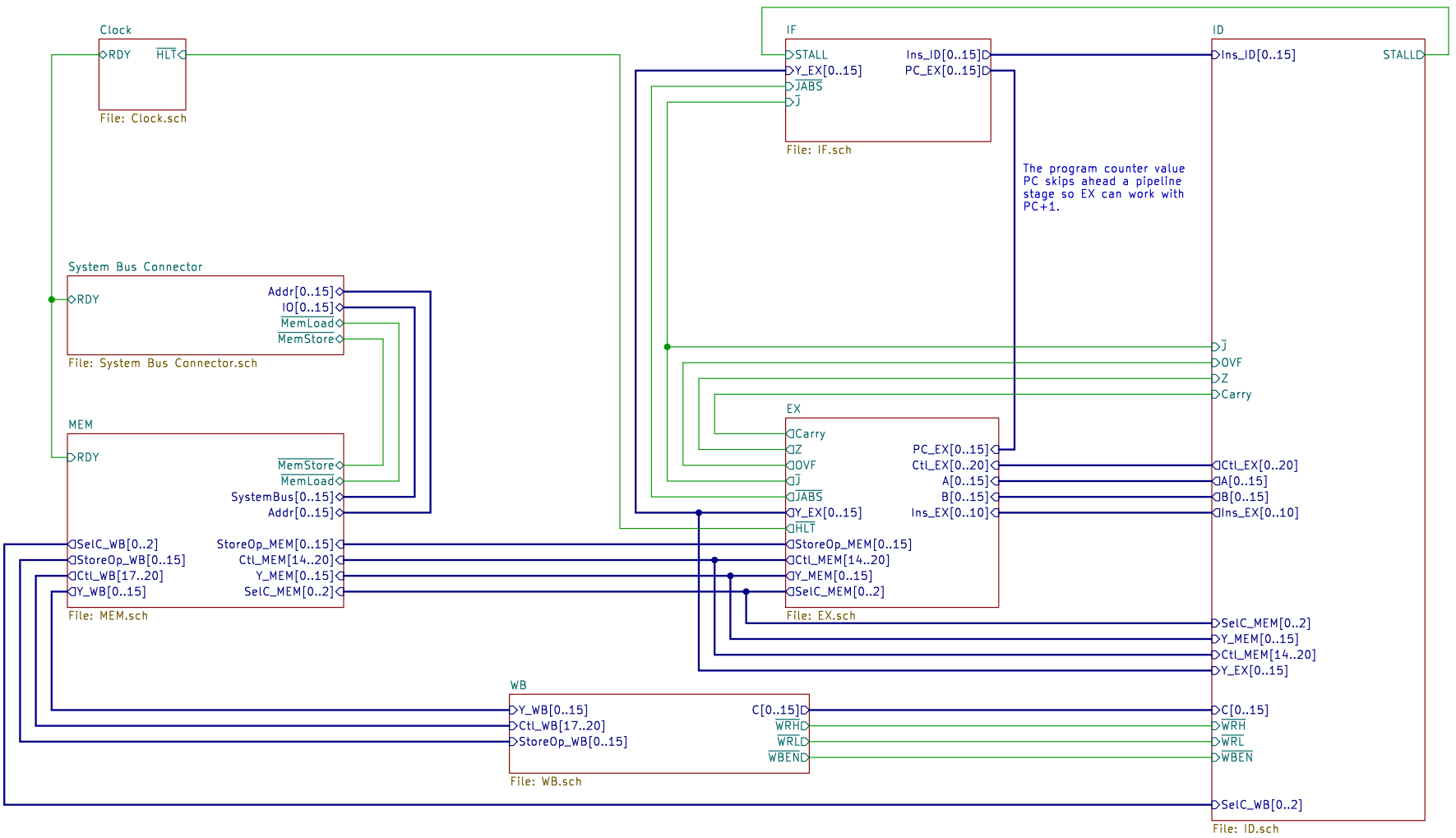


Microcomputer built from 74xx series logic chips.

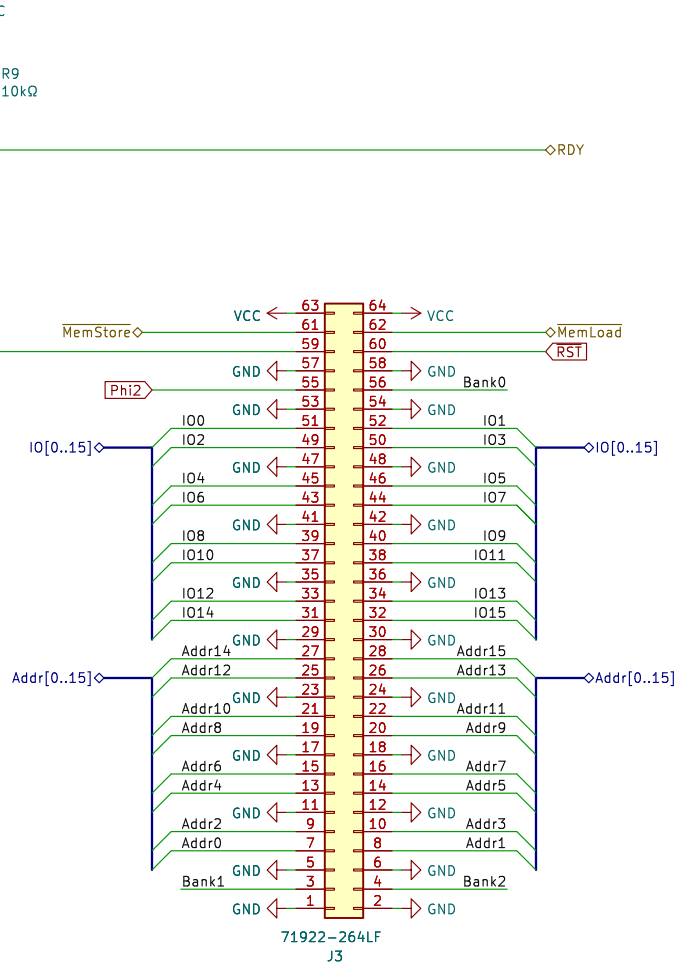
Sheet: /
File: MainBoard.sch

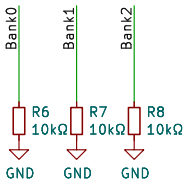
Title: Turtle16: Main Board

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 1/39

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.



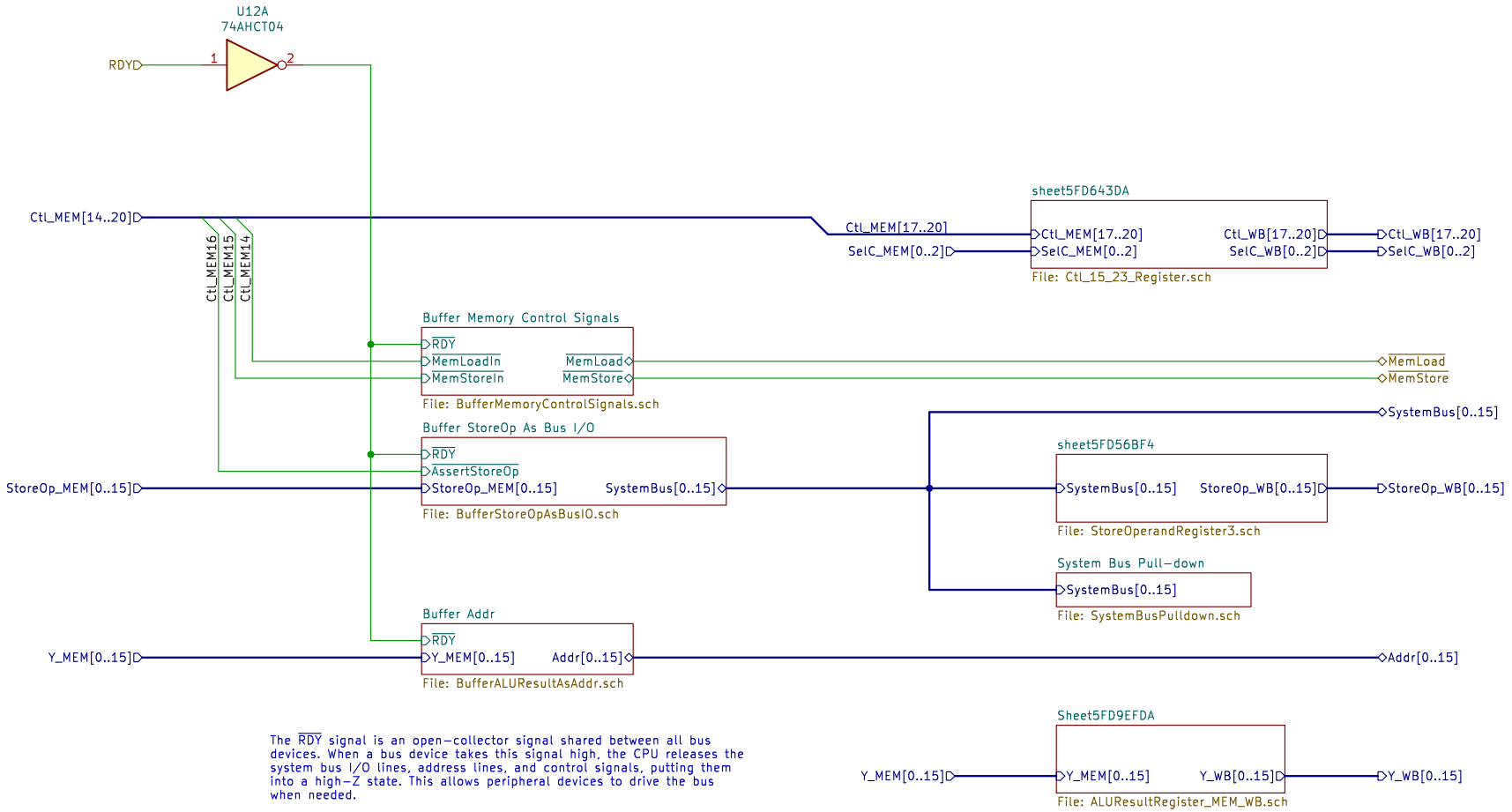


Peripheral devices on separate boards connect to the system bus here.

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

Title: System Bus Connector

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 2/39

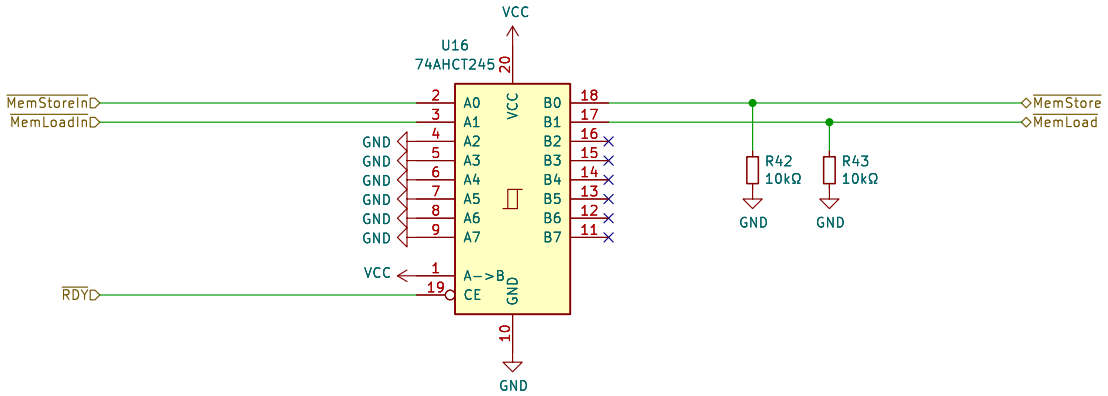


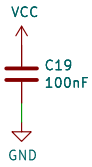
The MEM stage interfaces with memory and memory-mapped peripherals.
These devices connect to the main board via a connector described in another sheet.
Devices on the bus may take the open-collector ~RDY~ signal high to force the CPU to place bus lines into tristate and halt the Phi1 clock.

Sheet: /MEM/
File: MEM.sch

Title: MEM

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 3/39

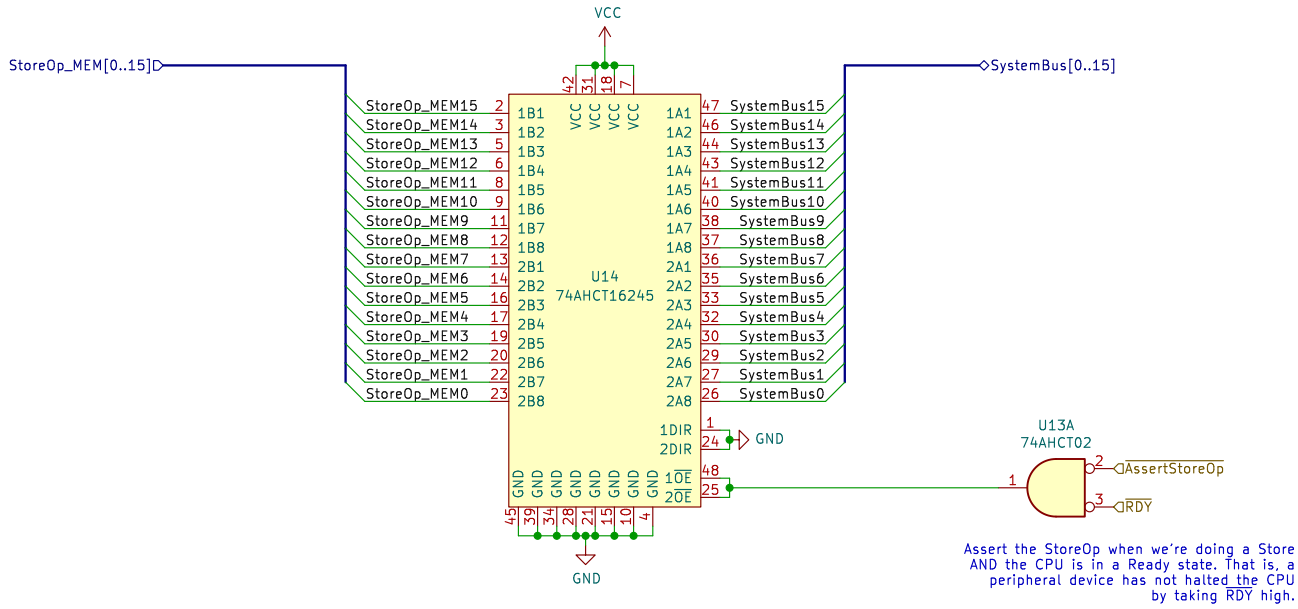


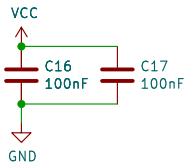


Sheet: /MEM/Buffer Memory Control Signals/
File: BufferMemoryControlSignals.sch

Title:

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 4/39



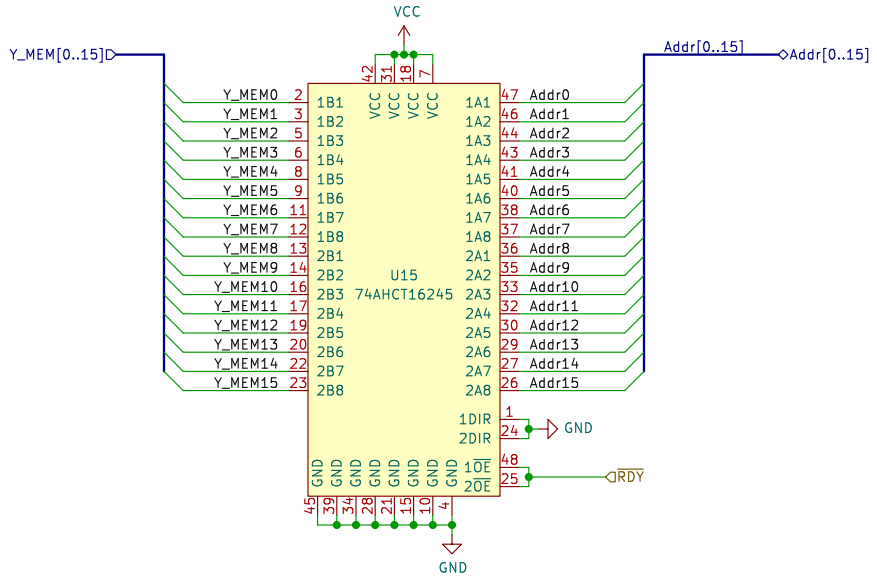


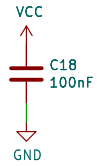
The store operand can be disconnected from the system memory bus via tristate outputs.
This allows peripheral devices to assert their own value to the bus during a Load.

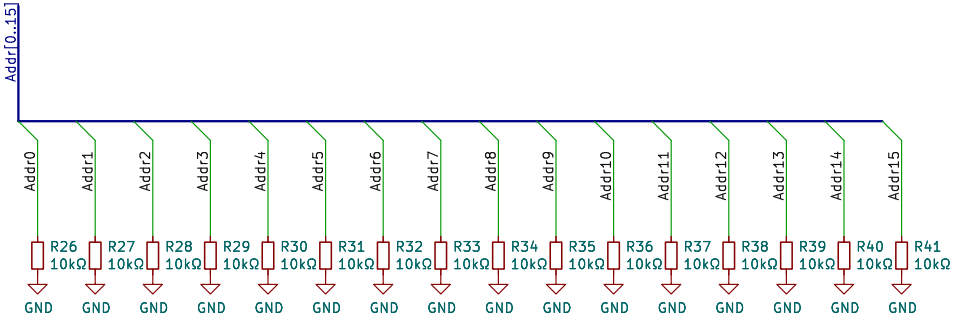
Sheet: /MEM/Buffer StoreOp As Bus I/O/
File: BufferStoreOpAsBusIO.sch

Title: MEM: Buffer StoreOp

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 5/39





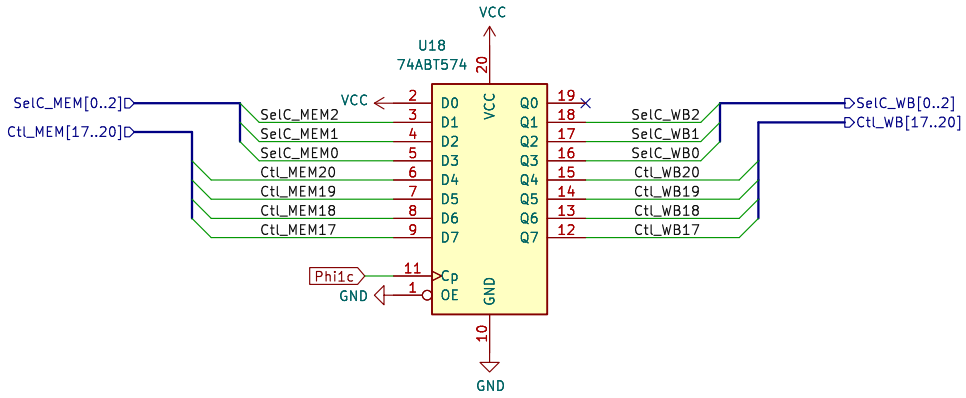


Buffer the effective address before it leaves the main board.

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

Title: MEM: Buffer ALUResult

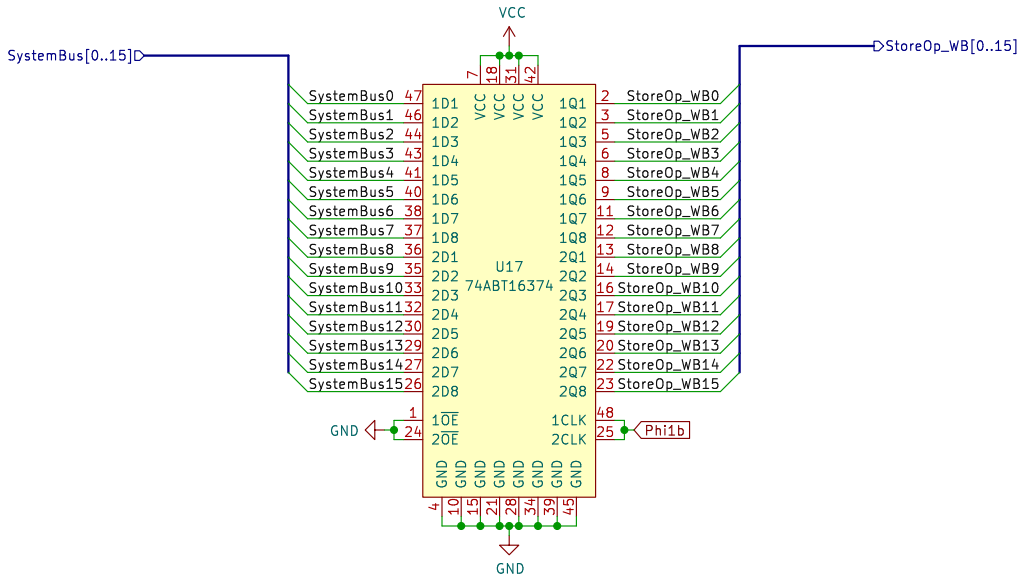
Size: USLetter	Date: 2021-04-28	Rev: A (c8cebf3f)
KiCad E.D.A. kicad 7.0.8		Id: 6/39

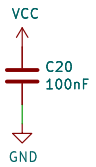


Sheet: /MEM/sheet5FD643DA/
File: Ctl_15_23_Register.sch

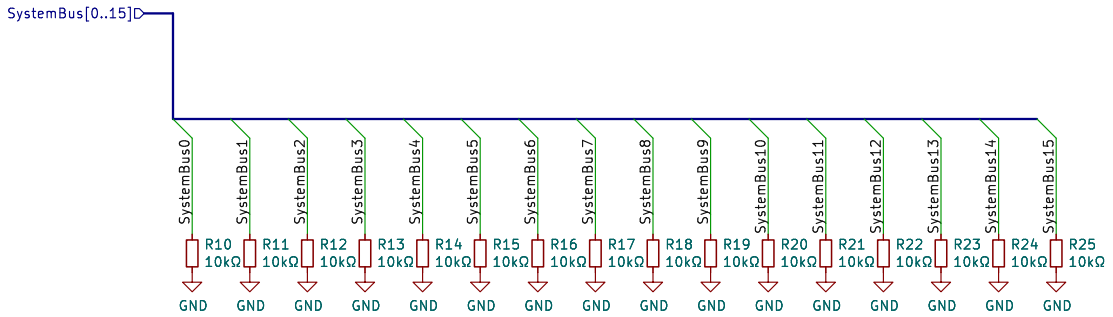
Title: MEM/WB: Control Word Register

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 7/39





Sheet: /MEM/sheet5FD56BF4/	
File: StoreOperandRegister3.sch	
Title: MEM/WB: Store Operand Register	
Size: USLetter | Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8	Id: 8/39

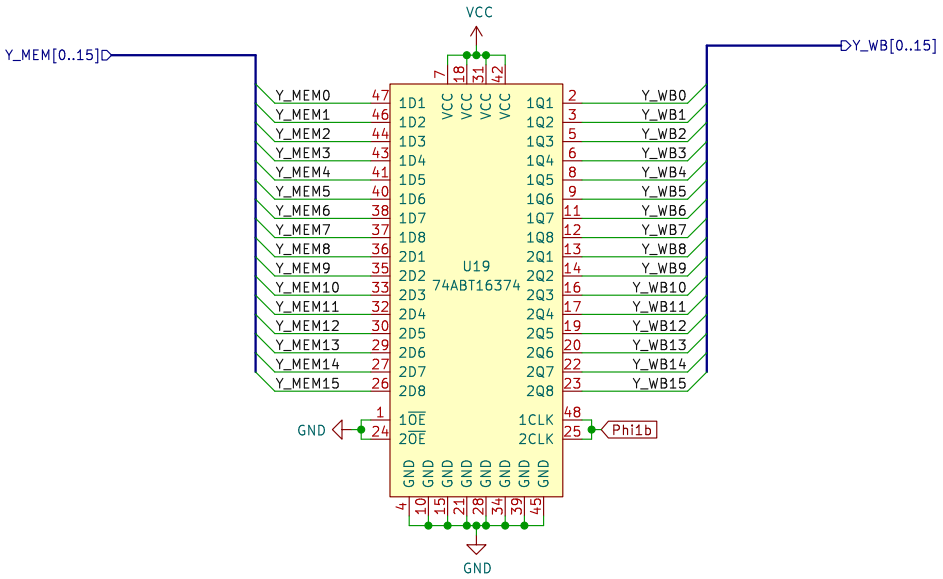


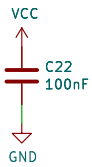
The bus needs pull-down resistors since these lines may otherwise float.?sometimes.

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

Title: MEM: System Bus Pull-down Resistors

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 9/39



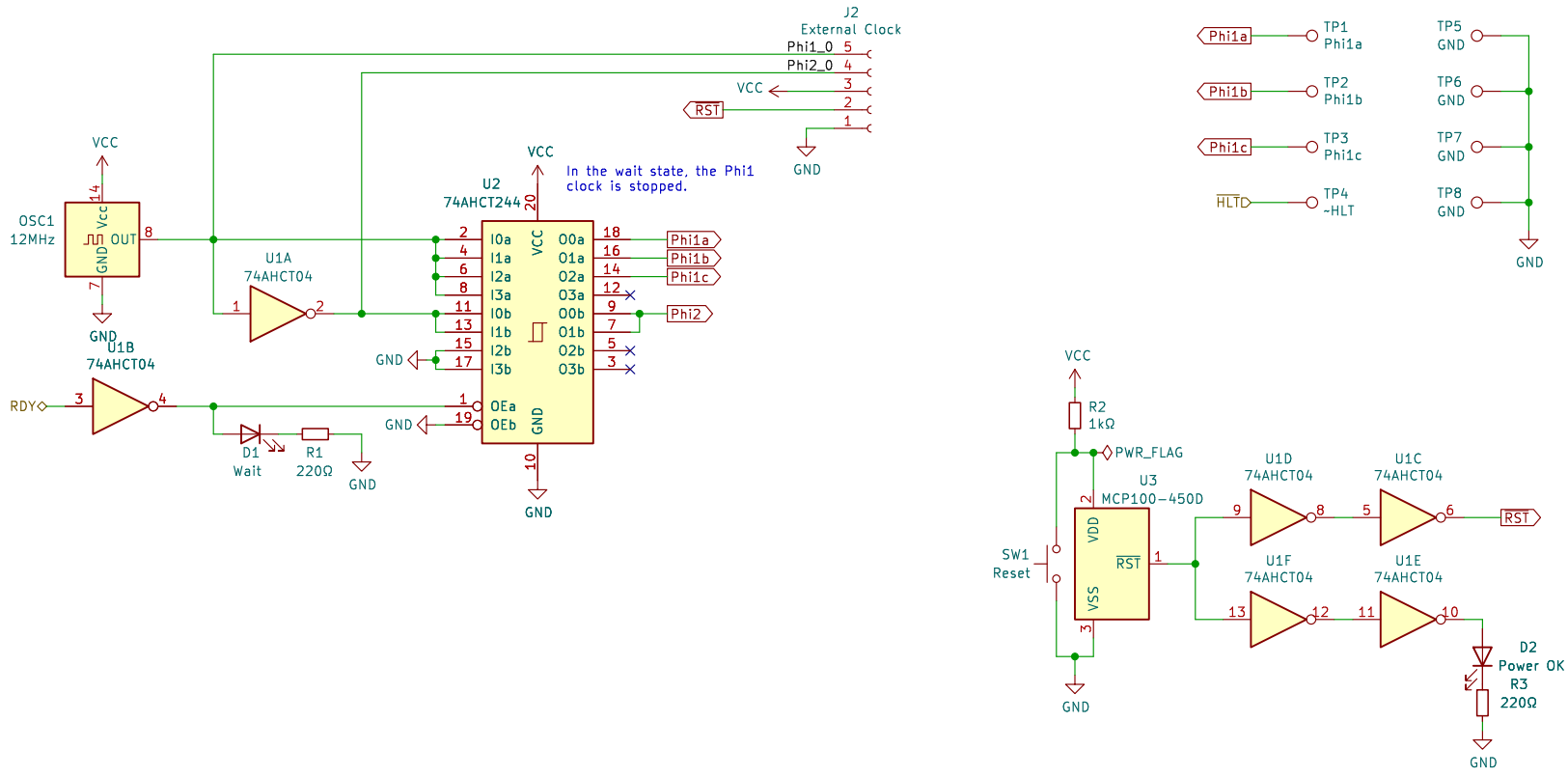


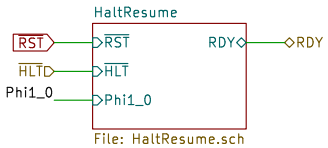
Sheet: /MEM/Sheet5FD9EFDA/
File: ALUResultRegister_MEM_WB.sch

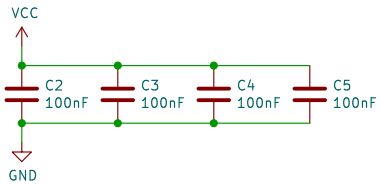
Title: ALU Result Register MEM/WB

Size: USLetter | Date: 2021-04-28 | **Rev: A (c8cebf3f)**

KiCad E.D.A. kicad 7.0.8 | Id: 10/39





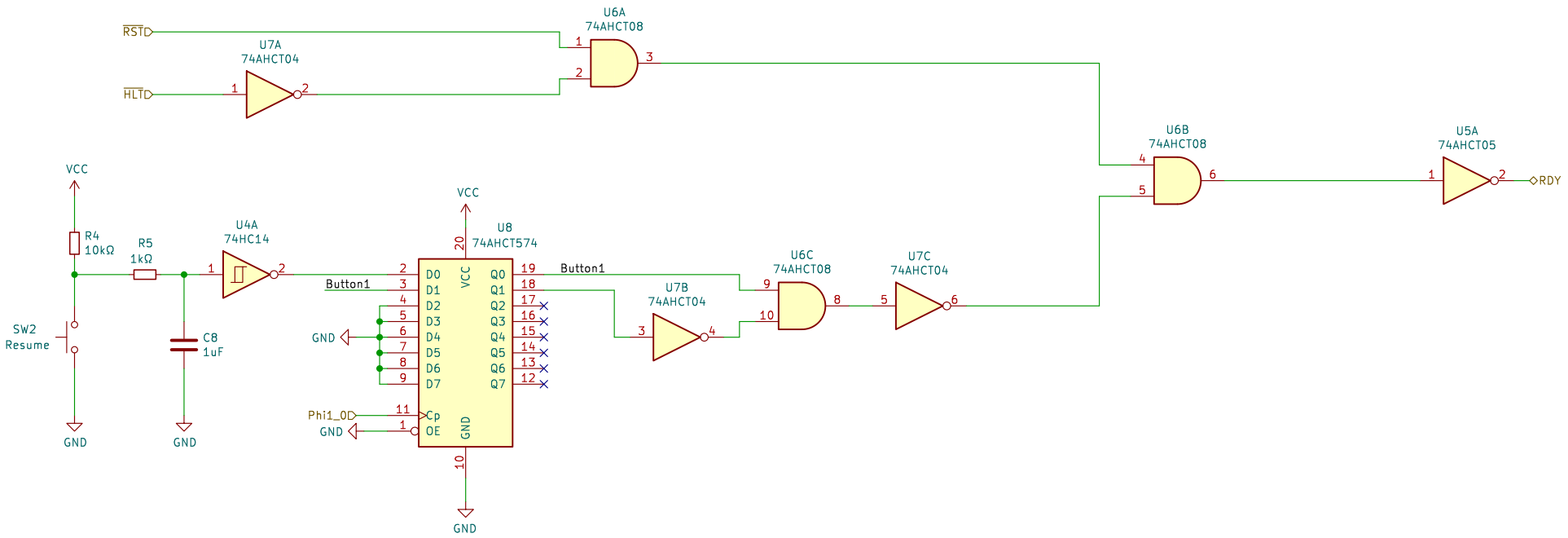


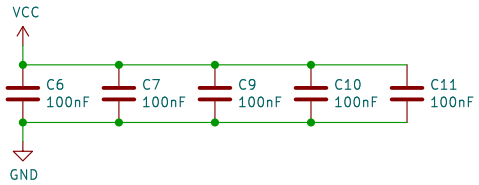
The Phi1 and Phi2 clocks have 180 degrees phase shift.
The Phi1 clock stops when the RDY shared line is pulled low.
Includes the voltage supervisor which controls the reset cycle.
Includes logic for the HLT state and a Resume button.

Sheet: /Clock/
File: Clock.sch

Title: Clock

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 11/39



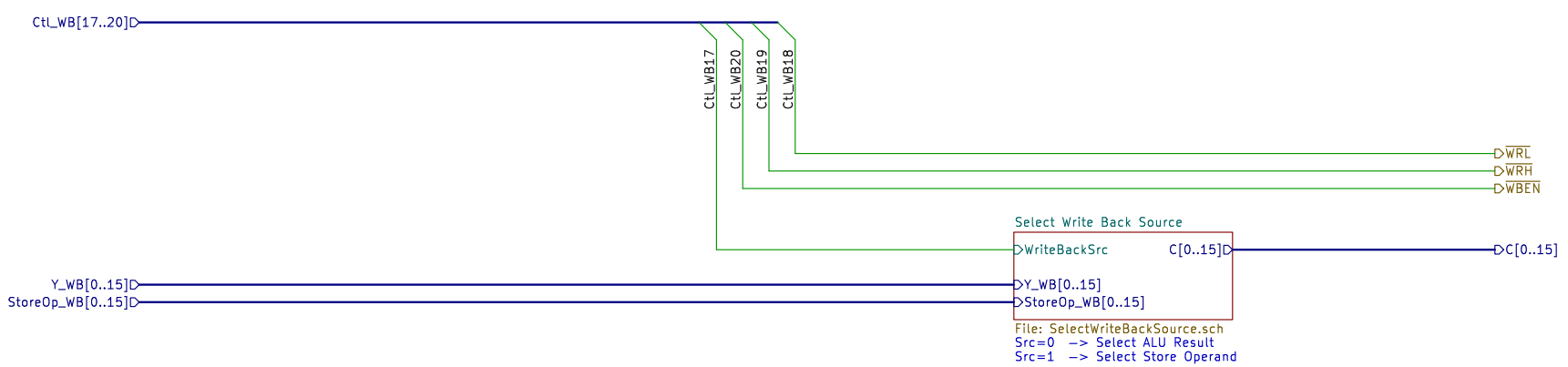


Pulls the RDY shared line low when the HLT control signal is active.
Resume execution when the button is pressed, and also on reset.

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

Title:

Size: A4	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 12/39

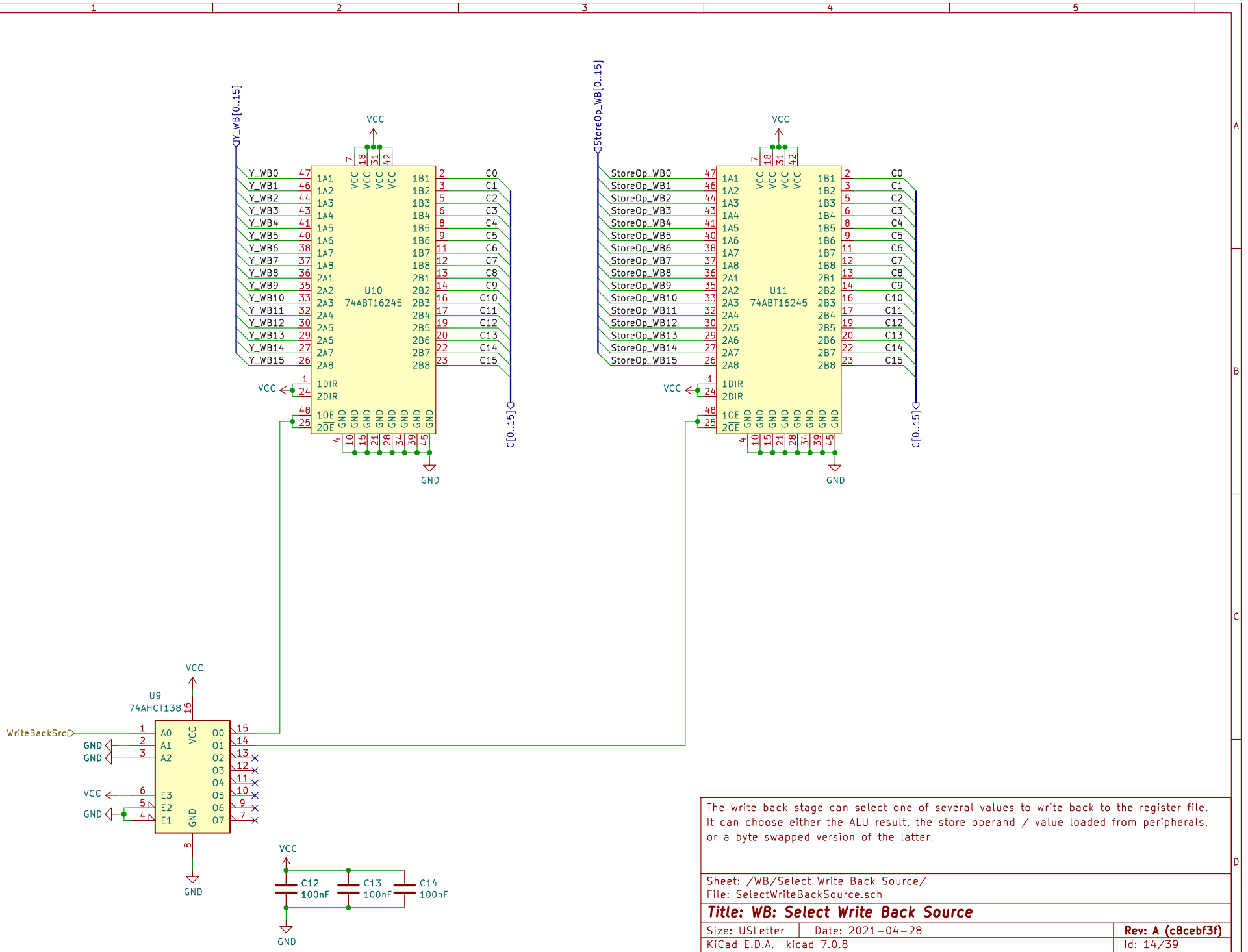


The Write Back stage chooses a value and writes it back to the register file.
This can load the upper eight bits of the register, the lower eight bits, or both,
allowing instructions to set half of a sixteen-bit word without bitwise masks.

Sheet: /WB/
File: WB.sch

Title: WB

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 13/39

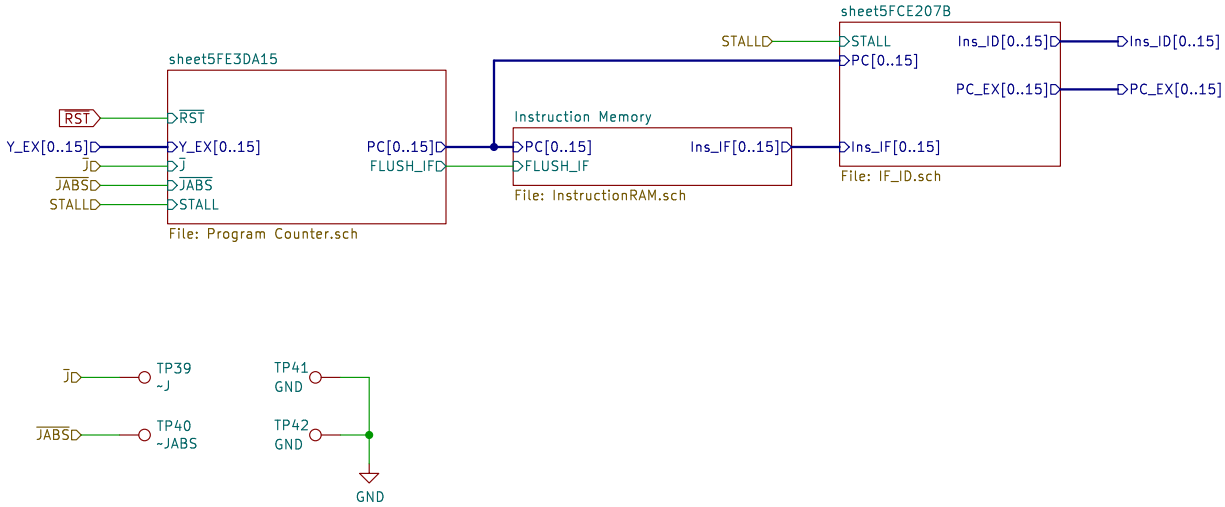


The write back stage can select one of several values to write back to the register file. It can choose either the ALU result, the store operand / value loaded from peripherals, or a byte swapped version of the latter.

Sheet: /WB/Select Write Back Source/
File: SelectWriteBackSource.sch

Title: WB: Select Write Back Source

Size: USLetter	Date: 2021-04-28	Rev: A (c8cebf3f)
KiCad E.D.A. kicad 7.0.8		Id: 14/39



The Instruction Fetch stage retrieves sixteen-bit instructions from a dual port RAM serving as Instruction Memory. The second port of the RAM is mapped into the data address space to allow the program to be modified.
All jumps are PC-relative jumps which add an offset to the program counter.

Sheet: /IF/ File: IF.sch		
Title: IF		
Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 15/39

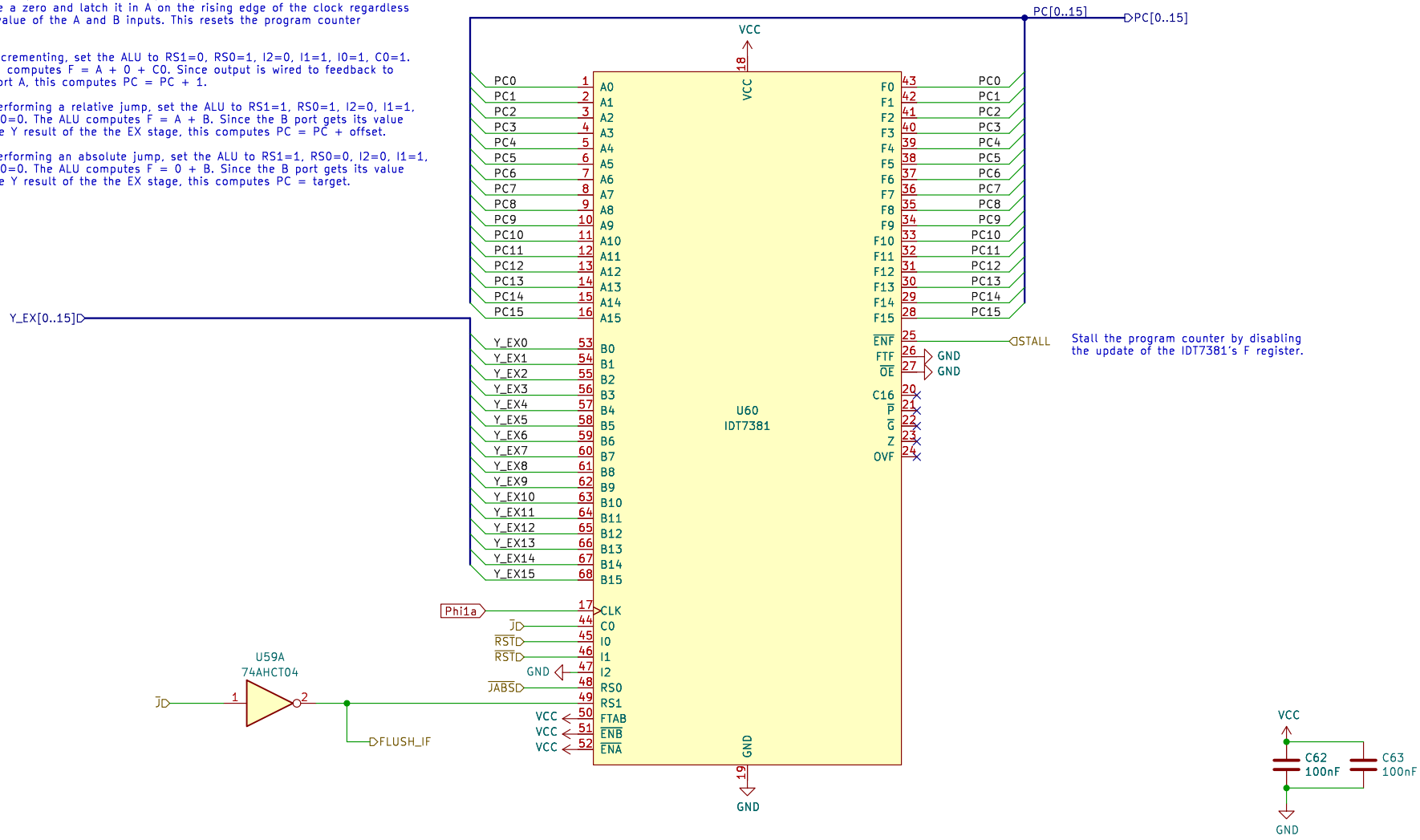
Configure the ALU for FTAB=1 and FTF=0. This causes the A and B registers to by bypassed entirely. The F output updates on the next rising edge of the clock.

During reset, set the ALU to I2=0, I1=0, I0=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, I0=1, C0=1. The ALU computes F = A + 0 + C0. 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, I0=1, C0=0. The ALU computes F = A + B. Since the B port gets its value from the Y result of the 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, I0=1, C0=0. The ALU computes F = 0 + B. Since the B port gets its value from the Y result of the the EX stage, this computes PC = target.

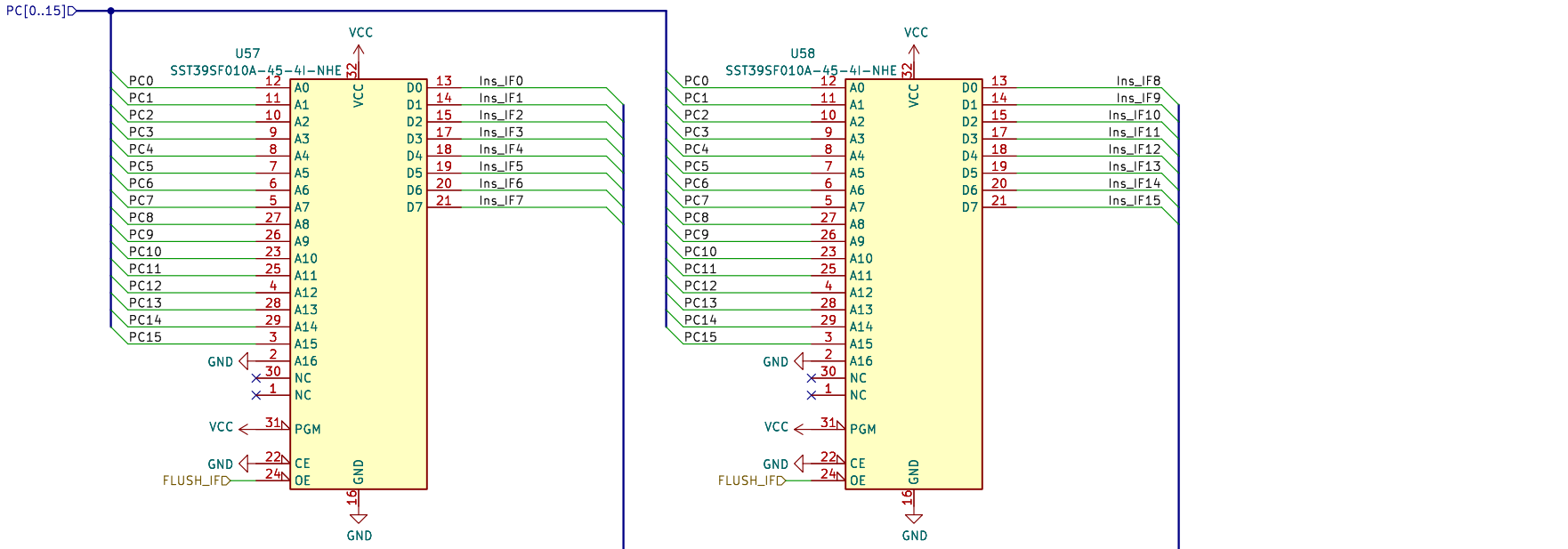


Sixteen-bit program counter will either increment on the clock, add a specified sixteen-bit offset, or else reset to zero.

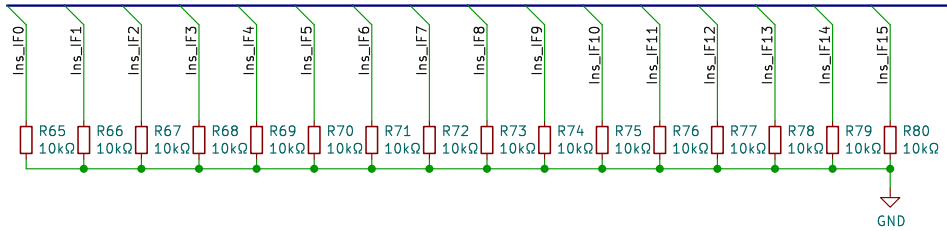
Sheet: /IF/sheet5FE3DA15/
File: Program Counter.sch

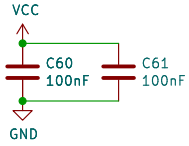
Title: Program Counter

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 16/39



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



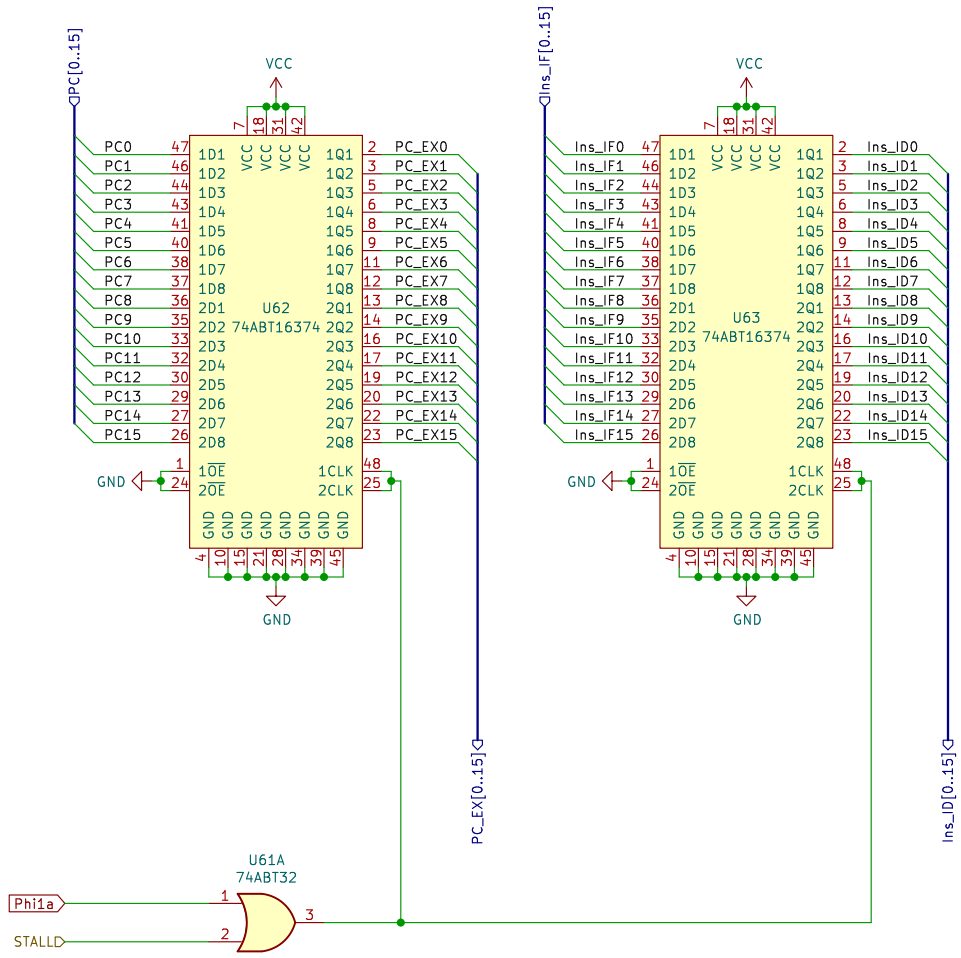


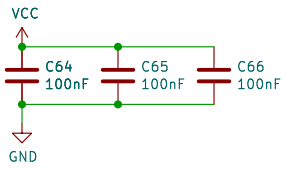
Instructions are stored in a pair of EEPROMs in ZIF sockets.

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

Title: Instruction ROM

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 17/39



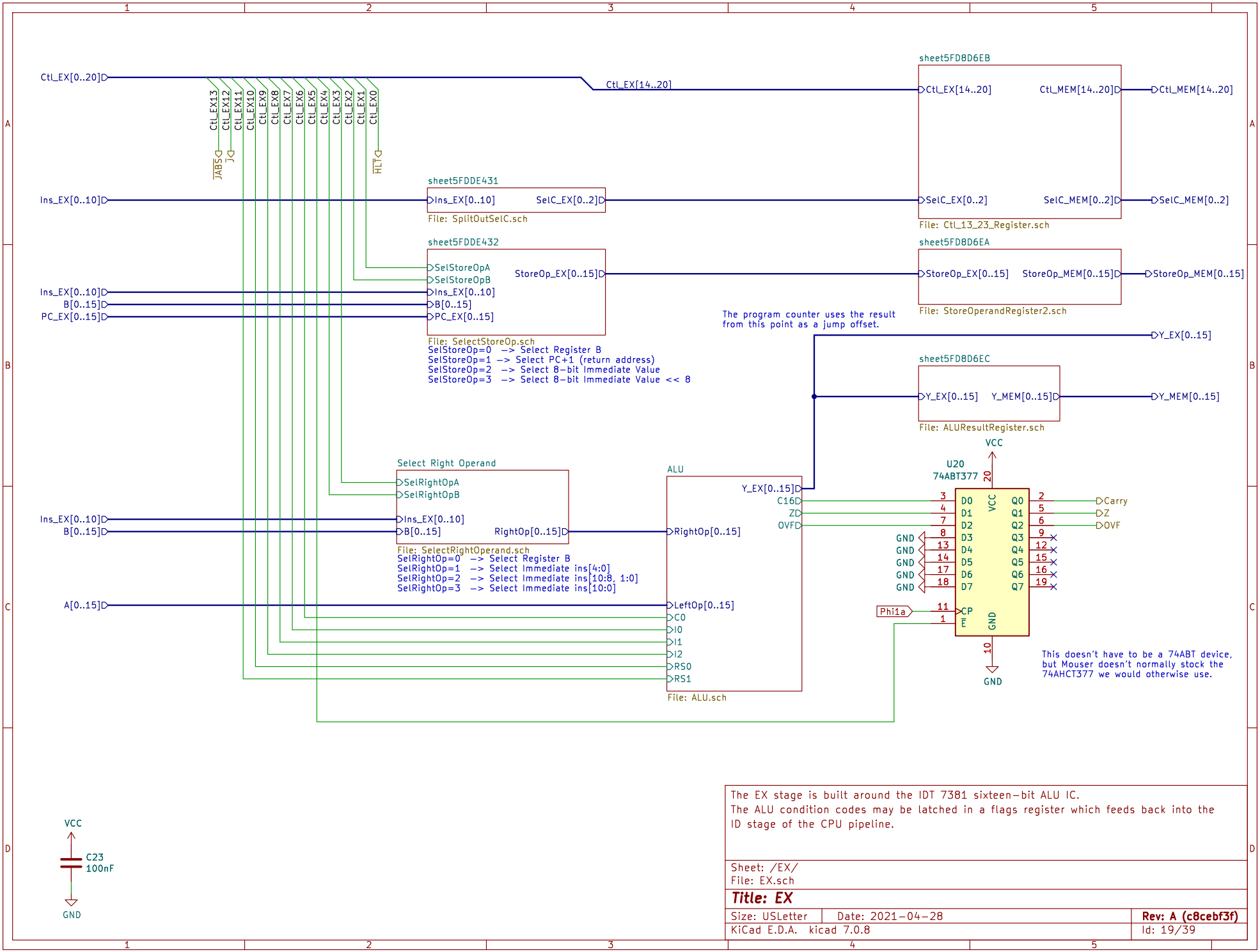


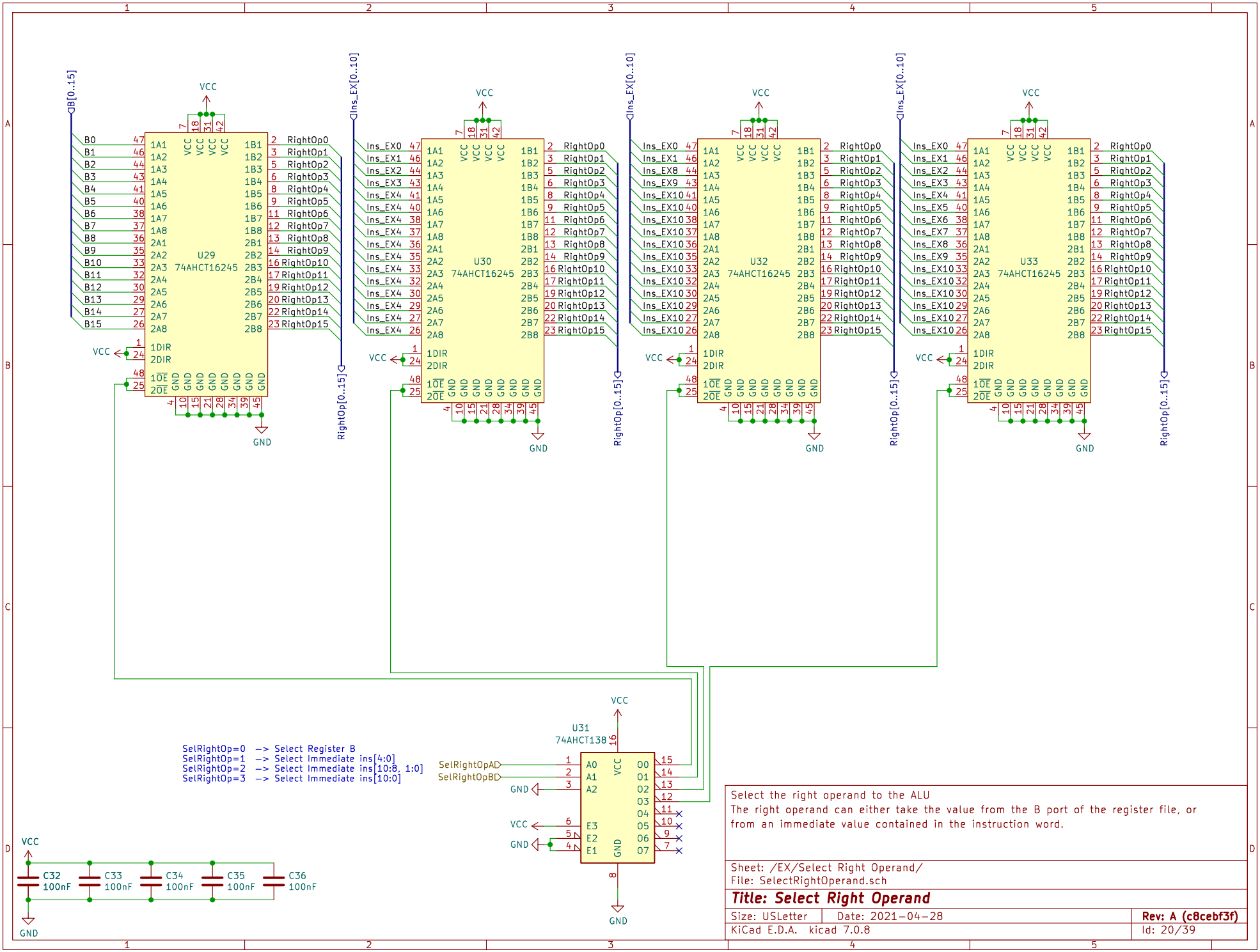
Interstage pipeline registers between IF and ID

Sheet: /IF/sheet5FCE207B/
File: IF_ID.sch

Title: IF/ID

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 18/39

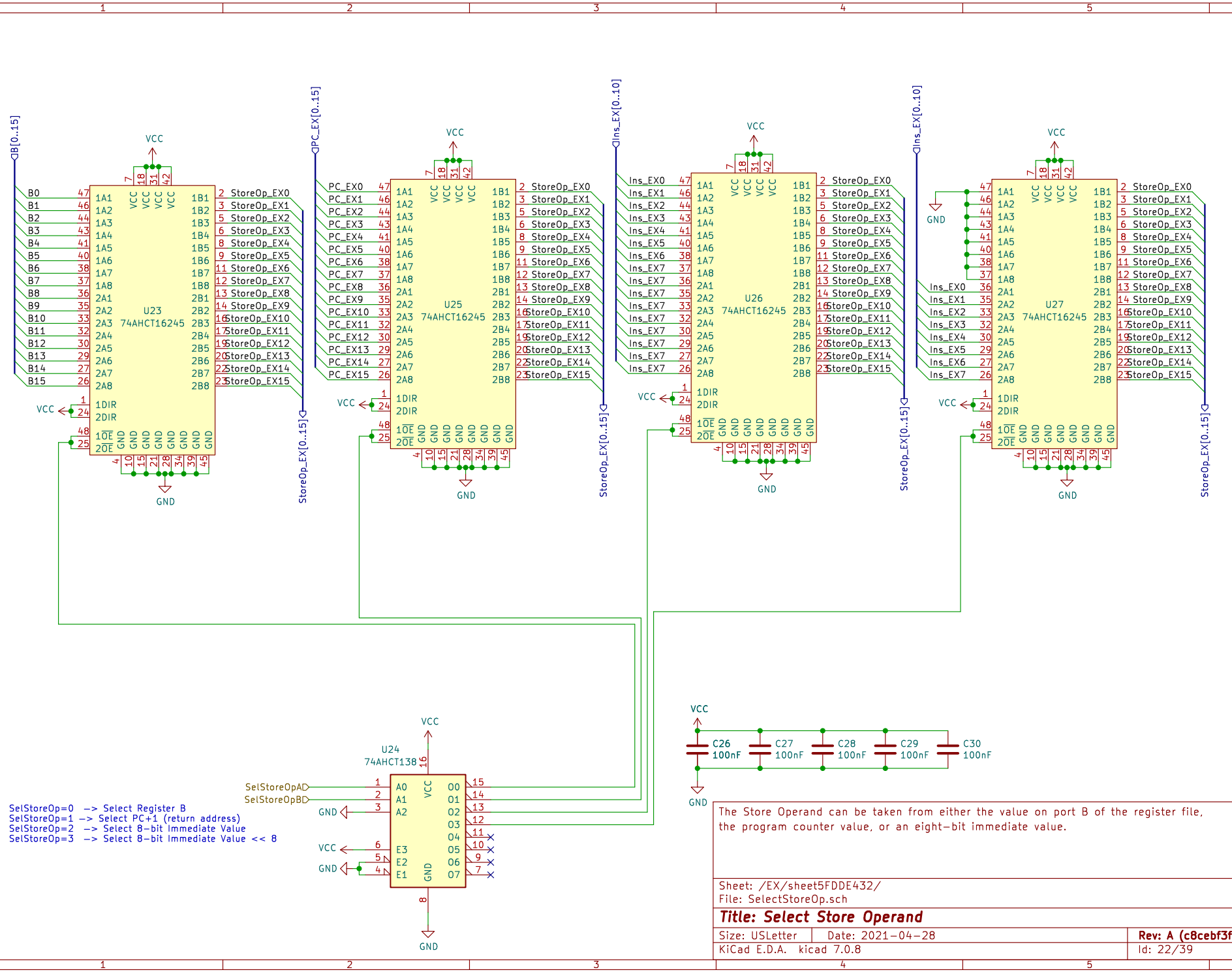

Ctl_EX[0..20]
Ctl_EX13
Ctl_EX12
Ctl_EX11
Ctl_EX10
Ctl_EX9
Ctl_EX8
Ctl_EX7
Ctl_EX6
Ctl_EX5
Ctl_EX4
Ctl_EX3
Ctl_EX2
Ctl_EX1
Ctl_EX0
JABS
J
HLT
Ctl_EX[14..20]
sheet5FD8D6EB
Ctl_EX[14..20]
Ctl_MEM[14..20]
Ctl_MEM[14..20]
File: Ctl_13_23_Register.sch
Ins_EX[0..10]
sheet5FDDE431
Ins_EX[0..10]
SelC_EX[0..2]
File: SplitOutSelC.sch
SelC_EX[0..2]
SelC_MEM[0..2]
SelC_MEM[0..2]
sheet5FDDE432
SelStoreOpA
SelStoreOpB
Ins_EX[0..10]
B[0..15]
PC_EX[0..15]
StoreOp_EX[0..15]
File: SelectStoreOp.sch
SelStoreOp=0 -> Select Register B
SelStoreOp=1 -> Select PC+1 (return address)
SelStoreOp=2 -> Select 8-bit Immediate Value
SelStoreOp=3 -> Select 8-bit Immediate Value << 8
sheet5FD8D6EA
StoreOp_EX[0..15]
StoreOp_MEM[0..15]
StoreOp_MEM[0..15]
File: StoreOperandRegister2.sch
The program counter uses the result
from this point as a jump offset.
Y_EX[0..15]
sheet5FD8D6EC
Y_EX[0..15]
Y_MEM[0..15]
Y_MEM[0..15]
File: ALUResultRegister.sch
Select Right Operand
SelRightOpA
SelRightOpB
Ins_EX[0..10]
B[0..15]
RightOp[0..15]
File: SelectRightOperand.sch
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]
ALU
Y_EX[0..15]
C16
ZD
OVFD
RightOp[0..15]
A[0..15]
LeftOp[0..15]
C0
I0
I1
I2
RS0
RS1
File: ALU.sch
VCC
U20
74ABT377
20
D0
D1
D2
D3
D4
D5
D6
D7
VCC
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
GND
GND
GND
GND
GND
GND
Phi1a
CP
E
GND
10
GND
Carry
Z
OVF
This doesn't have to be a 74ABT device,
but Mouser doesn't normally stock the
74AHCT377 we would otherwise use.
VCC
C23
100nF
GND
The EX stage is built around the IDT 7381 sixteen-bit ALU IC.
The ALU condition codes may be latched in a flags register which feeds back into the
ID stage of the CPU pipeline.
Sheet: /EX/
File: EX.sch
Title: EX
Size: USLetter
Date: 2021-04-28
Rev: A (c8cebf3f)
KiCad E.D.A. kicad 7.0.8
Id: 19/39

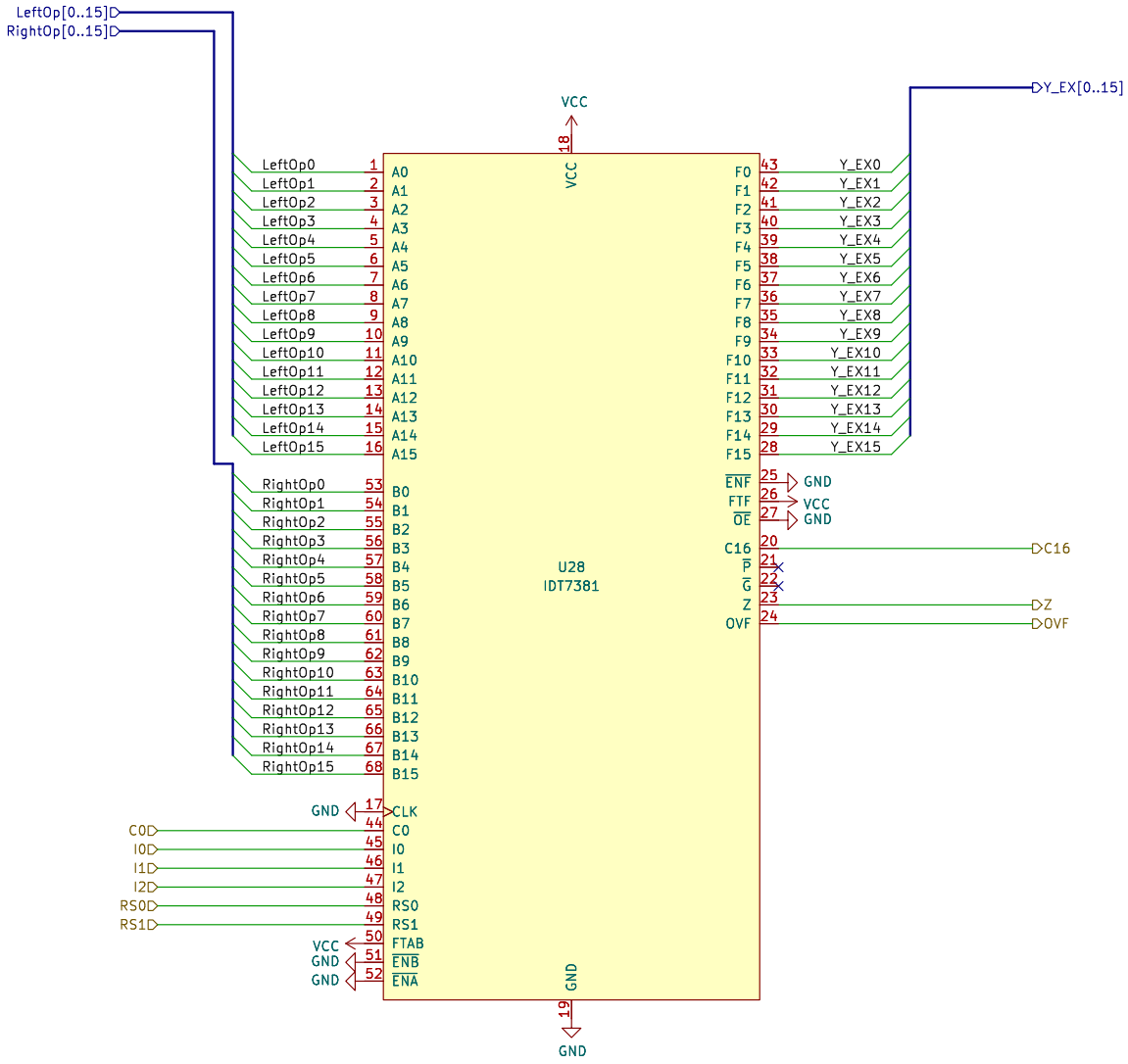

B[0..15]
Ins_EX[0..10]
RightOp[0..15]
U29
U30
U32
U33
74AHCT16245
VCC
GND
SelRightOp=0 -> Select Register B
SelRightOp=1 -> Select Immediate ins[4:0]
SelRightOp=2 -> Select Immediate ins[10:8, 1:0]
SelRightOp=3 -> Select Immediate ins[10:0]
U31
74AHCT138
SelRightOpA
SelRightOpB
C32 100nF
C33 100nF
C34 100nF
C35 100nF
C36 100nF
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: 2021-04-28
KiCad E.D.A. kicad 7.0.8
Rev: A (c8cebf3f)
Id: 20/39

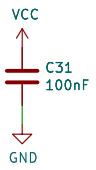
Sheet: /EX/sheet5FDDE431/
File: SplitOutSelC.sch

Title:

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 21/39


B[0..15]
PC_EX[0..15]
Ins_EX[0..10]
Ins_EX[0..10]
StoreOp_EX[0..15]
StoreOp_EX[0..15]
StoreOp_EX[0..15]
StoreOp_EX[0..15]
U23
74AHCT16245
U25
74AHCT16245
U26
74AHCT16245
U27
74AHCT16245
U24
74AHCT138
VCC
GND
SelStoreOpA
SelStoreOpB
C26 100nF
C27 100nF
C28 100nF
C29 100nF
C30 100nF
SelStoreOp=0 -> Select Register B
SelStoreOp=1 -> Select PC+1 (return address)
SelStoreOp=2 -> Select 8-bit Immediate Value
SelStoreOp=3 -> Select 8-bit Immediate Value << 8
The Store Operand can be taken from either the value on port B of the register file, the program counter value, or an eight-bit immediate value.
Sheet: /EX/sheet5FDDE432/
File: SelectStoreOp.sch
Title: Select Store Operand
Size: USLetter
Date: 2021-04-28
Rev: A (c8cebf3f)
KiCad E.D.A. kicad 7.0.8
Id: 22/39



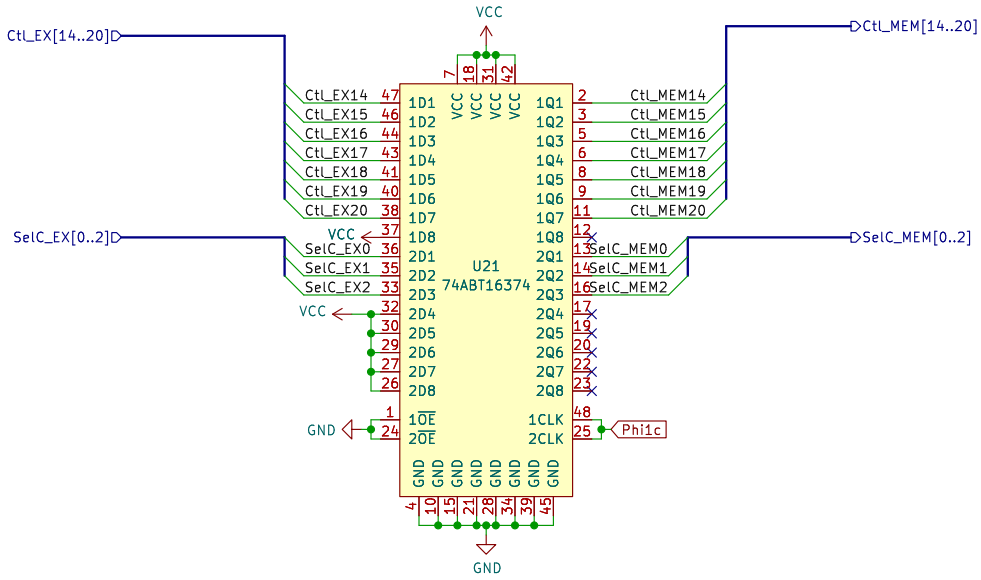


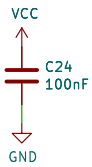
The ALU is built around the IDT 7831 sixteen-bit monolithic ALU IC.
This is configured for pure combinatorial operation where operands and outputs are latched in registers in other parts of the circuit.

Sheet: /EX/ALU/
File: ALU.sch

Title: ALU

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 23/39

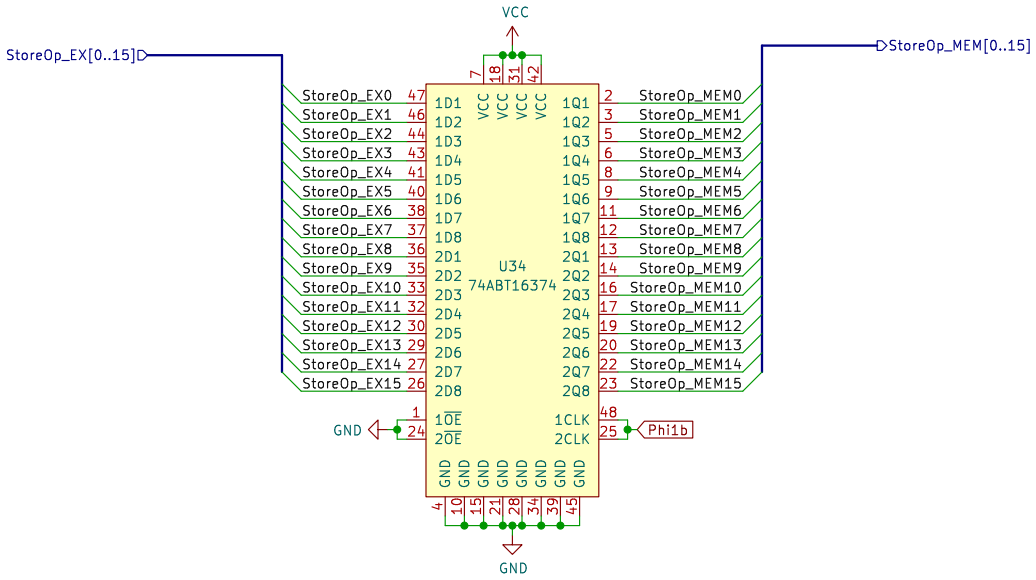


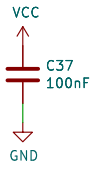


Sheet: /EX/sheet5FD8D6EB/
File: Ctl_13_23_Register.sch

Title: EX/MEM: Control Word Register

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 24/39

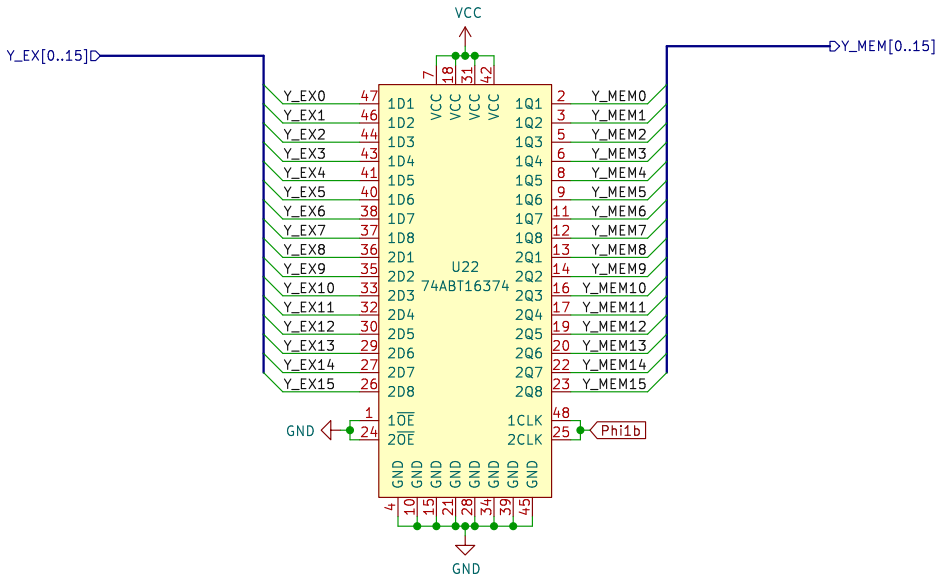


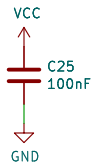


Sheet: /EX/sheet5FD8D6EA/
File: StoreOperandRegister2.sch

Title: EX/MEM: Store Operand Register

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 25/39

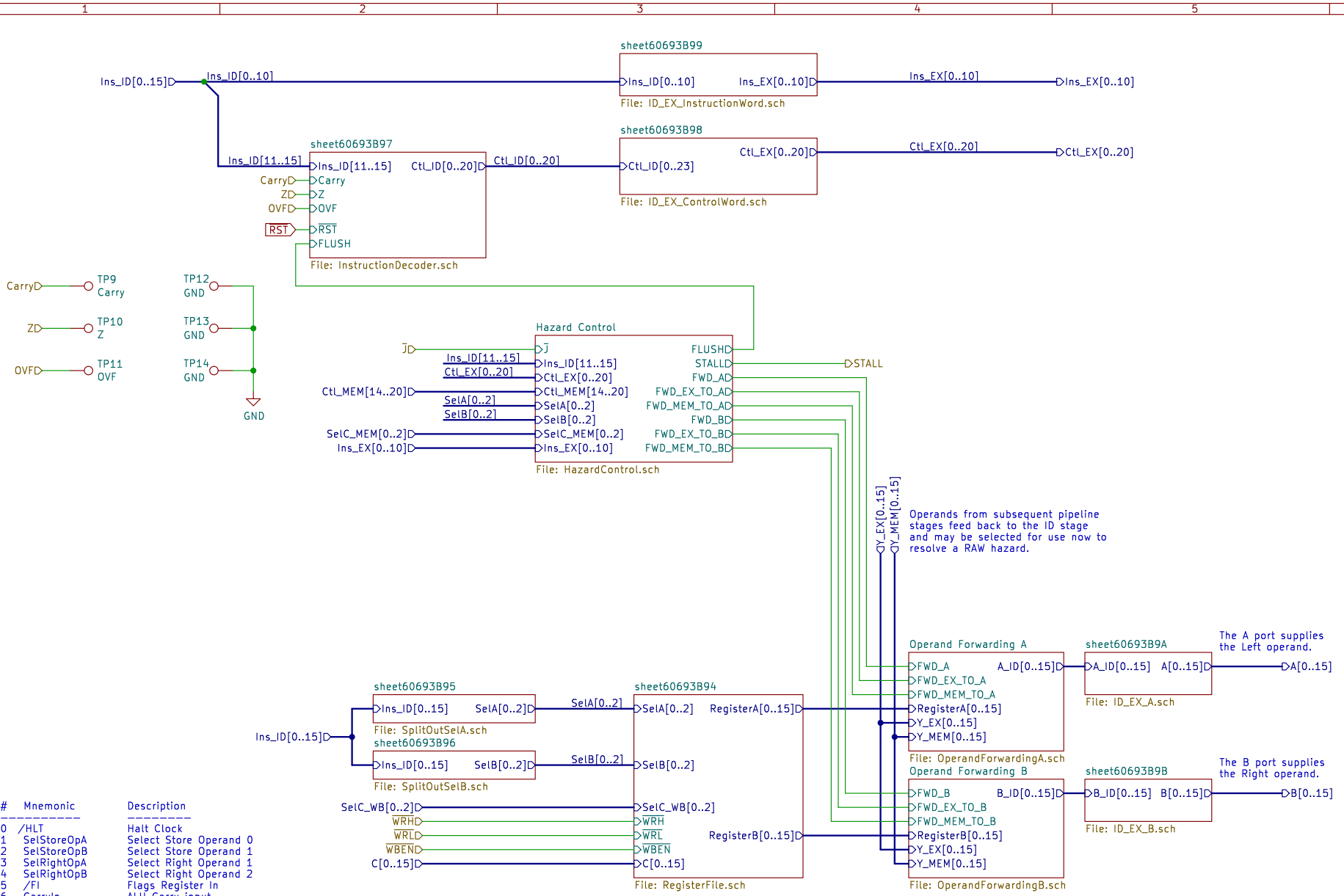




Sheet: /EX/sheet5FD8D6EC/
File: ALUResultRegister.sch

Title: EX/MEM: ALUResult Register

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 26/39



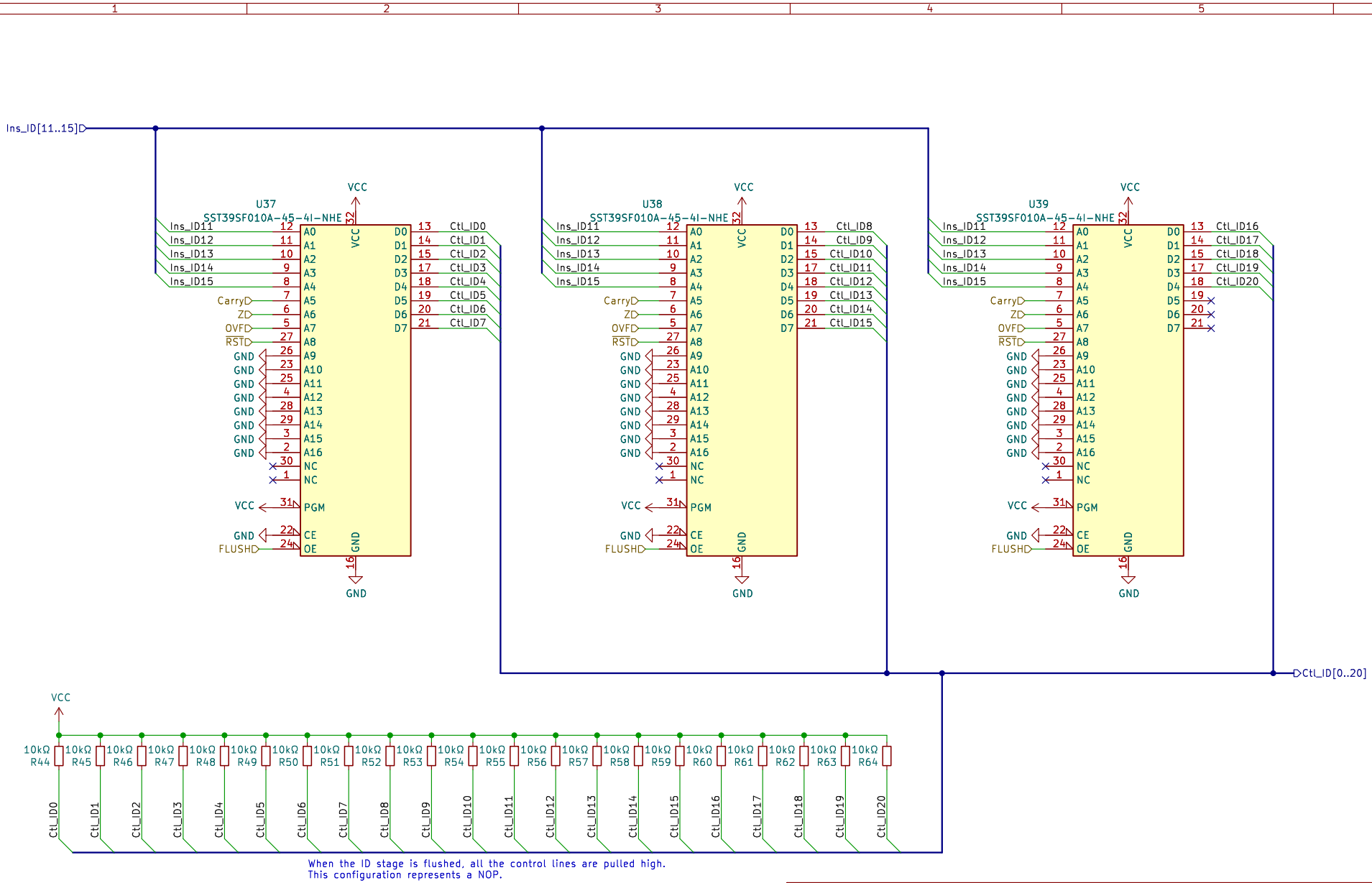
#	Mnemonic	Description
0	/HLT	Halt Clock
1	SelStoreOpA	Select Store Operand 0
2	SelStoreOpB	Select Store Operand 1
3	SelRightOpA	Select Right Operand 1
4	SelRightOpB	Select Right Operand 2
5	/FI	Flags Register In
6	CarryIn	ALU Carry input
7	I0	ALU I0 input
8	I1	ALU I1 input
9	I2	ALU I2 input
10	RS0	ALU RS0 input
11	RS1	ALU RS1 input
12	/J	Jump
13	/JABS	Absolute Jump
14	/MemLoad	Memory Load
15	/MemStore	Memory Store
16	/AssertStoreOp	Drive the Store operand onto the bus I/O lines
17	WriteBackSrc	Select write back source
18	/WRL	Write back low byte
19	/WRH	Write back high byte
20	/WBEN	Enable write back to register file

The instruction decoder turns a 5-bit opcode into an array of control signals.
The decoder takes the condition code from the flags register into account to effect conditional instructions.
Simultaneously, read the register file using indices extracted from the instruction word.

Sheet: /ID/
File: ID.sch

Title: ID

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 27/39



Decodes an instruction opcode into an array of control signals.

Sheet: /ID/sheet60693B97/
File: InstructionDecoder.sch

Title: Instruction Decoder

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 28/39

Sheet: /ID/sheet60693B95/
File: SplitOutSelA.sch

Title:

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 29/39

Sheet: /ID/sheet60693B96/		
File: SplitOutSelB.sch		
Title:		
Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 30/39

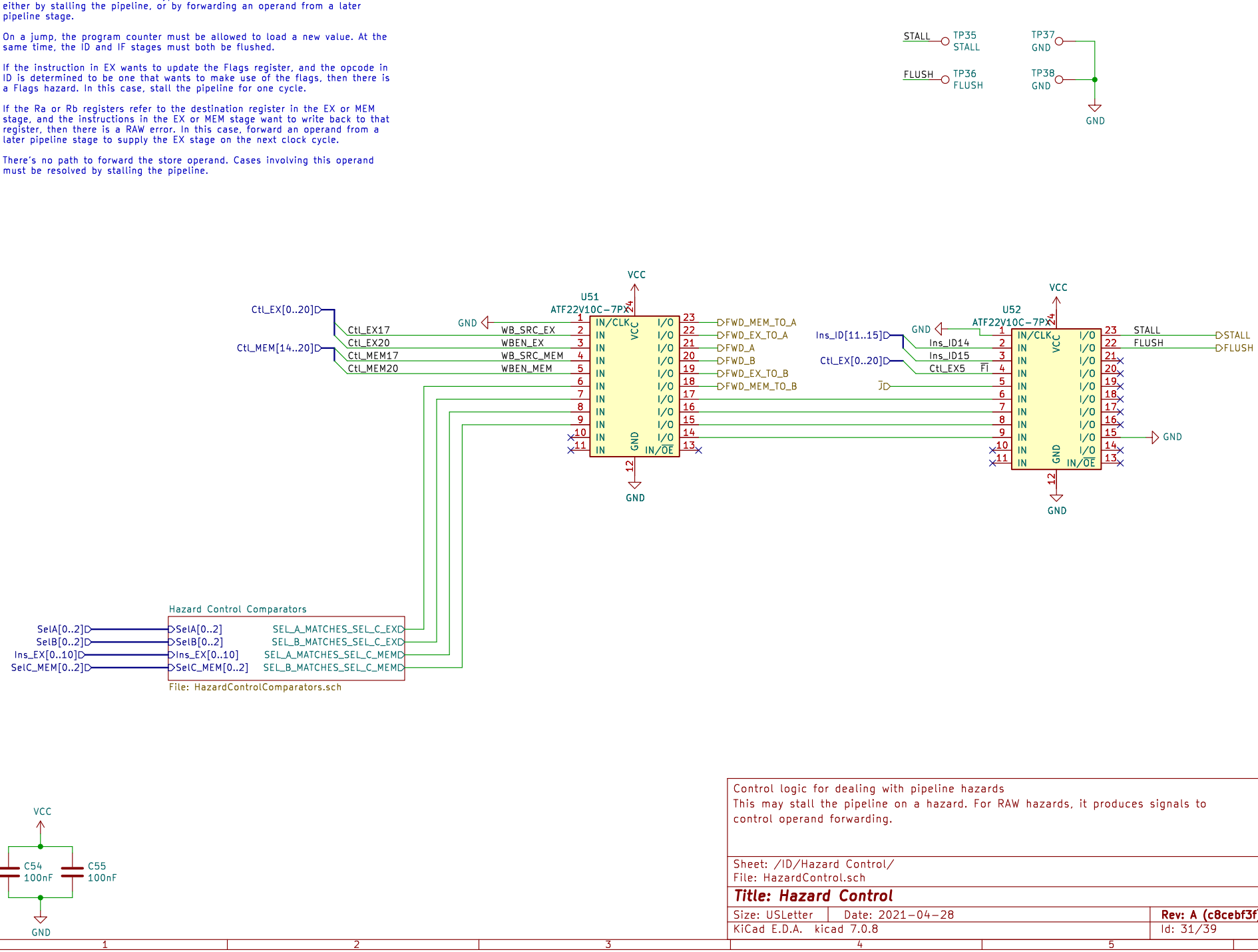
Hazard Control detects several types of hazards and resolves each one either by stalling the pipeline, or by forwarding an operand from a later pipeline stage.

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.

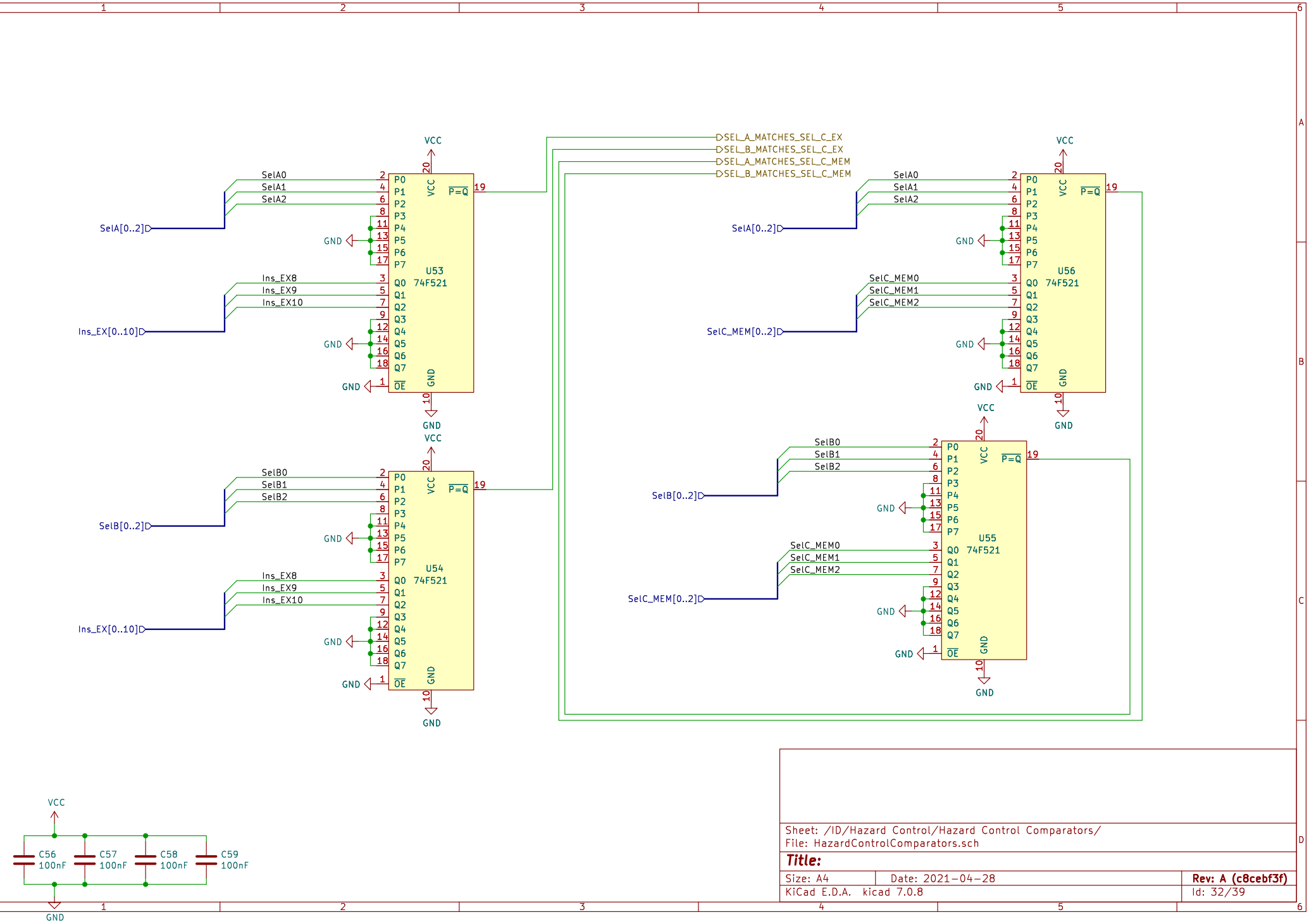


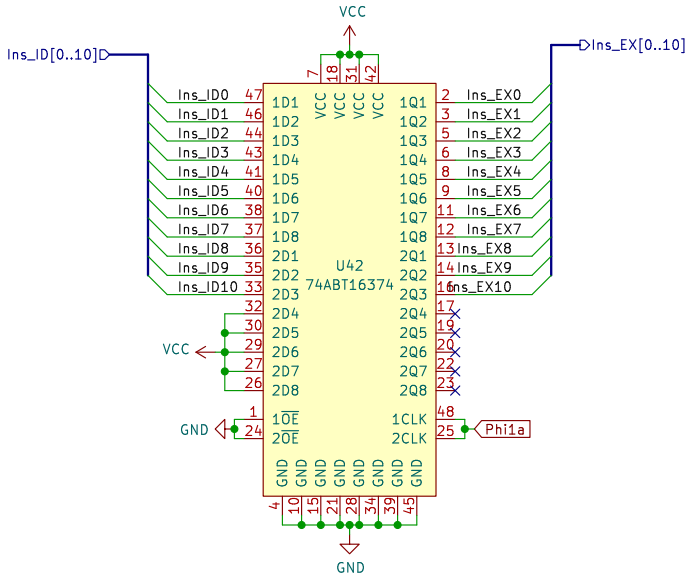
Control logic for dealing with pipeline hazards
This may stall the pipeline on a hazard. For RAW hazards, it produces signals to control operand forwarding.

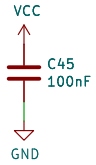
Sheet: /ID/Hazard Control/
File: HazardControl.sch

Title: Hazard Control

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 31/39


SEL_A_MATCHES_SEL_C_EX
SEL_B_MATCHES_SEL_C_EX
SEL_A_MATCHES_SEL_C_MEM
SEL_B_MATCHES_SEL_C_MEM
U53
74F521
SelA[0..2]
SelA0
SelA1
SelA2
Ins_EX[0..10]
Ins_EX8
Ins_EX9
Ins_EX10
U54
74F521
SelB[0..2]
SelB0
SelB1
SelB2
U56
74F521
SelC_MEM[0..2]
SelC_MEM0
SelC_MEM1
SelC_MEM2
U55
74F521
VCC
GND
C56
100nF
C57
100nF
C58
100nF
C59
100nF
Sheet: /ID/Hazard Control/Hazard Control Comparators/
File: HazardControlComparators.sch
Title:
Size: A4
Date: 2021-04-28
Rev: A (c8cebf3f)
KiCad E.D.A. kicad 7.0.8
Id: 32/39

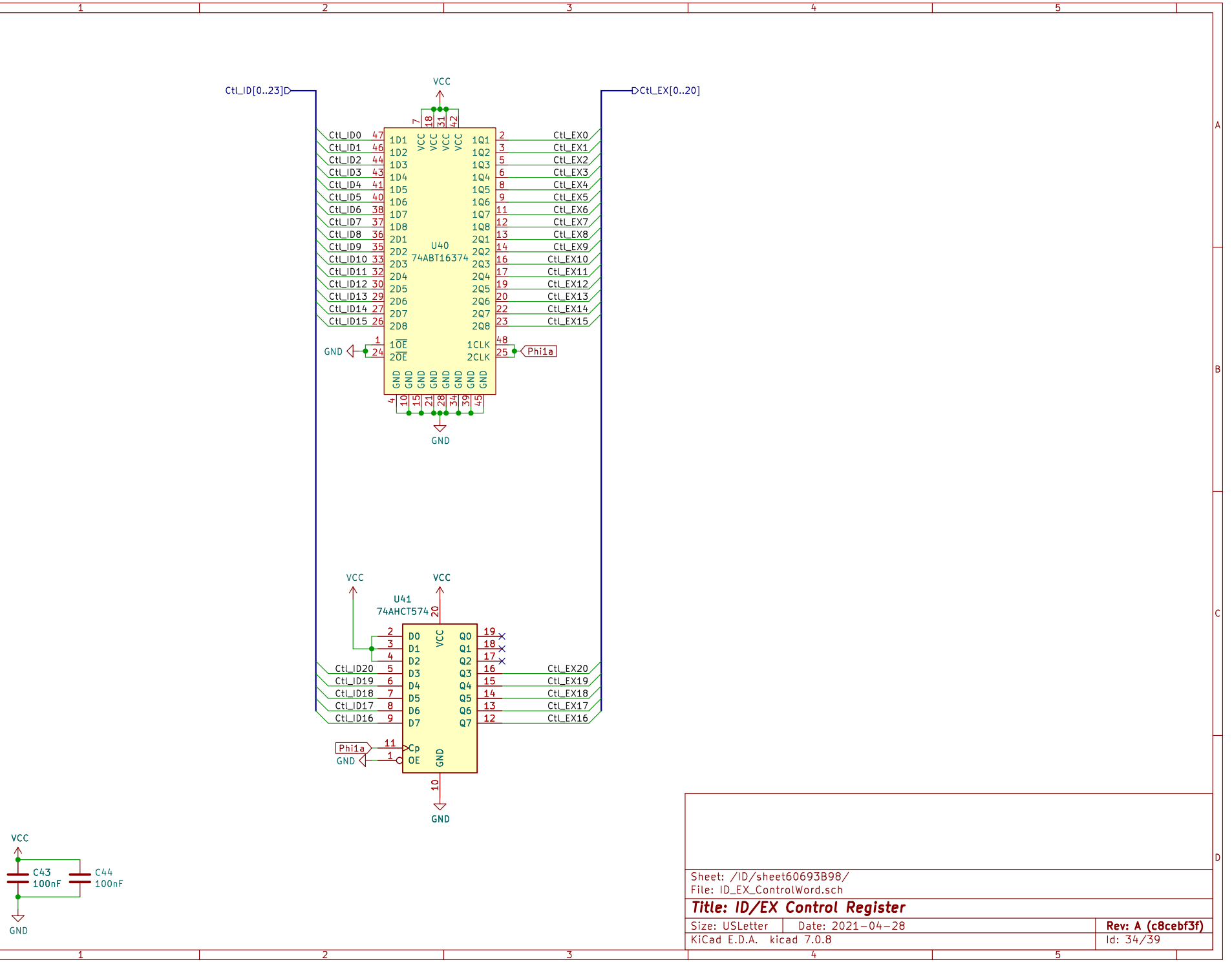




Sheet: /ID/sheet60693B99/
File: ID_EX_InstructionWord.sch

Title:

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 33/39

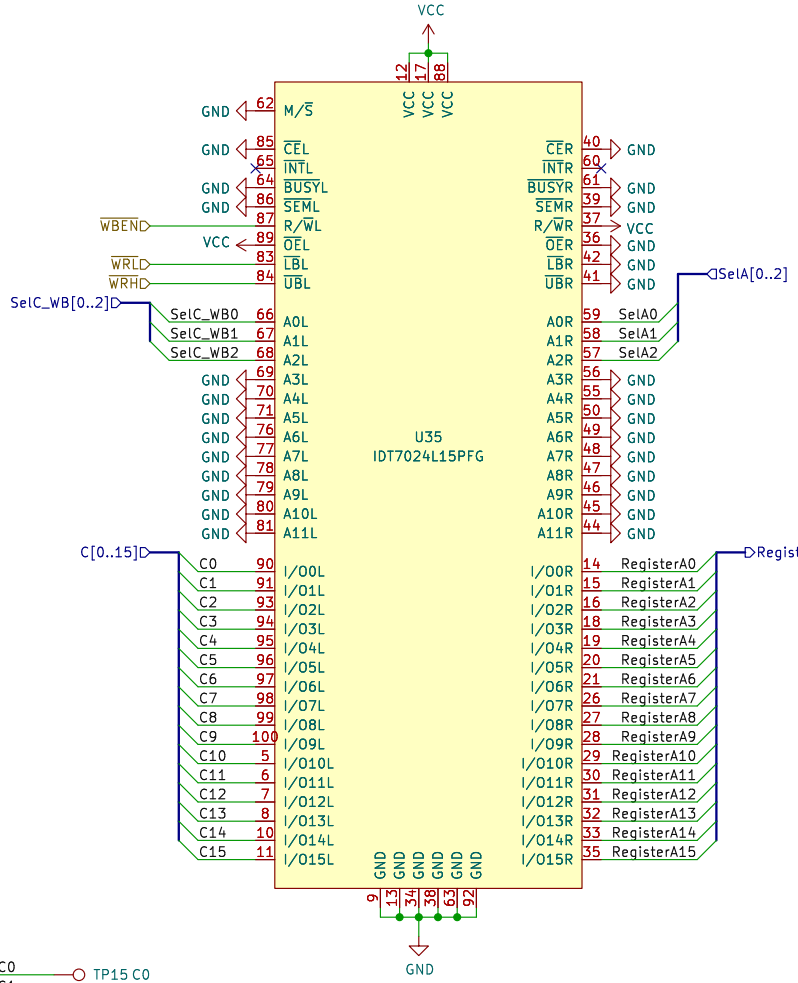


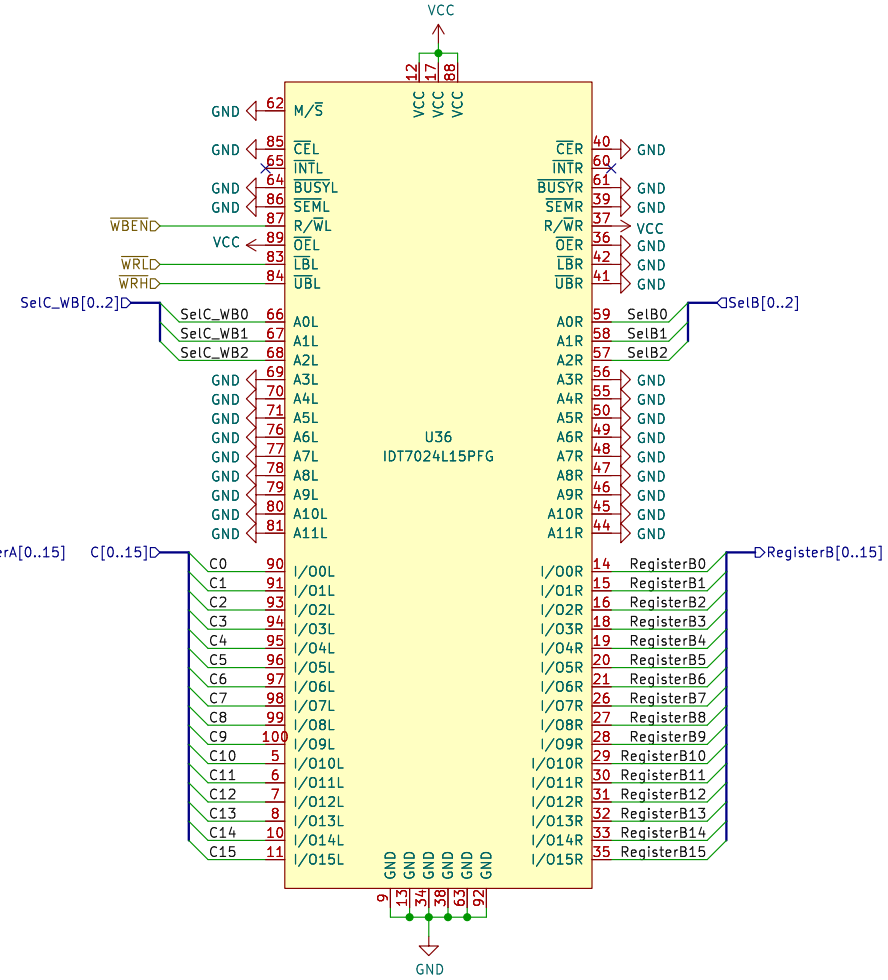
Sheet: /ID/sheet60693B98/
File: ID_EX_ControlWord.sch

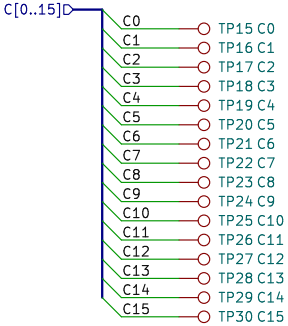
Title: ID/EX Control Register

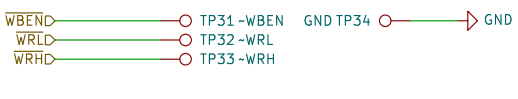
Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 34/39

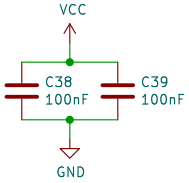
Both dual port SRAMs are configured in Slave mode. This disables the on board contention arbitration logic. Per application note, AN-91, a simultaneous read and write to the same cell will flow through from one port to another after a short delay.









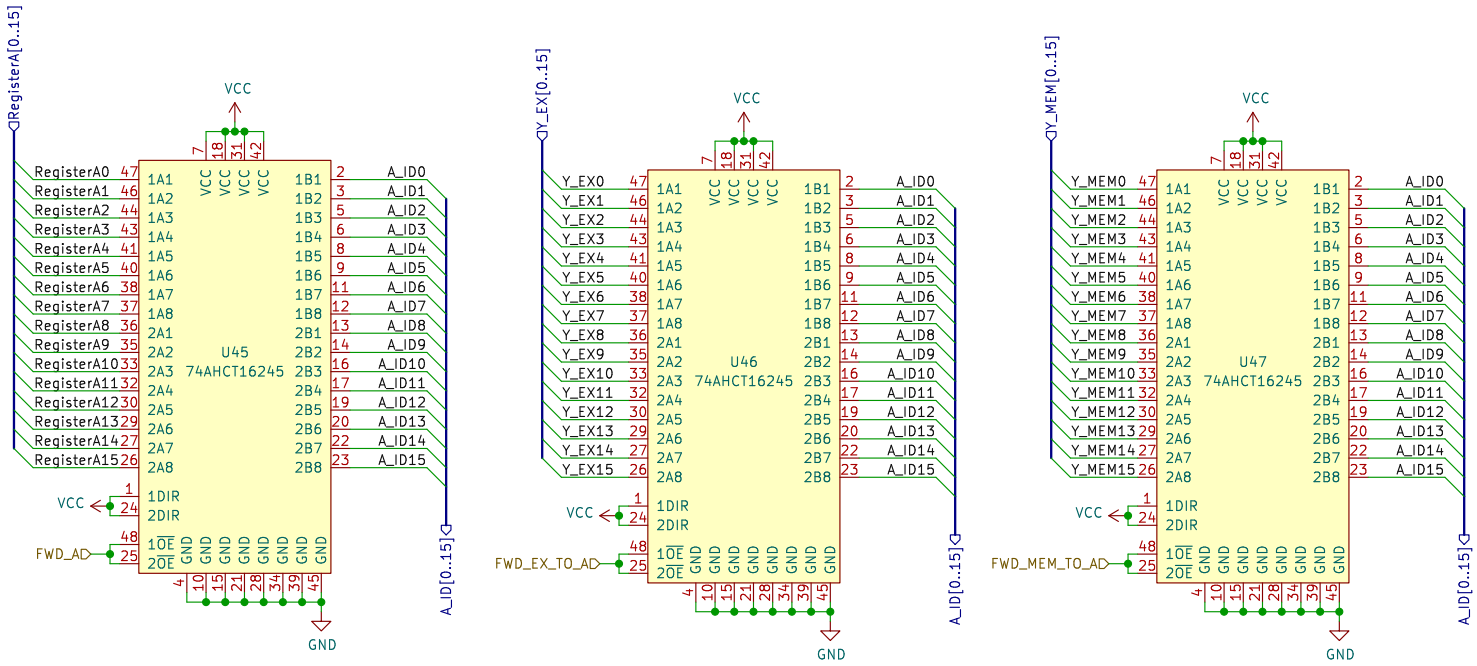


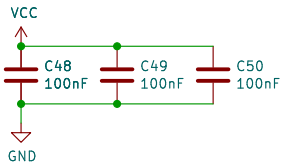
Triple Port Register File build from dual port SRAM

Sheet: /ID/sheet60693B94/
File: RegisterFile.sch

Title: Register File

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 35/39

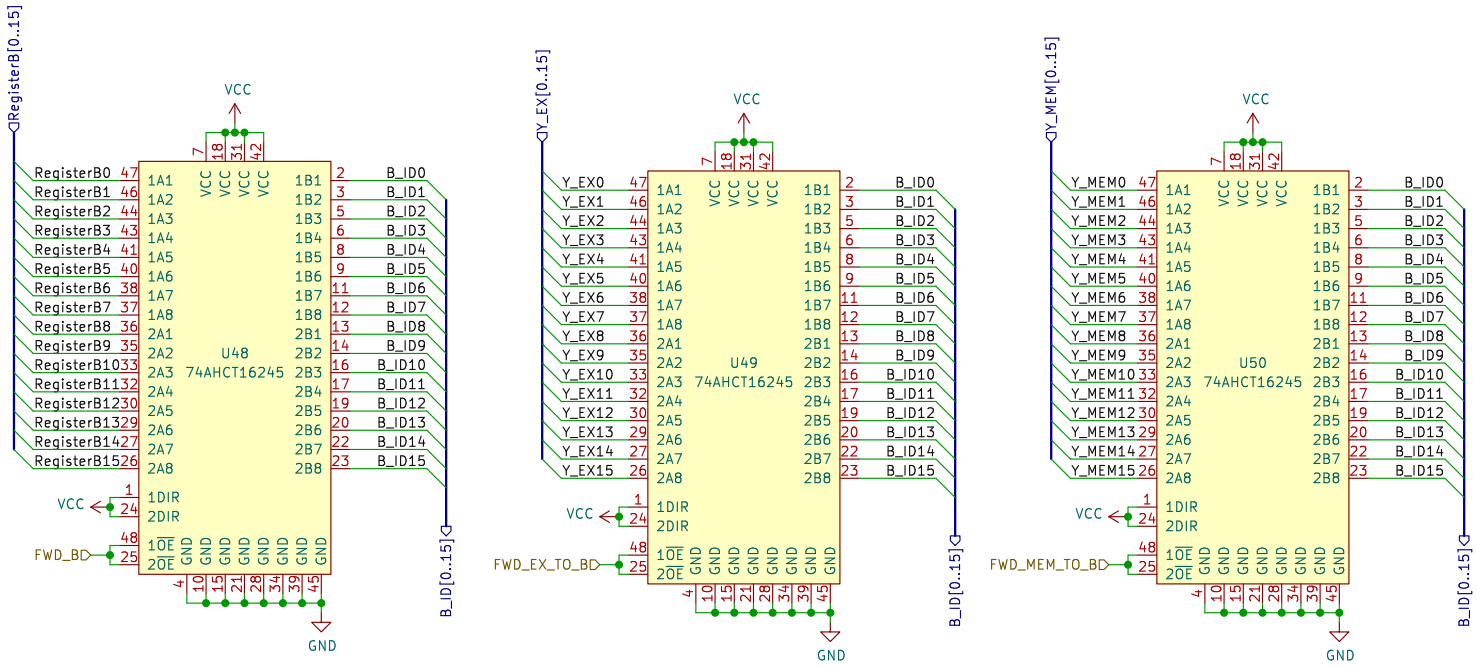


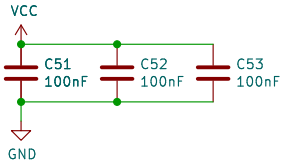


Sheet: /ID/Operand Forwarding A/
File: OperandForwardingA.sch

Title:

Size: A4	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 36/39

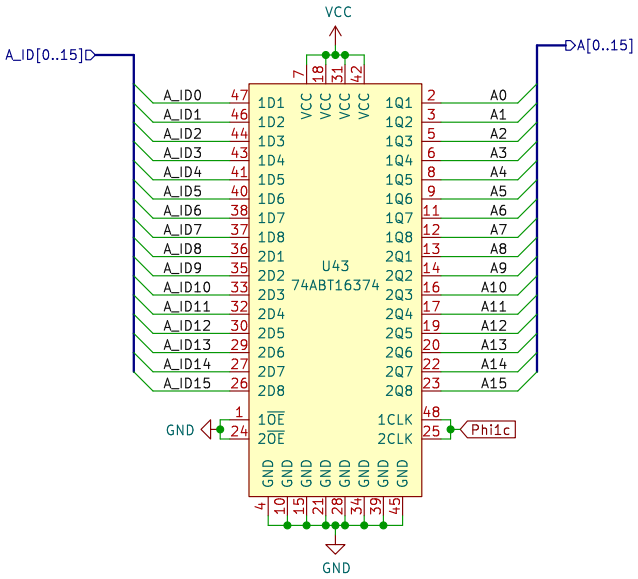




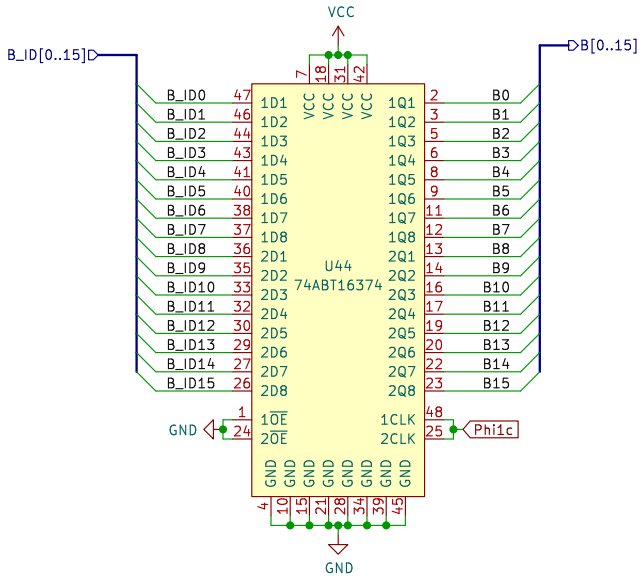
Sheet: /ID/Operand Forwarding B/
File: OperandForwardingB.sch

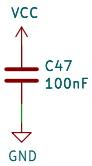
Title:

Size: A4	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 37/39



Sheet: /ID/sheet60693B9A/		
File: ID_EX_A.sch		
Title:		
Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 38/39





Sheet: /ID/sheet60693B9B/
File: ID_EX_B.sch

Title:

Size: USLetter	Date: 2021-04-28	**Rev: A (c8cebf3f)**
KiCad E.D.A. kicad 7.0.8		Id: 39/39