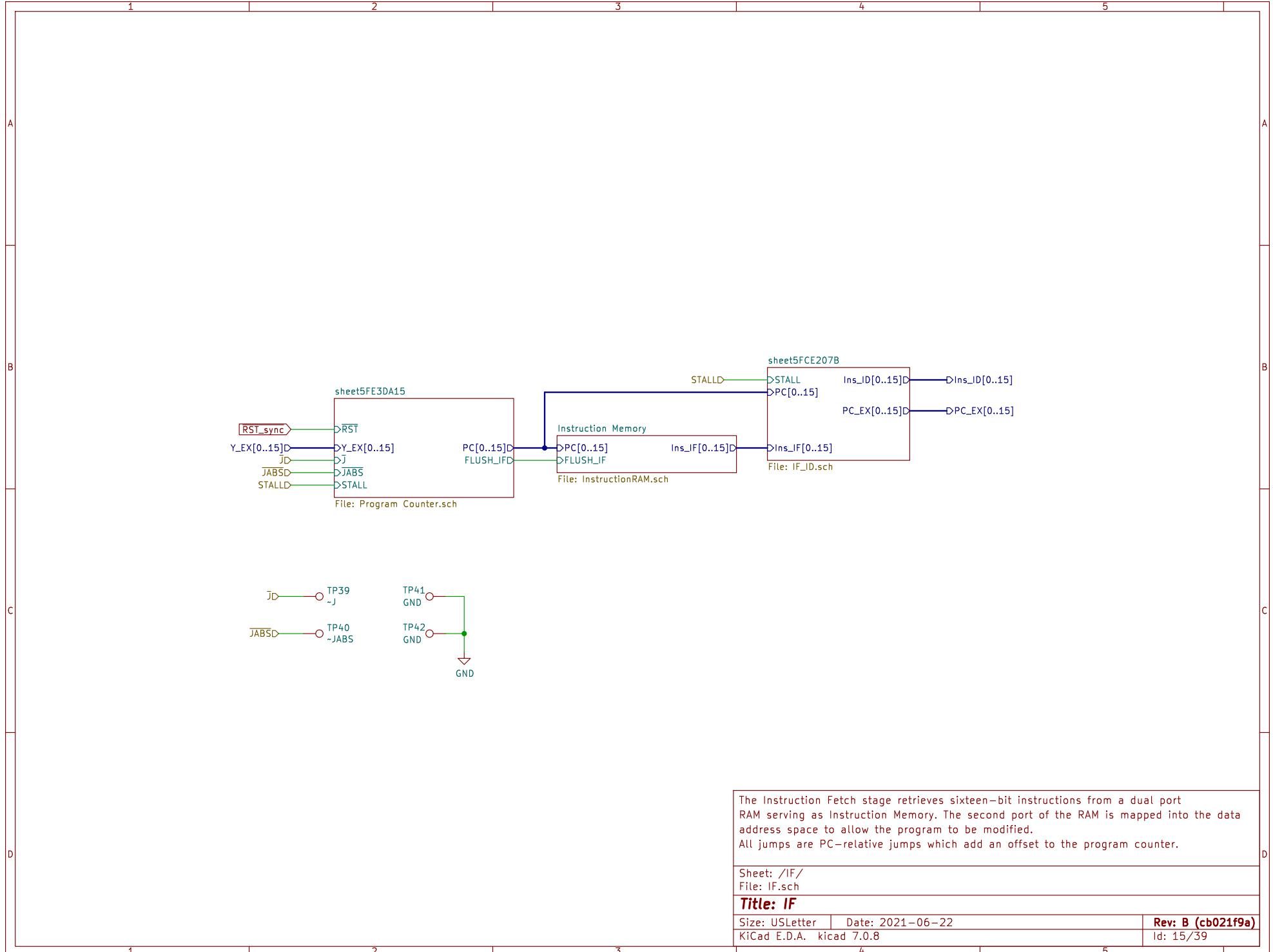
#### Title:

Size: A4 | Date: 2021-06-22  
KiCad E.D.A. kicad 7.0.8

Rev: B (cb021f9a)  
Id: 12/39







1 2 3 4 5

**A**  
Configure the ALU for FTAB=1 and FTF=0. This causes the A and B registers to be bypassed entirely. The F output updates on the next rising edge of the clock.

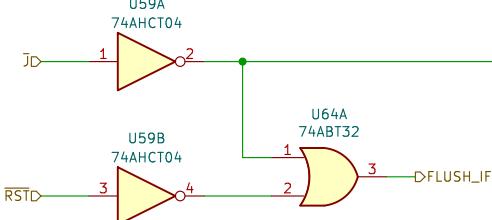
During reset, set the ALU to I2=0, I1=0, IO=0. This causes the ALU to compute a zero and latch it in A on the rising edge of the clock regardless of the value of the A and B inputs. This resets the program counter to zero.

When incrementing, set the ALU to RS1=0, RS0=1, I2=0, I1=1, IO=1, CO=1. The ALU computes  $F = A + 0 + CO$ . Since output is wired to feedback to input port A, this computes  $PC = PC + 1$ .

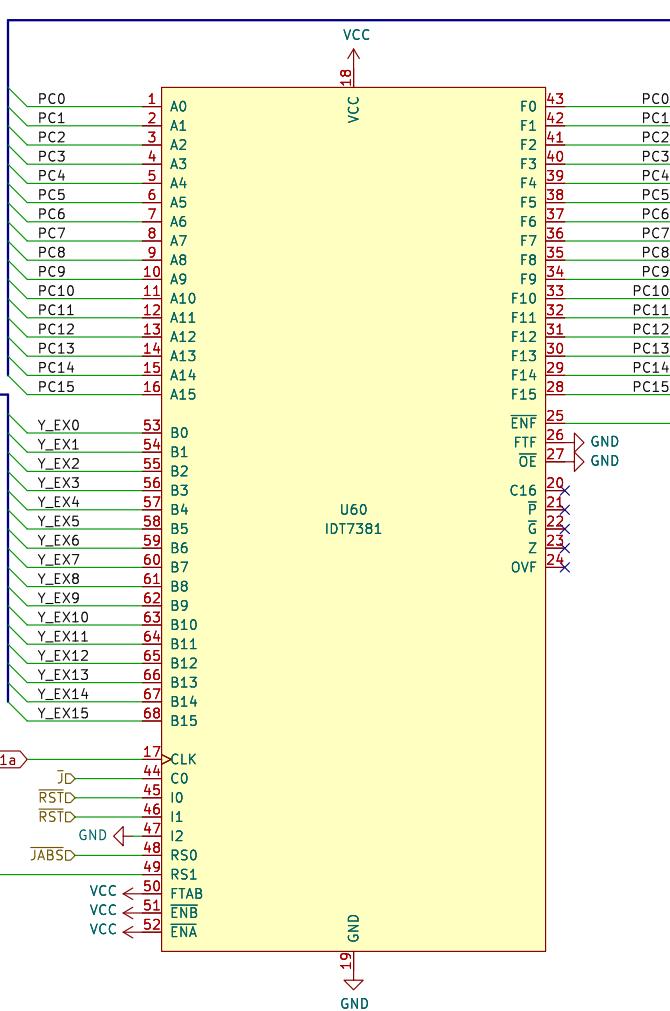
When performing a relative jump, set the ALU to RS1=1, RS0=1, I2=0, I1=1, IO=1, CO=0. The ALU computes  $F = A + B$ . Since the B port gets its value from the Y result of the EX stage, this computes  $PC = PC + offset$ .

When performing an absolute jump, set the ALU to RS1=1, RS0=0, I2=0, I1=1, IO=1, CO=0. The ALU computes  $F = 0 + B$ . Since the B port gets its value from the Y result of the EX stage, this computes  $PC = target$ .

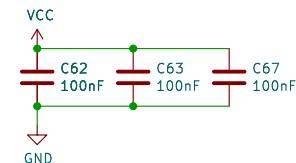
**B**  
 $Y_{EX[0..15]}$



Flush the IF stage on Jump and on Reset.  
The flush during Reset ensures the pipeline is filled with NOPs.



Stall the program counter by disabling the update of the IDT7381's F register.



Sixteen-bit program counter will either increment on the clock, add a specified sixteen-bit offset, or else reset to zero.

Sheet: /IF/sheet5FE3DA15/  
File: Program Counter.sch

**Title: Program Counter**

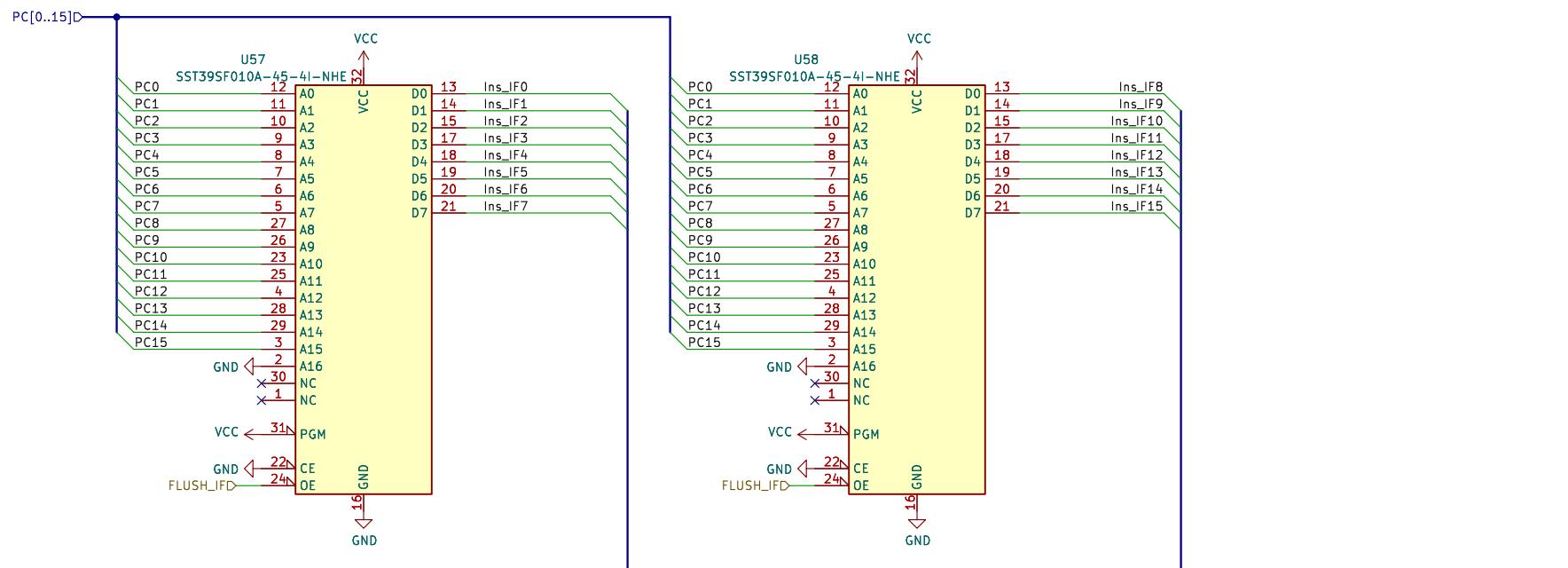
Size: USLetter Date: 2021-06-22

KiCad E.D.A. kicad 7.0.8

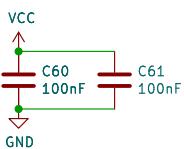
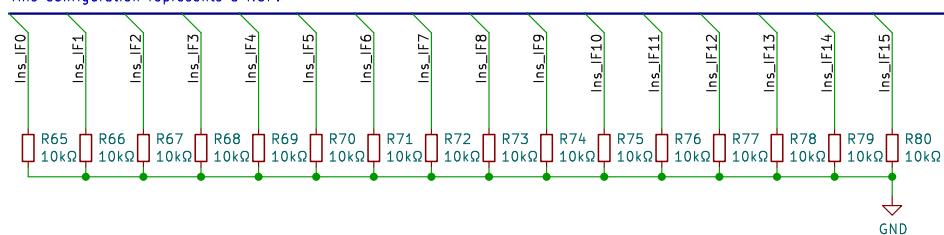
Rev: B (cb021f9a)

Id: 16/39

1 2 3 4 5



When the IF stage is flushed, all the output lines are pulled low.  
This configuration represents a NOP.



Instructions are stored in a pair of EEPROMs in ZIF sockets.

Sheet: /IF/Instruction Memory/  
File: InstructionRAM.sch

### Title: Instruction ROM

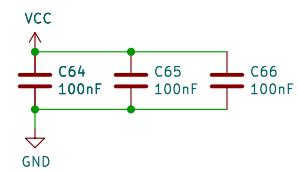
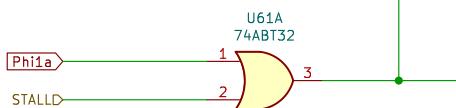
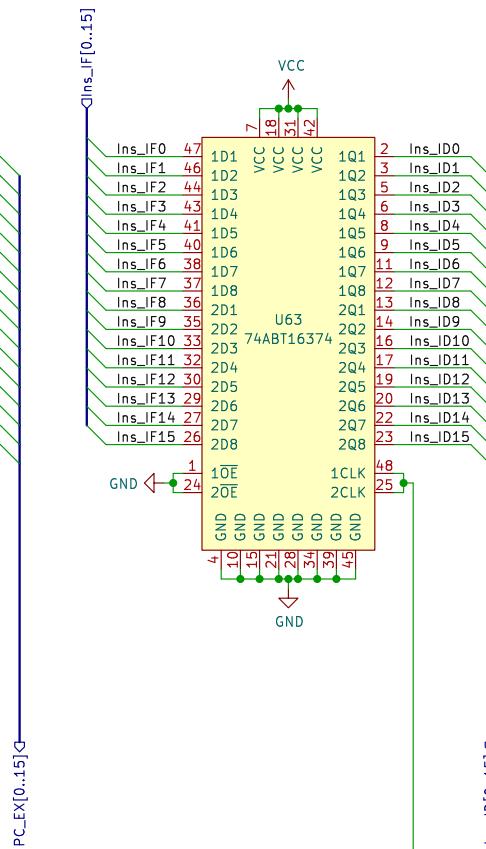
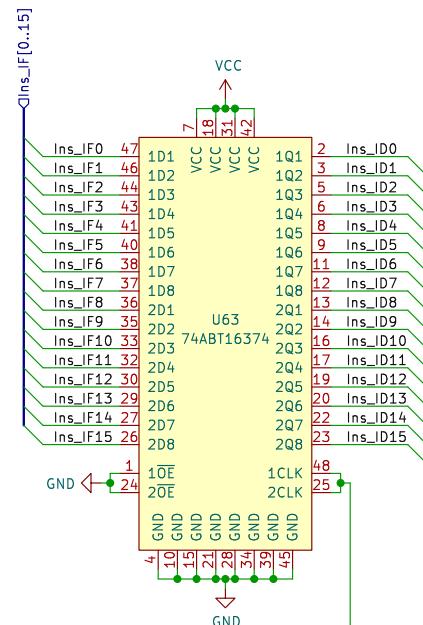
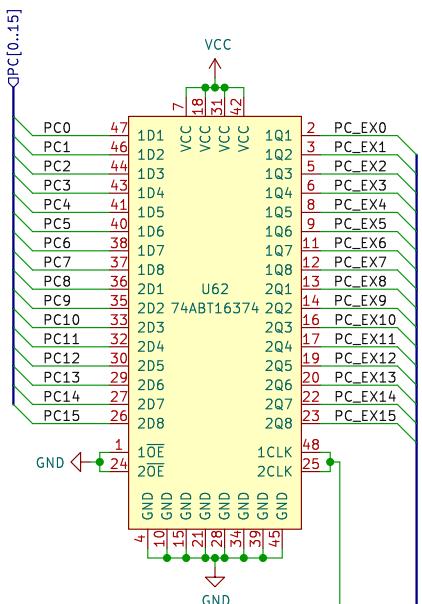
Size: USLetter | Date: 2021-06-22

KiCad E.D.A. kicad 7.0.8

Rev: B (cb021f9a)

Id: 17/39

A



#### Interstage pipeline registers between IF and ID

Sheet: /IF/sheet5FCE207B/

File: IF\_ID.sch

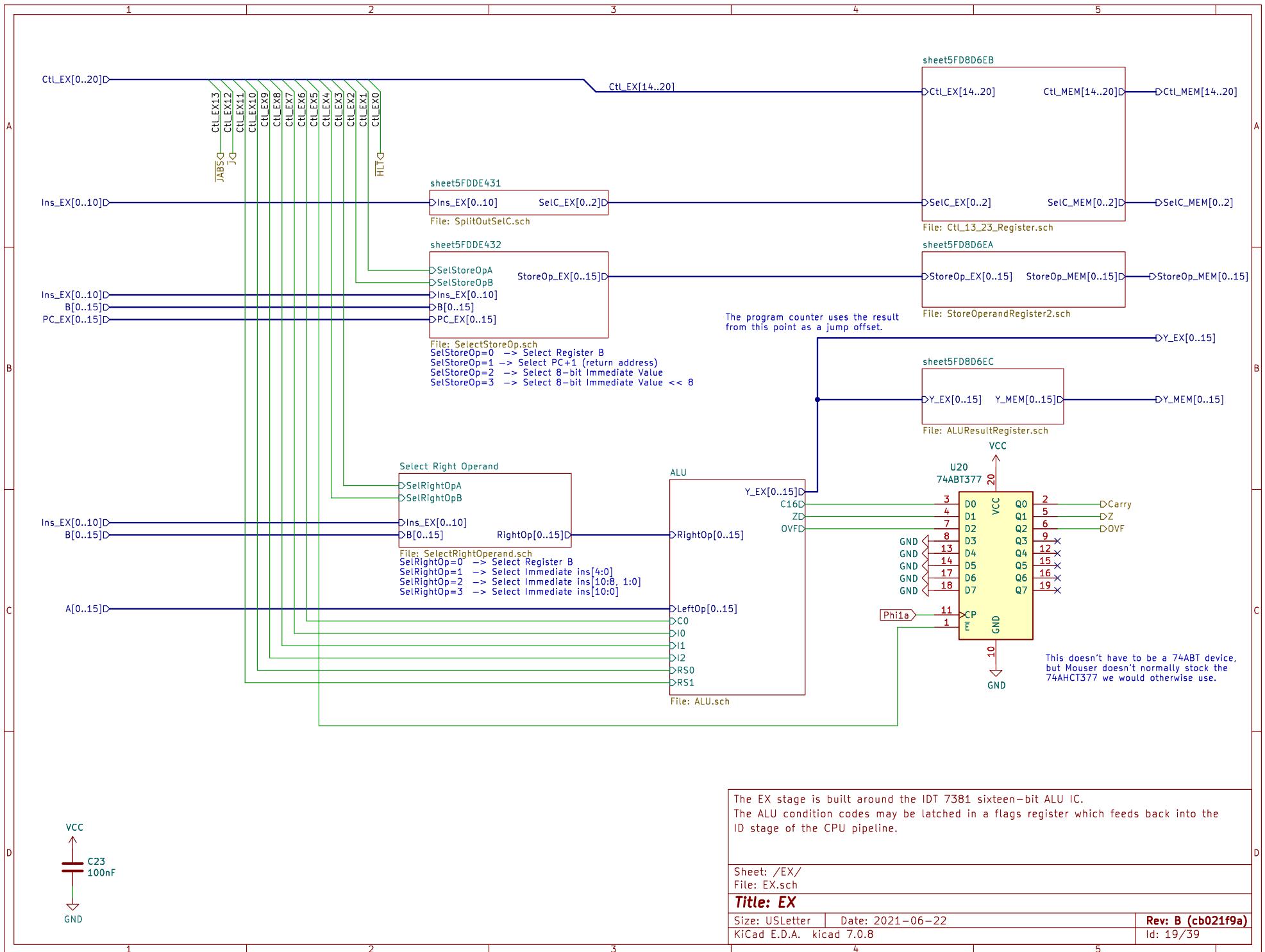
**Title: IF/ID**

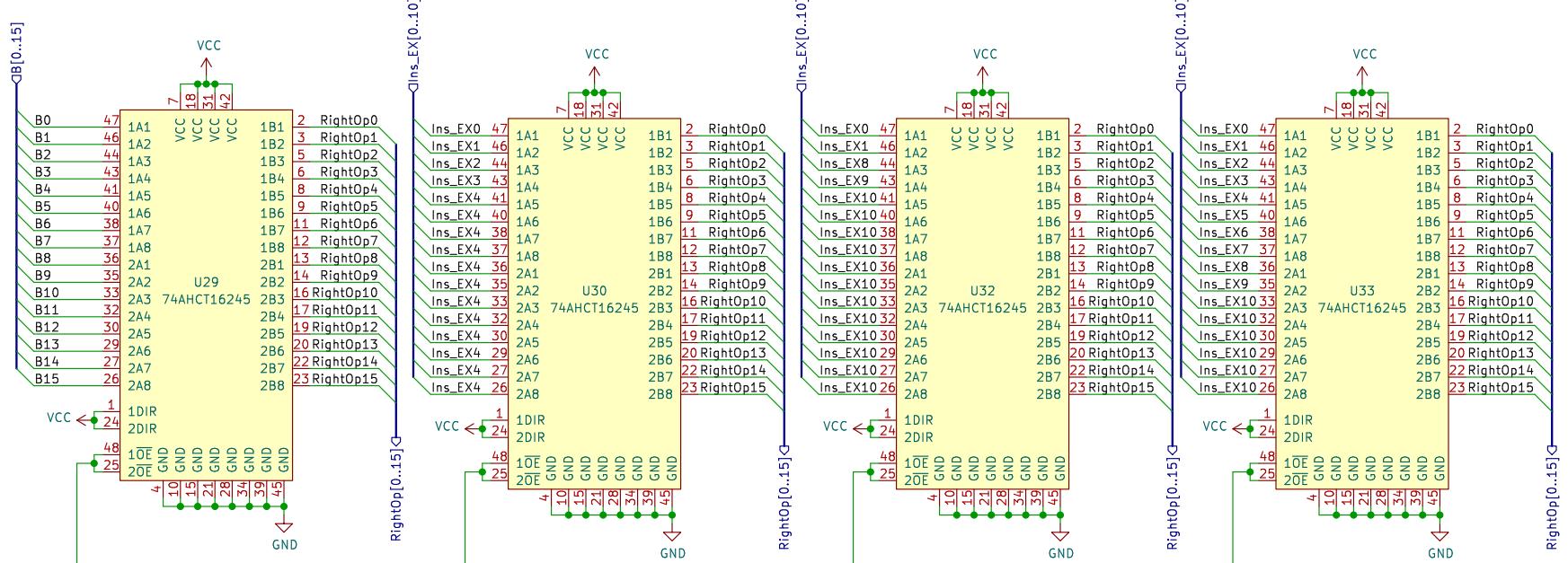
Size: USLetter Date: 2021-06-22

KiCad E.D.A. kicad 7.0.8

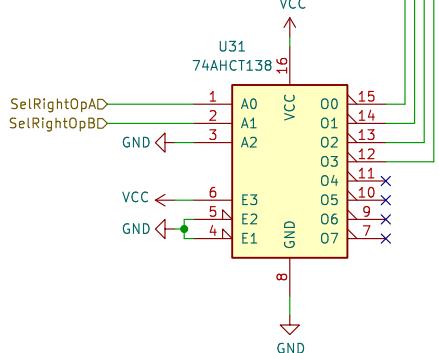
**Rev: B (cb021f9a)**

Id: 18/39





SelRightOp=0 → Select Register B  
 SelRightOp=1 → Select Immediate ins[4:0]  
 SelRightOp=2 → Select Immediate ins[10:8, 1:0]  
 SelRightOp=3 → Select Immediate ins[10:0]



Select the right operand to the ALU  
 The right operand can either take the value from the B port of the register file, or from an immediate value contained in the instruction word.

Sheet: /EX>Select Right Operand/  
 File: SelectRightOperand.sch

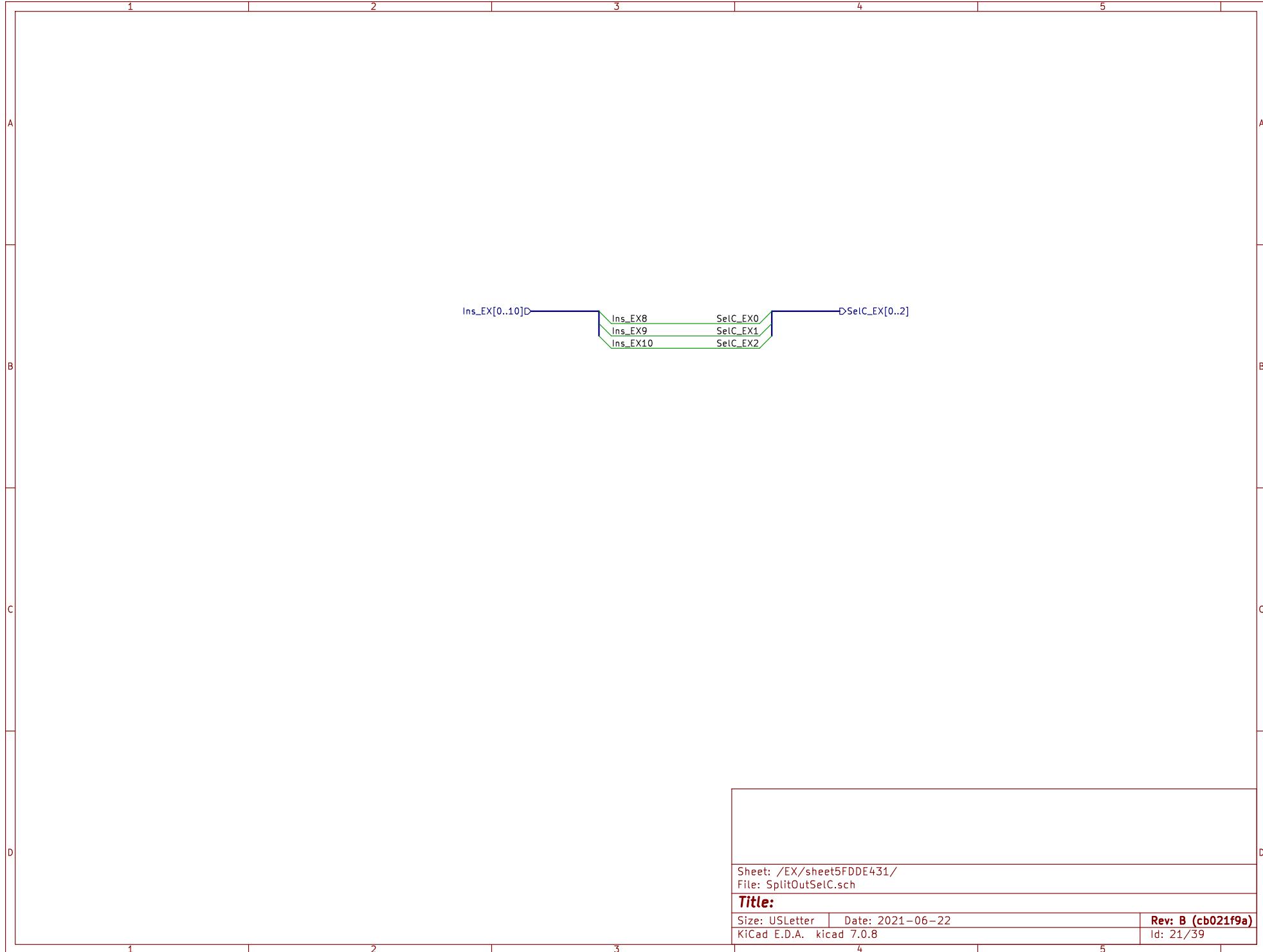
### Title: Select Right Operand

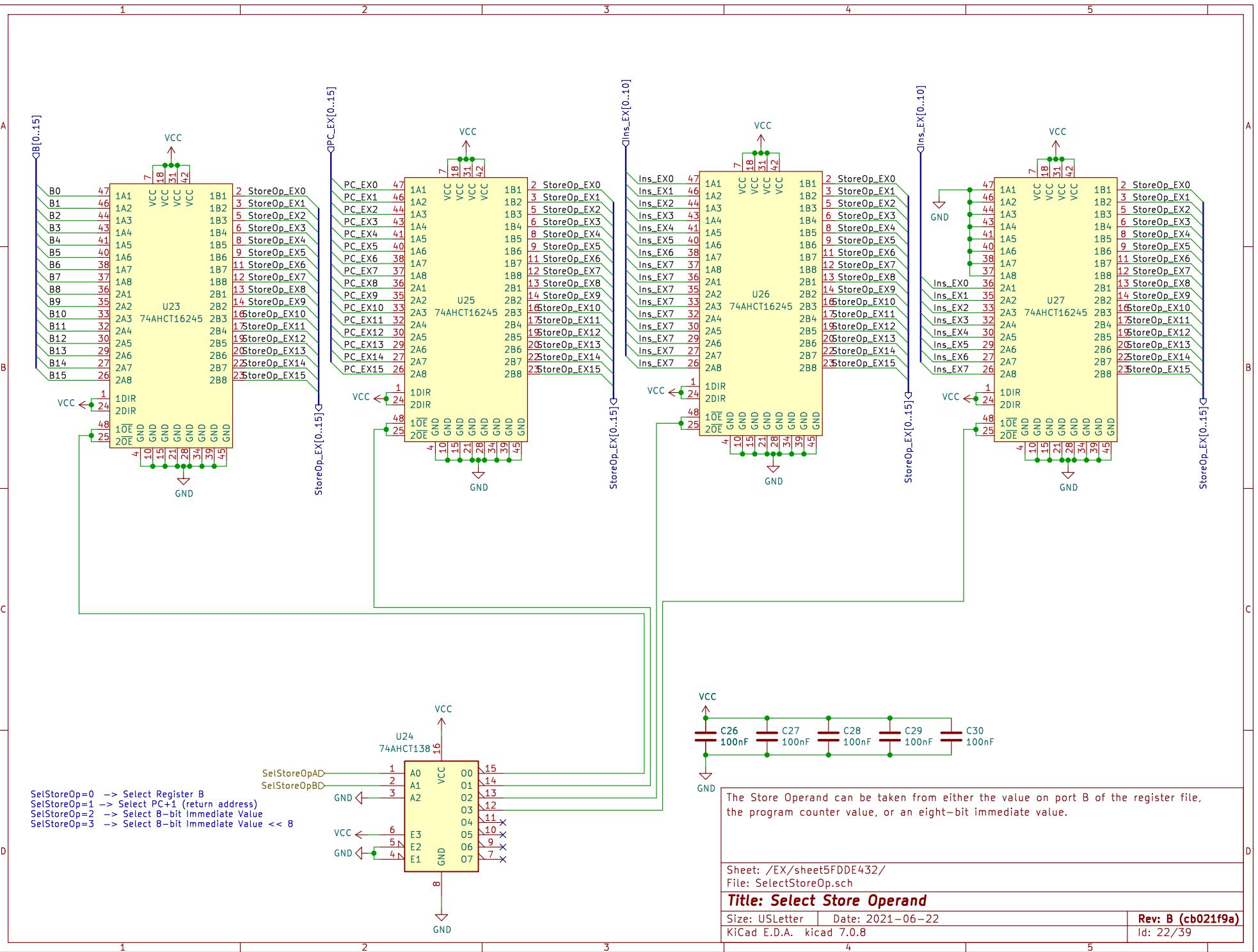
Size: USLetter Date: 2021-06-22

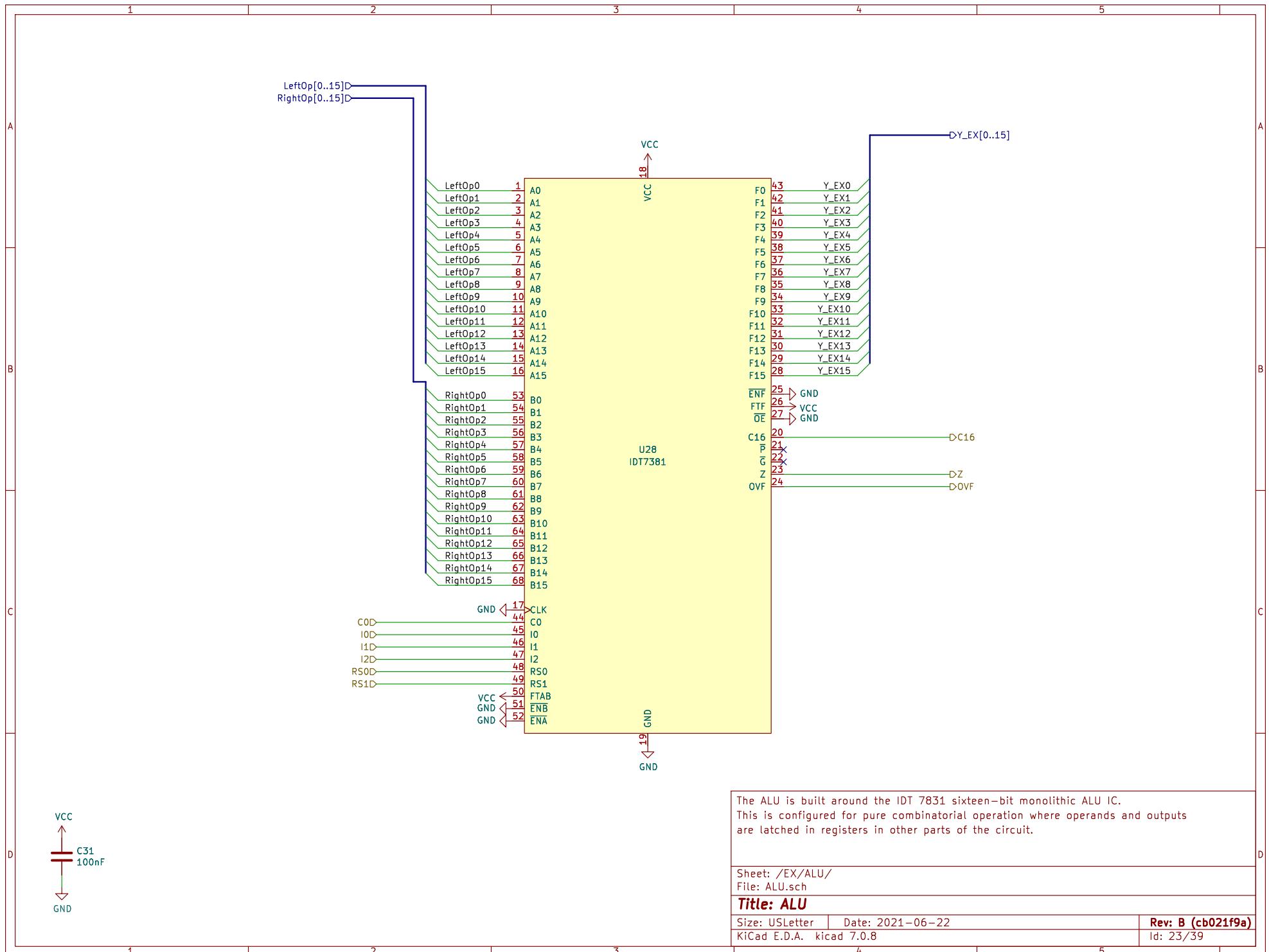
KiCad E.D.A. kicad 7.0.8

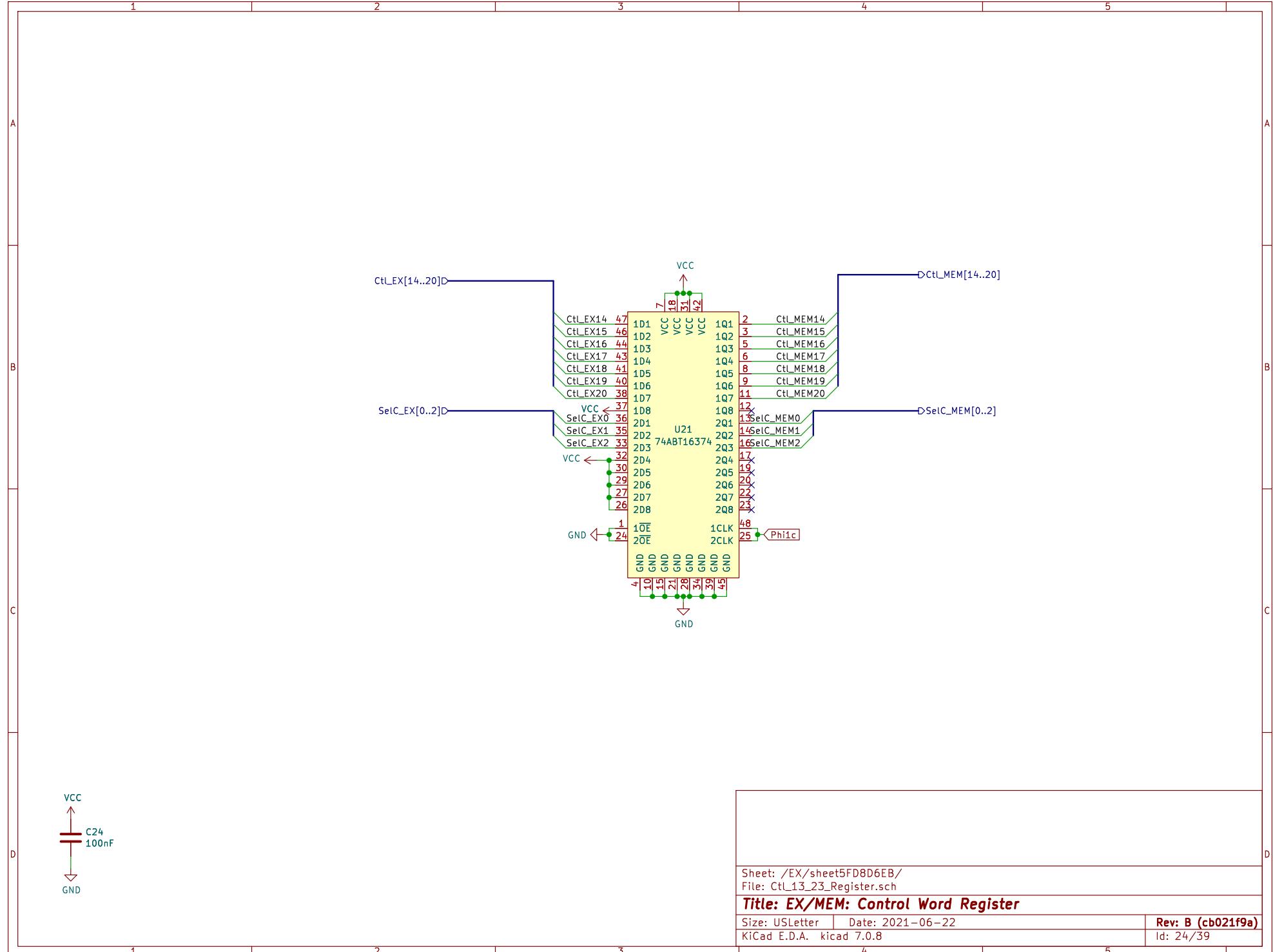
Rev: B (cb021f9a)

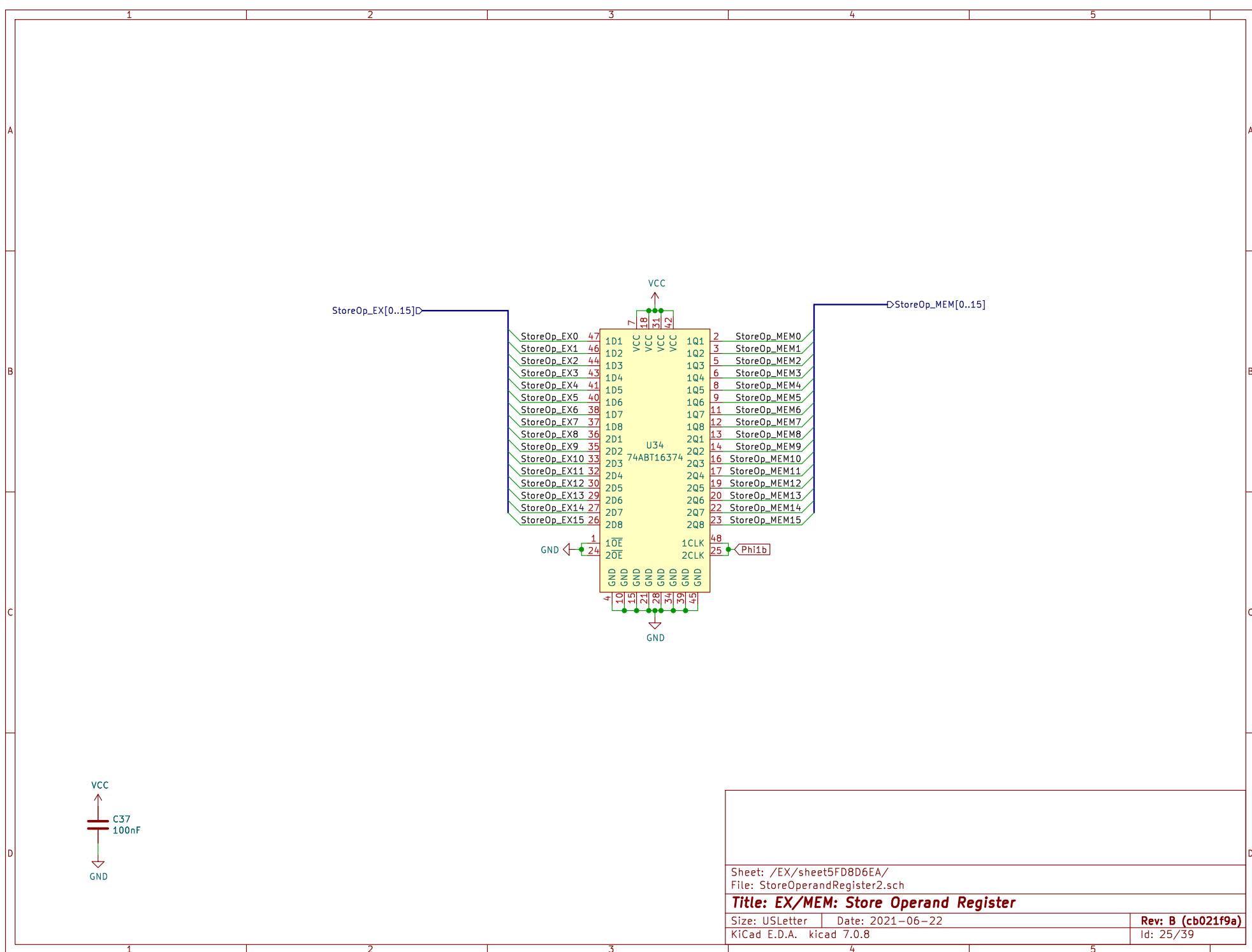
Id: 20/39

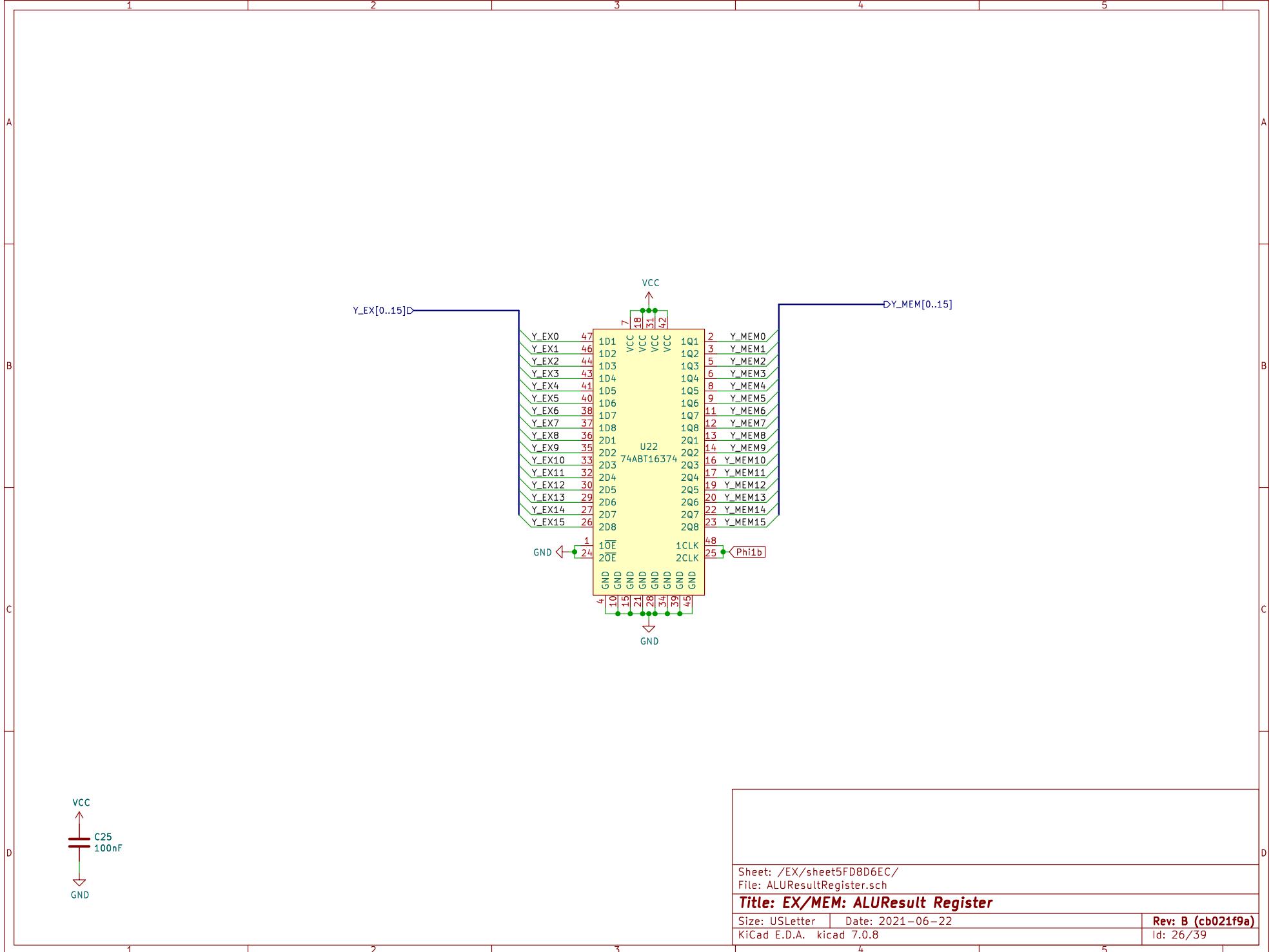


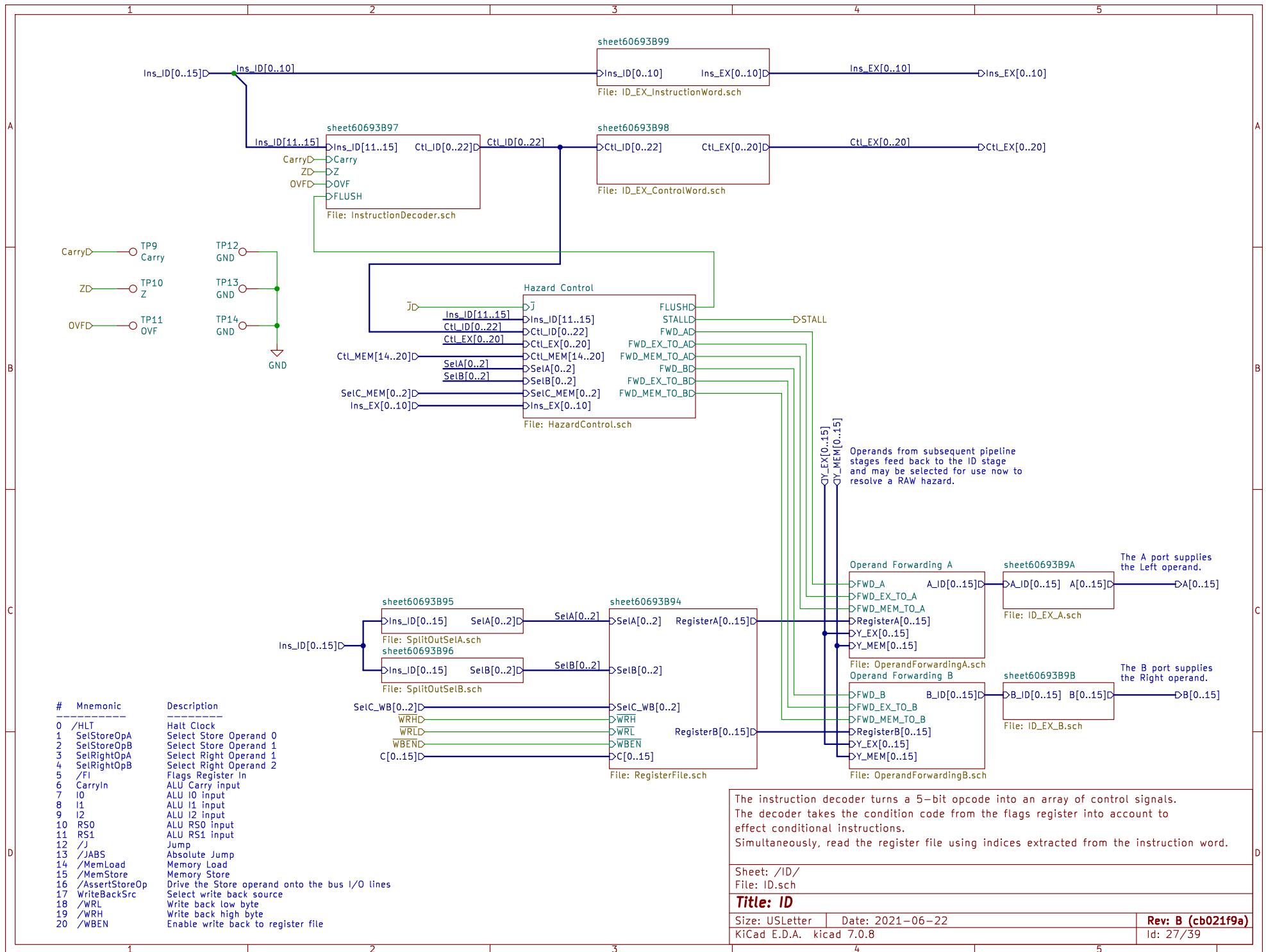


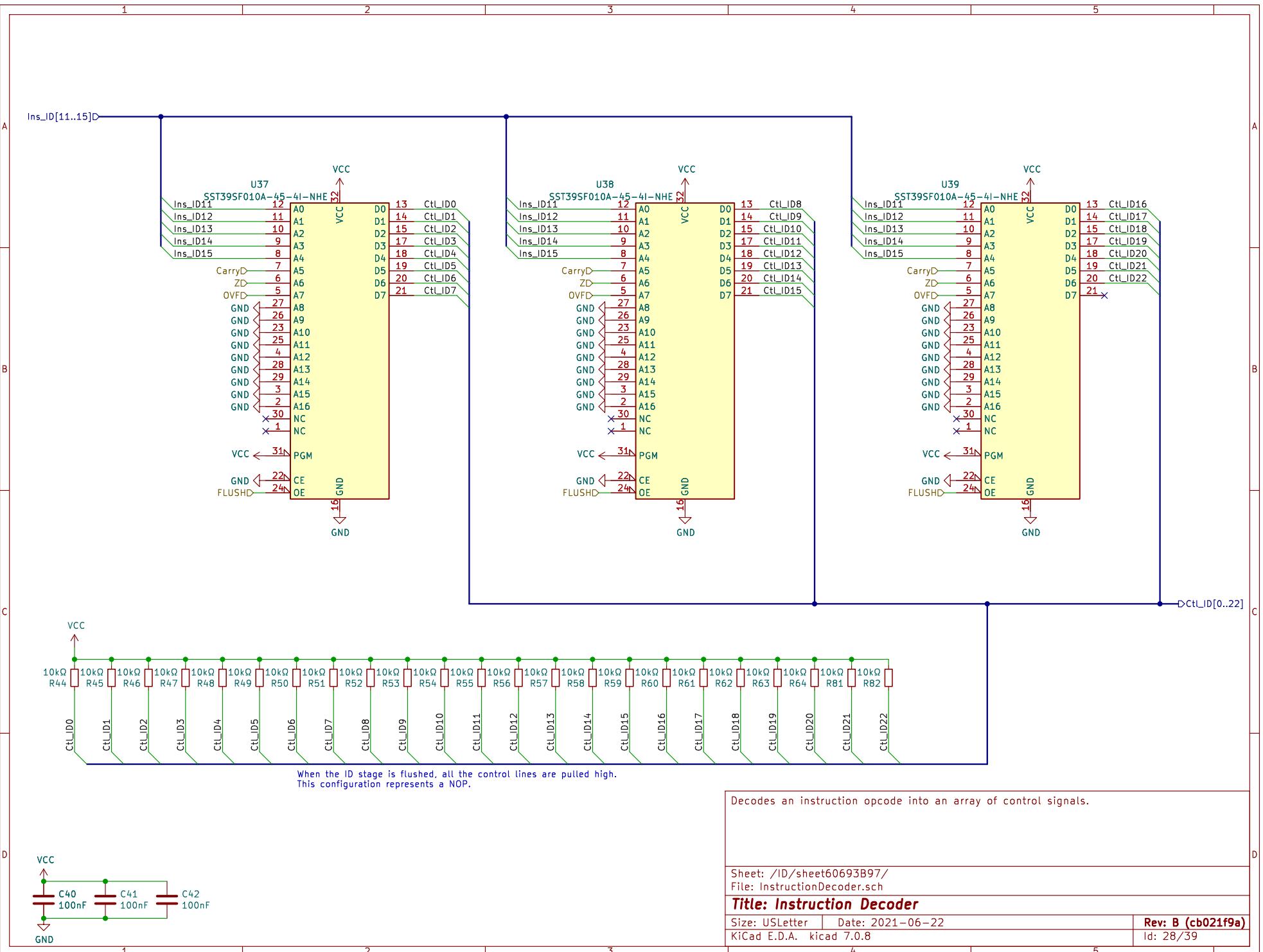


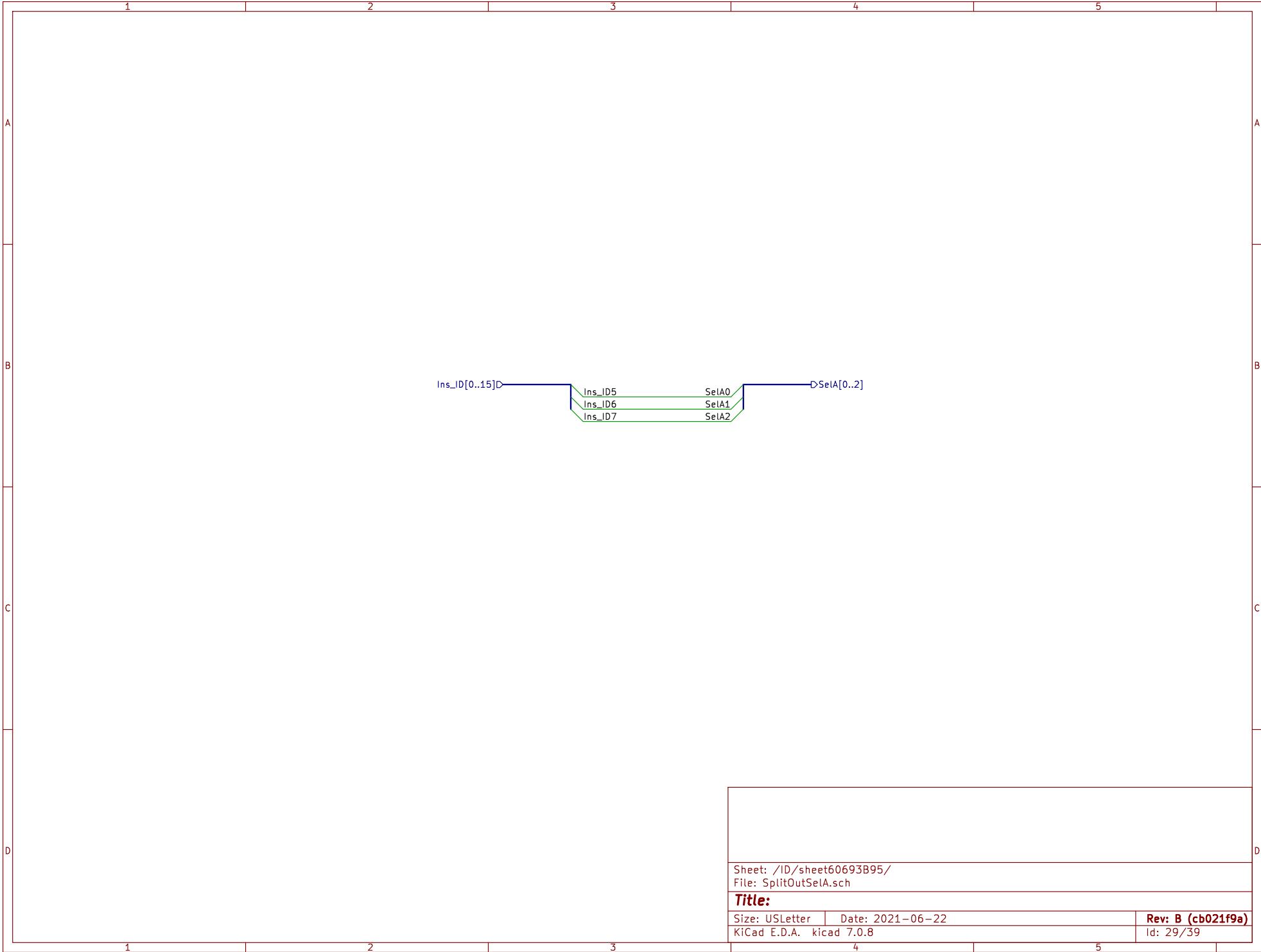


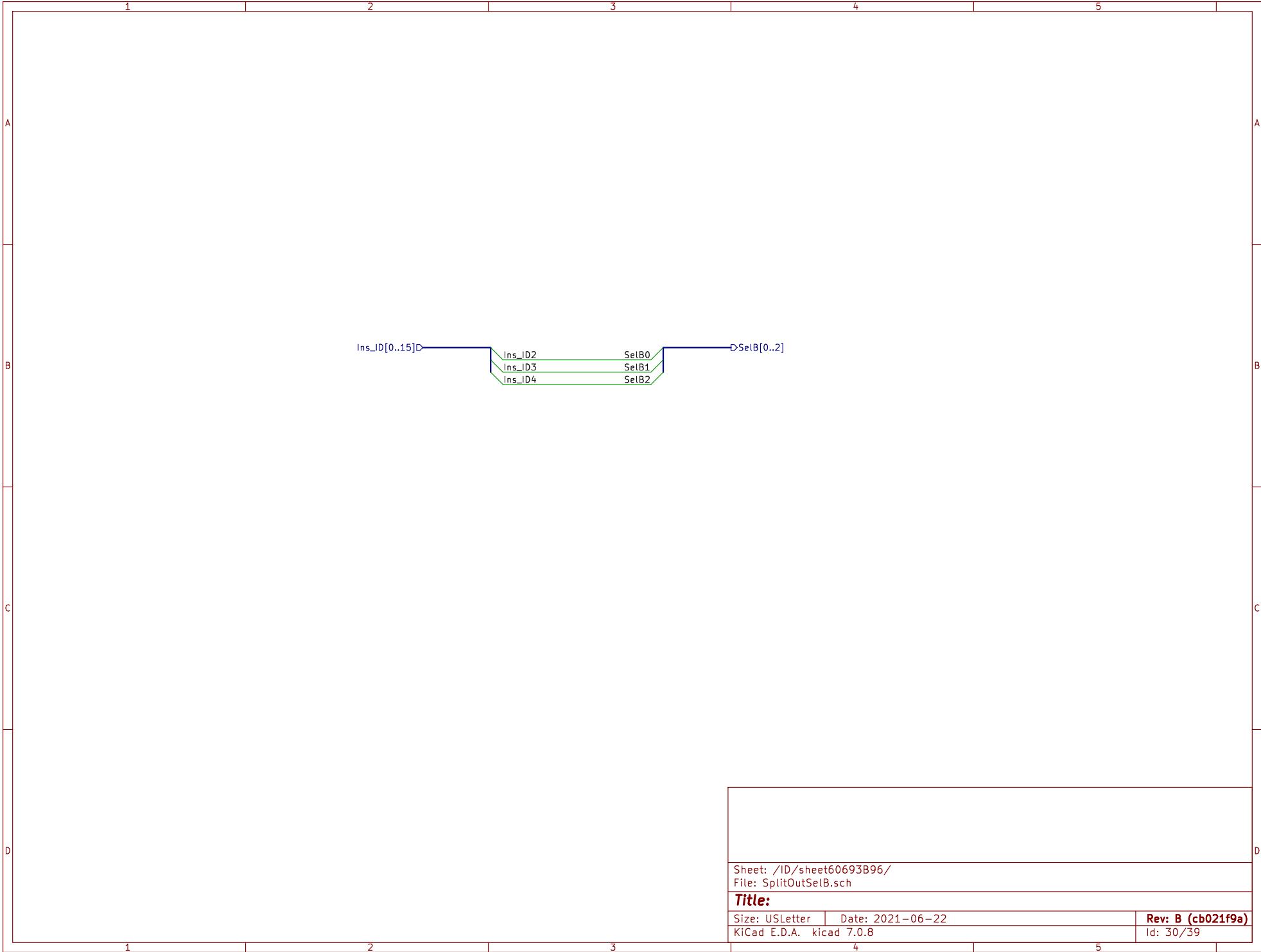












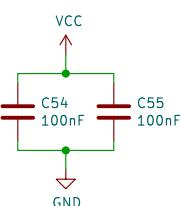
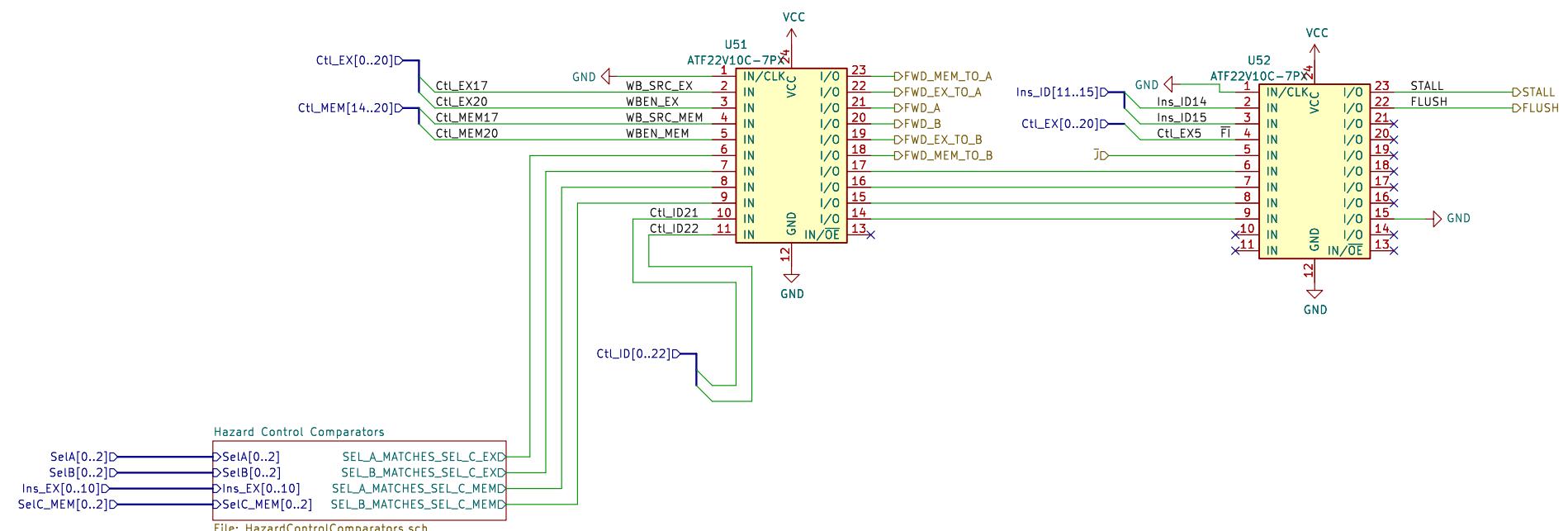
Hazard Control detects several types of hazards and resolves each one either by stalling the pipeline, or by forwarding an operand from a later pipeline stage.

On a jump, the program counter must be allowed to load a new value. At the same time, the ID and IF stages must both be flushed.

If the instruction in EX wants to update the Flags register, and the opcode in ID is determined to be one that wants to make use of the flags, then there is a Flags hazard. In this case, stall the pipeline for one cycle.

If the Ra or Rb registers refer to the destination register in the EX or MEM stage, and the instructions in the EX or MEM stage want to write back to that register, then there is a RAW error. In this case, forward an operand from a later pipeline stage to supply the EX stage on the next clock cycle.

There's no path to forward the store operand. Cases involving this operand must be resolved by stalling the pipeline.



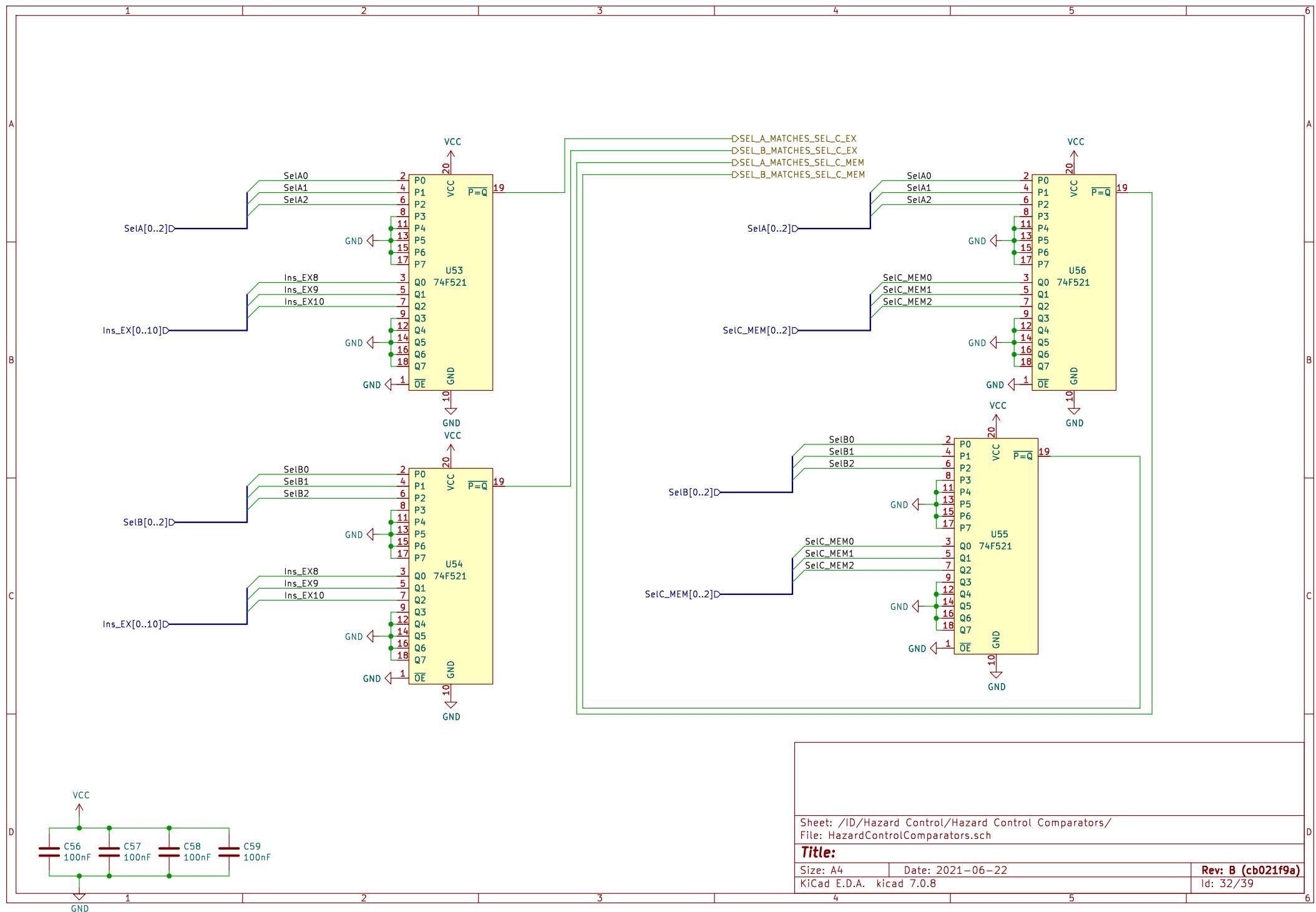
Control logic for dealing with pipeline hazards  
This may stall the pipeline on a hazard. For RAW hazards, it produces signals to control operand forwarding.

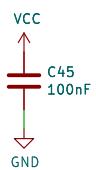
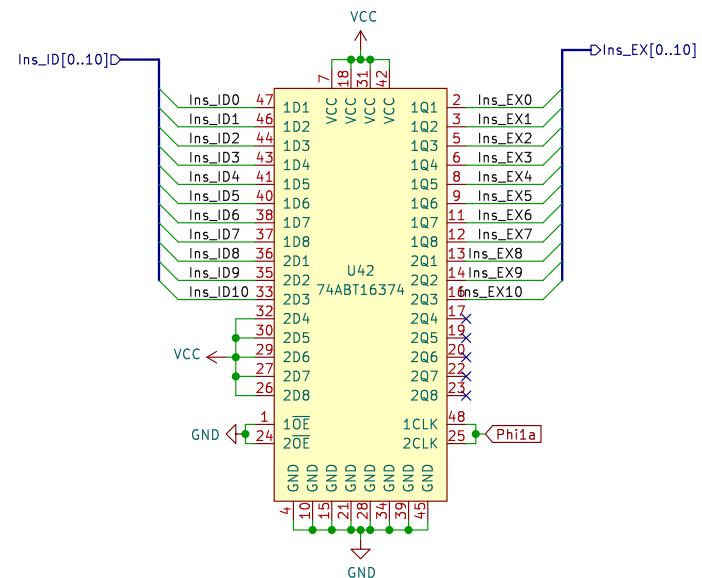
Sheet: /ID/Hazard Control/  
File: HazardControl.sch

### Title: Hazard Control

Size: USLetter | Date: 2021-06-22  
KiCad E.D.A. kicad 7.0.8

Rev: B (cb021f9a)  
Id: 31/39



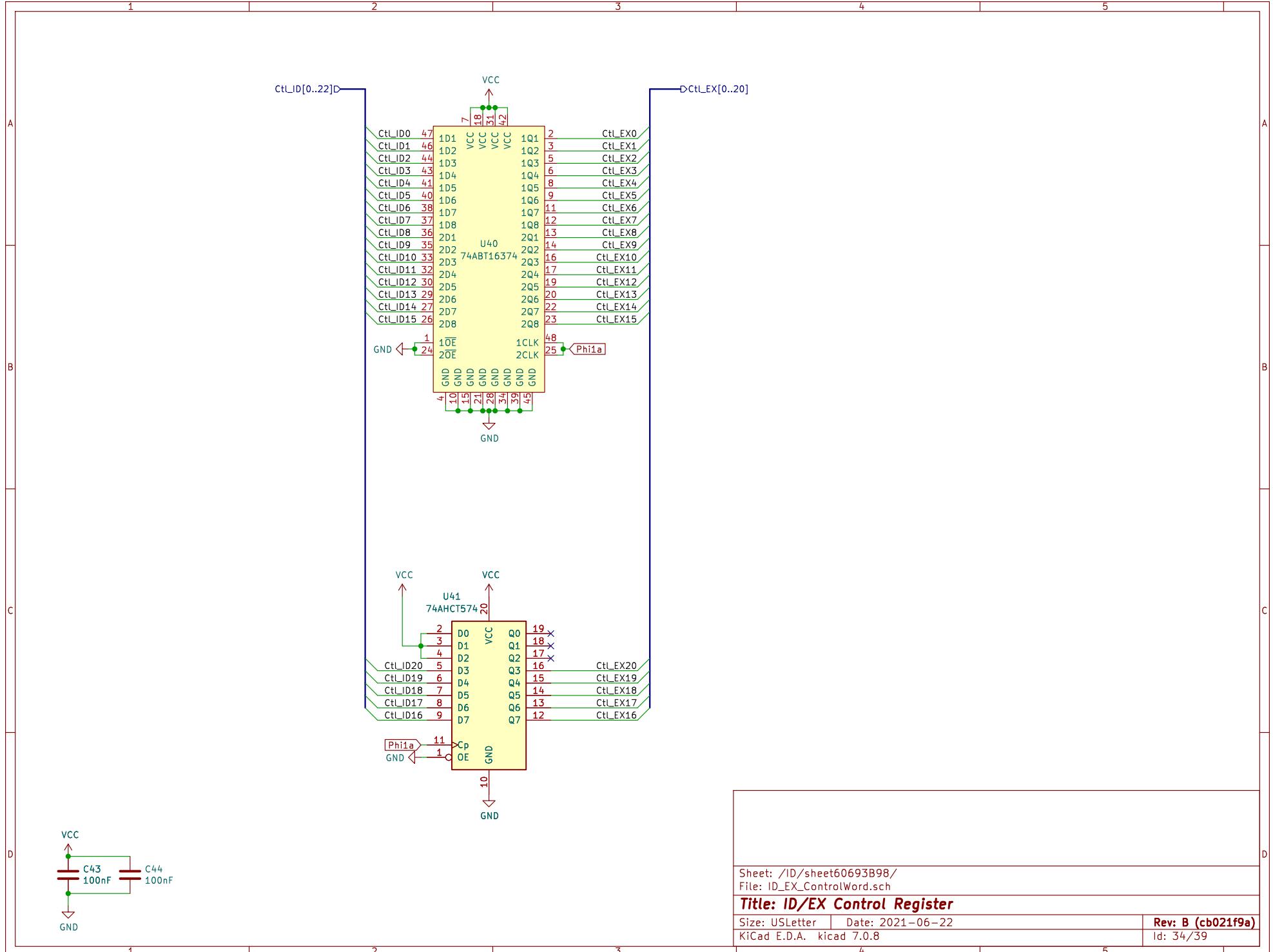


Sheet: /ID/sheet60693B99/  
File: ID\_EX\_InstructionWord.sch

**Title:**

Size: USLetter | Date: 2021-06-22  
KiCad E.D.A. kicad 7.0.8

Rev: B (cb021f9a)  
Id: 33/39



Sheet: /ID/sheet60693B98/  
File: ID\_EX\_ControlWord.sch

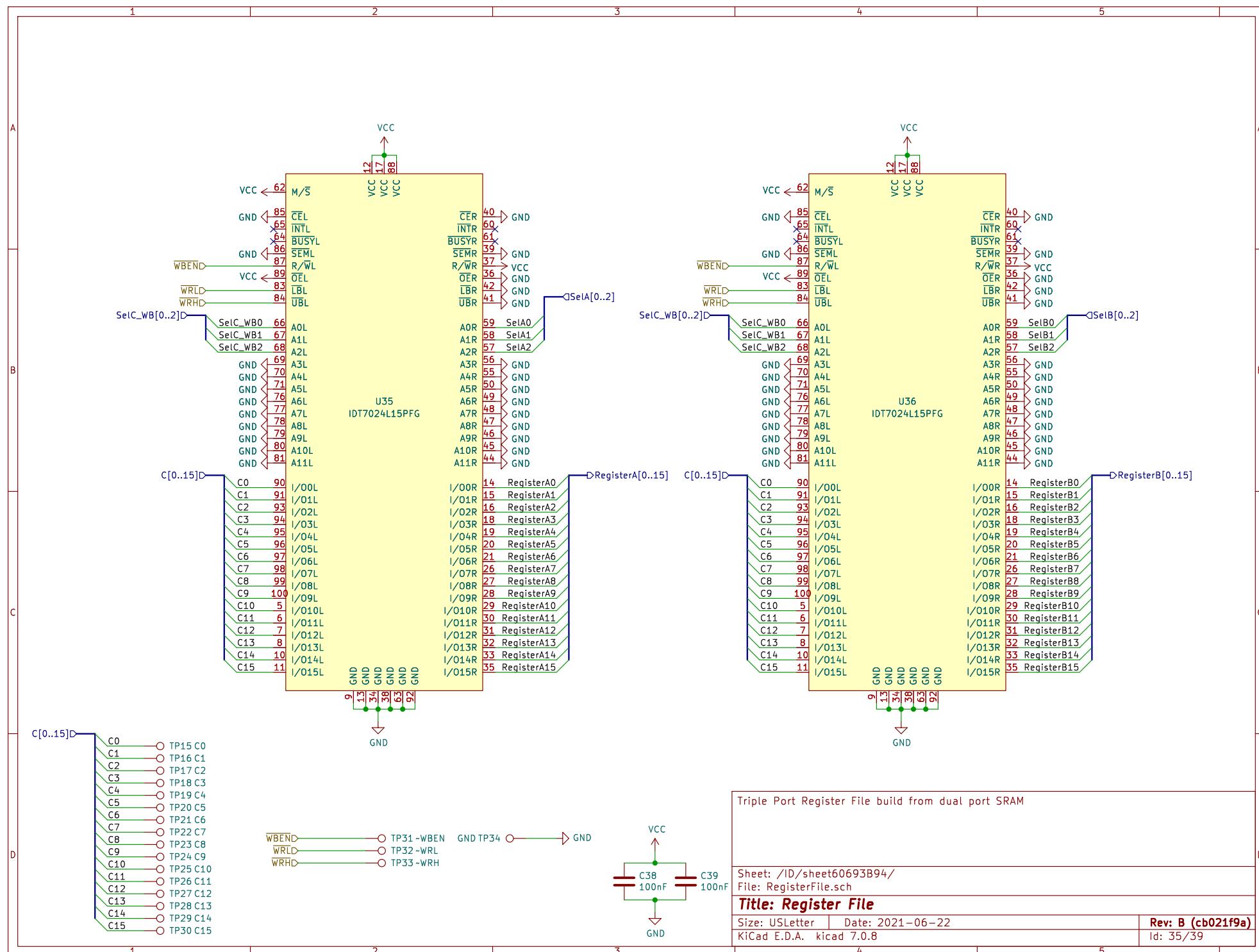
### Title: ID/EX Control Register

Size: USLetter Date: 2021-06-22

KiCad E.D.A. kicad 7.0.8

Rev: B (cb021f9a)

Id: 34/39



A

B

C

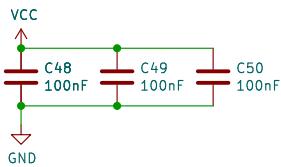
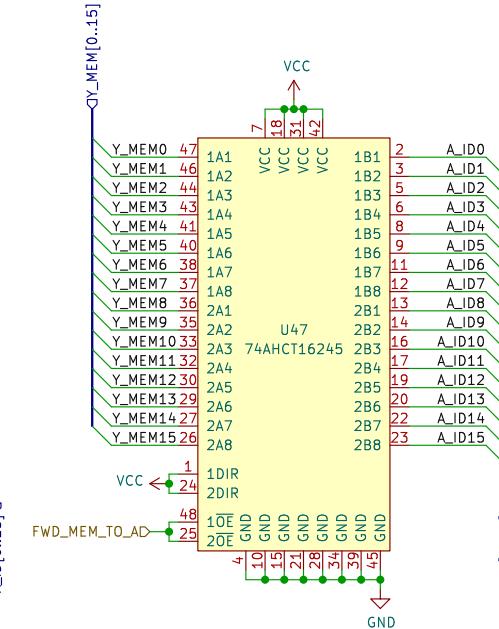
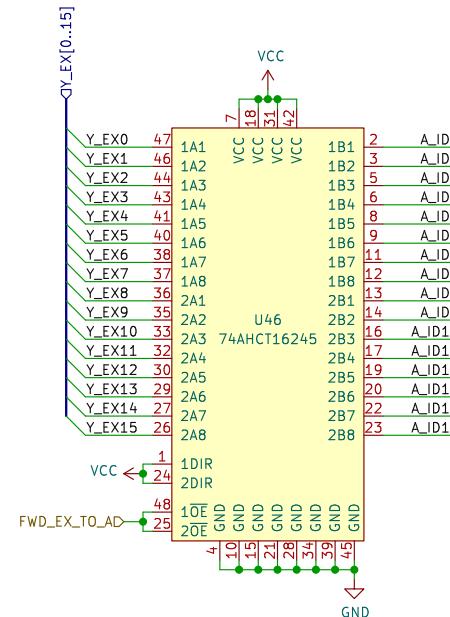
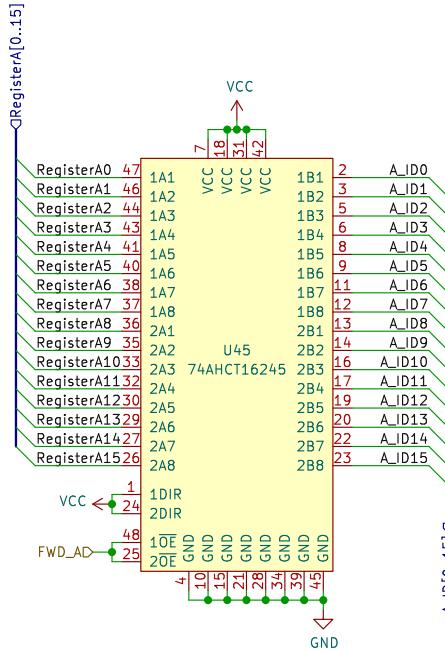
D

A

B

C

D



Sheet: /ID/Operand Forwarding A/  
File: OperandForwardingA.sch

**Title:**

Size: A4 | Date: 2021-06-22  
KiCad E.D.A. kicad 7.0.8

Rev: B (cb021f9a)  
Id: 36/39

A

B

C

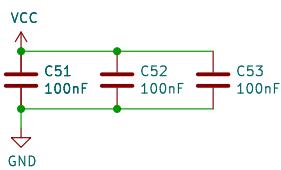
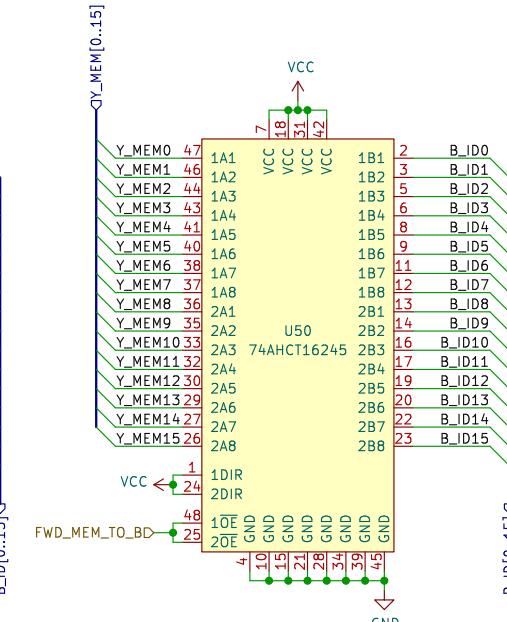
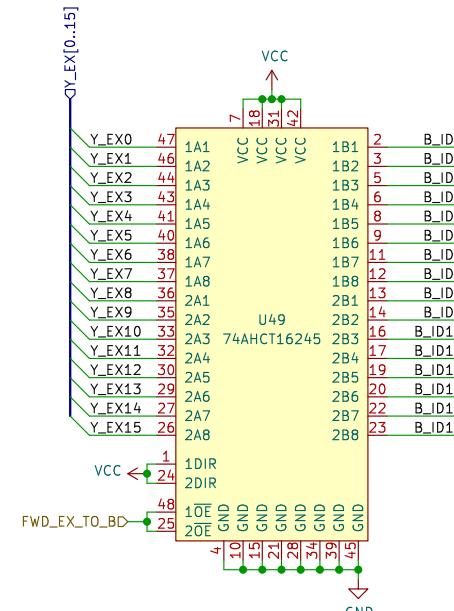
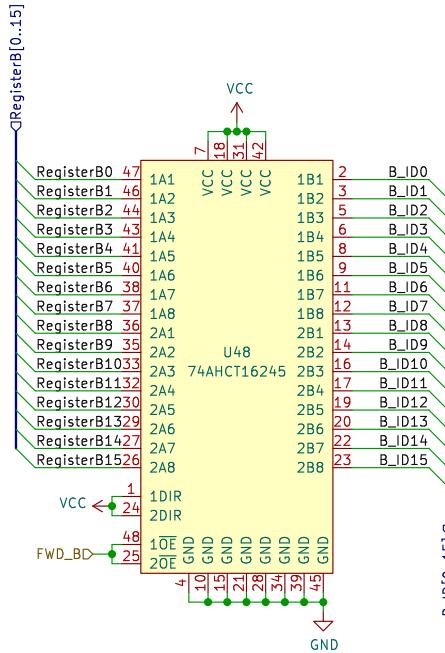
D

A

B

C

D

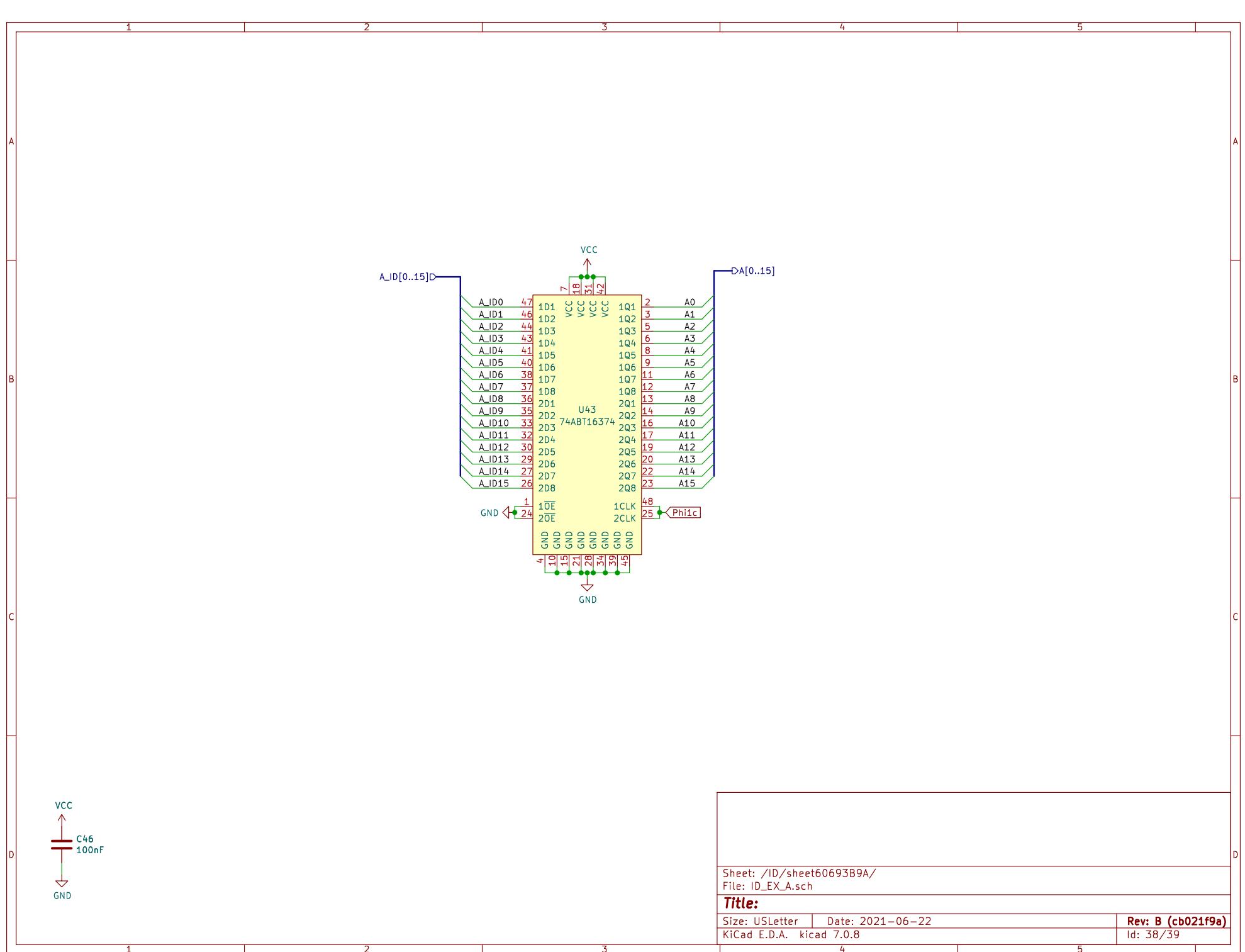


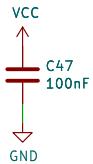
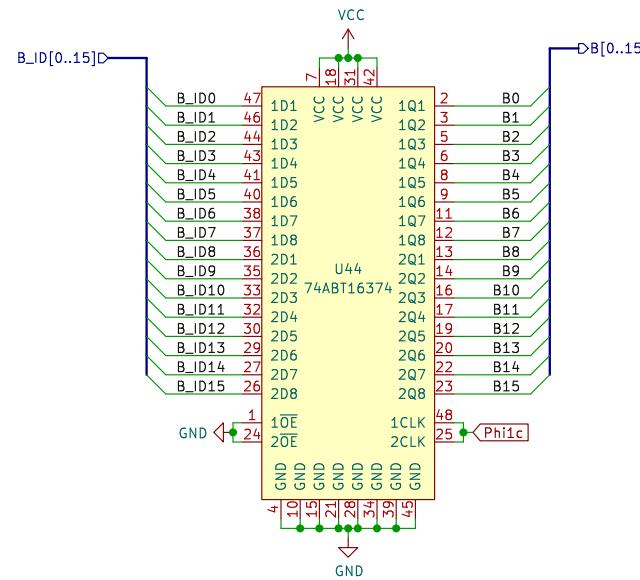
Sheet: /ID/Operand Forwarding B/  
File: OperandForwardingB.sch

**Title:**

Size: A4 | Date: 2021-06-22  
KiCad E.D.A. kicad 7.0.8

Rev: B (cb021f9a)  
Id: 37/39





Sheet: /ID/sheet60693B9B/  
File: ID\_EX\_B.sch

**Title:**

Size: USLetter | Date: 2021-06-22  
KiCad E.D.A. kicad 7.0.8

Rev: B (cb021f9a)  
Id: 39/39