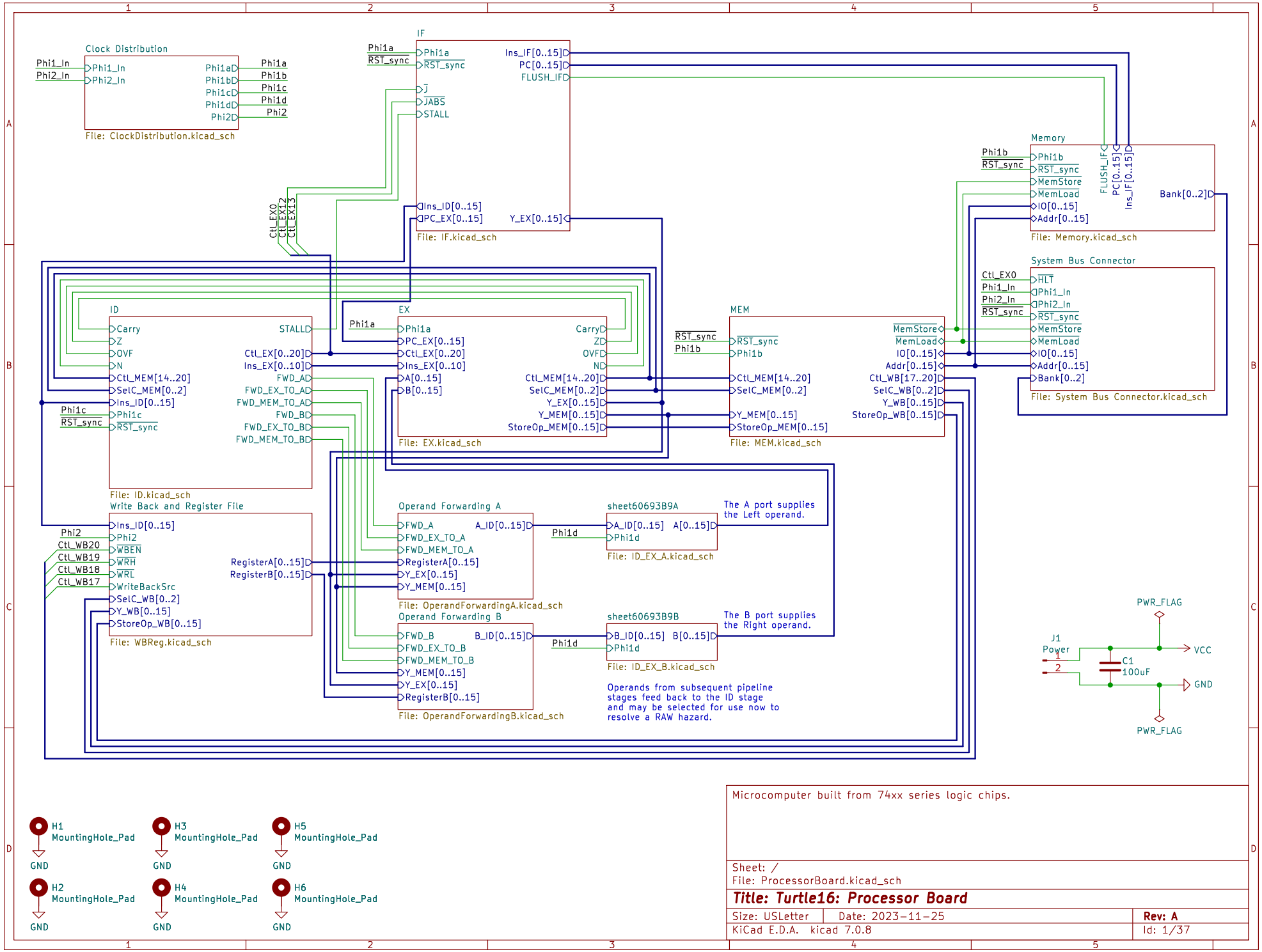
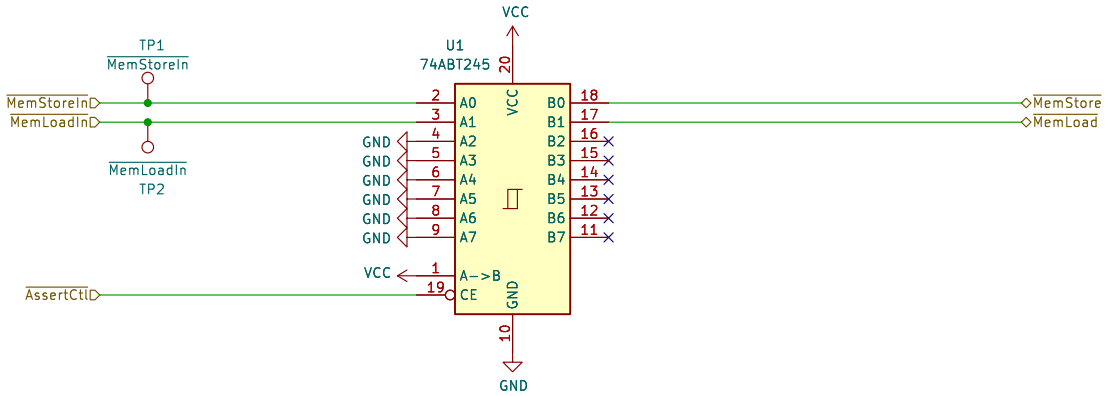

Clock Distribution
Phi1_In
Phi2_In
Phi1a
Phi1b
Phi1c
Phi1d
Phi2
File: ClockDistribution.kicad_sch
IF
Ins_IF[0..15]
PC[0..15]
FLUSH_IF
Ins_ID[0..15]
PC_EX[0..15]
Y_EX[0..15]
File: IF.kicad_sch
Memory
Bank[0..2]
File: Memory.kicad_sch
System Bus Connector
File: System Bus Connector.kicad_sch
ID
File: ID.kicad_sch
EX
File: EX.kicad_sch
MEM
File: MEM.kicad_sch
Write Back and Register File
File: WBReg.kicad_sch
Operand Forwarding A
File: OperandForwardingA.kicad_sch
Operand Forwarding B
File: OperandForwardingB.kicad_sch
sheet60693B9A
File: ID_EX_A.kicad_sch
sheet60693B9B
File: ID_EX_B.kicad_sch
The A port supplies the Left operand.
The B port supplies the Right operand.
Operands from subsequent pipeline stages feed back to the ID stage and may be selected for use now to resolve a RAW hazard.
J1
Power
C1
100uF
VCC
GND
PWR_FLAG
H1 MountingHole_Pad
H2 MountingHole_Pad
H3 MountingHole_Pad
H4 MountingHole_Pad
H5 MountingHole_Pad
H6 MountingHole_Pad
Microcomputer built from 74xx series logic chips.
Sheet: /
File: ProcessorBoard.kicad_sch
Title: Turtle16: Processor Board
Size: USLetter
Date: 2023-11-25
Rev: A
KiCad E.D.A. kicad 7.0.8
Id: 1/37

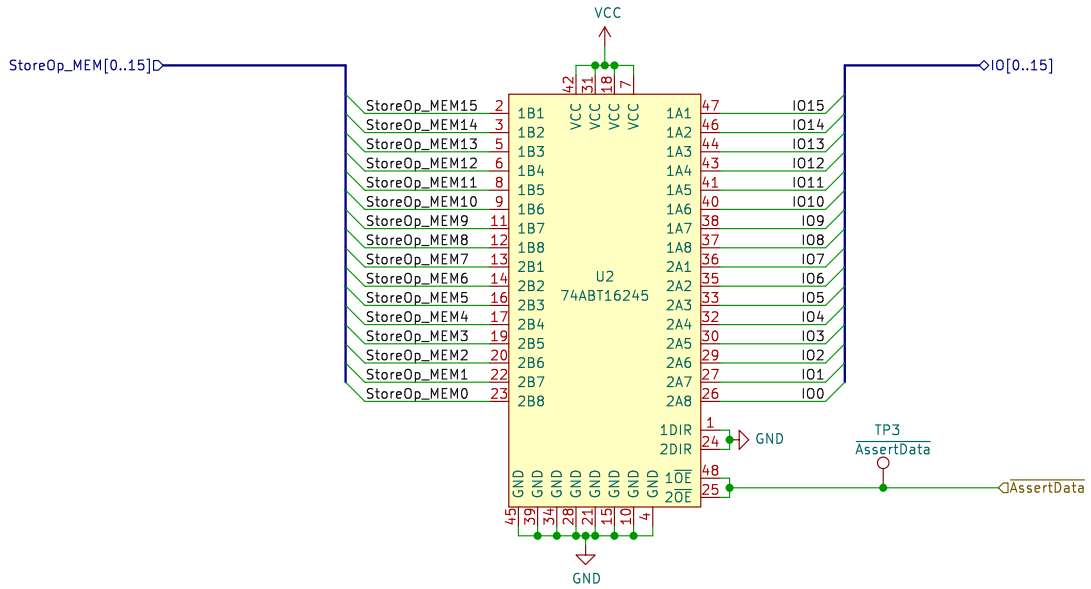




Sheet: /MEM/Buffer Memory Control Signals/
File: BufferMemoryControlSignals.kicad_sch

**Title:**

| | | |
|---|---|---|
| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
| KiCad E.D.A. kicad 7.0.8 | | Id: 4/37 |



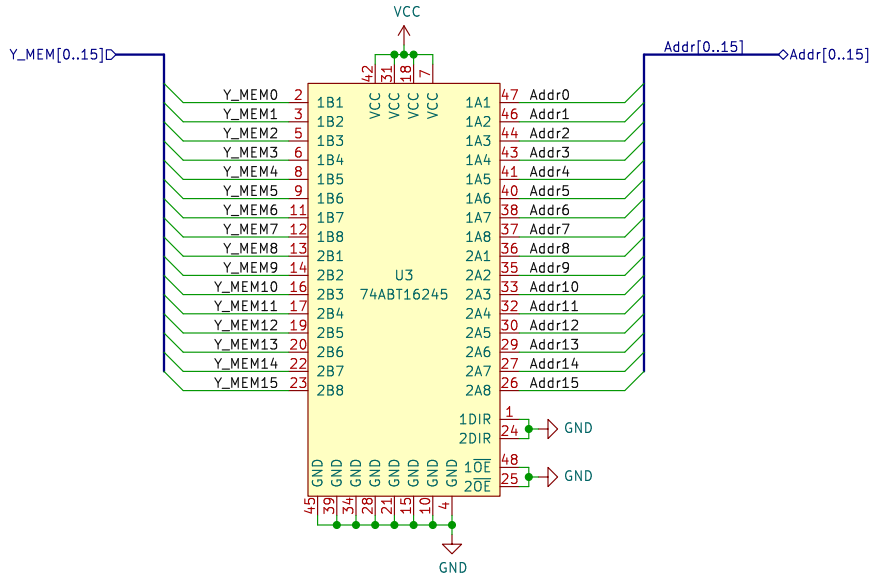


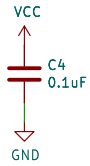
The store operand can be disconnected from the system memory bus via tristate outputs.
This allows peripheral devices to assert their own value to the bus during a Load.

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

**Title: MEM: Buffer StoreOp**

| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: 5/37 |



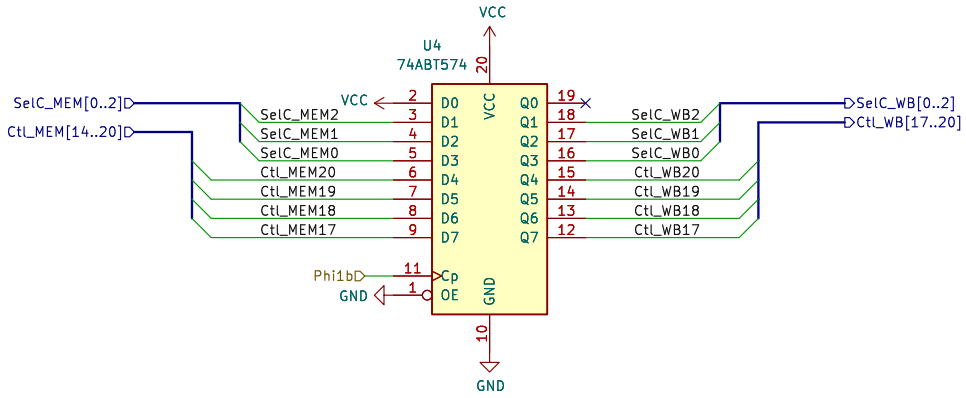



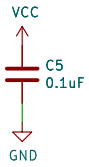

Buffer the effective address before it leaves the main board.

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

**Title: MEM: Buffer ALUResult**

| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: 6/37 |

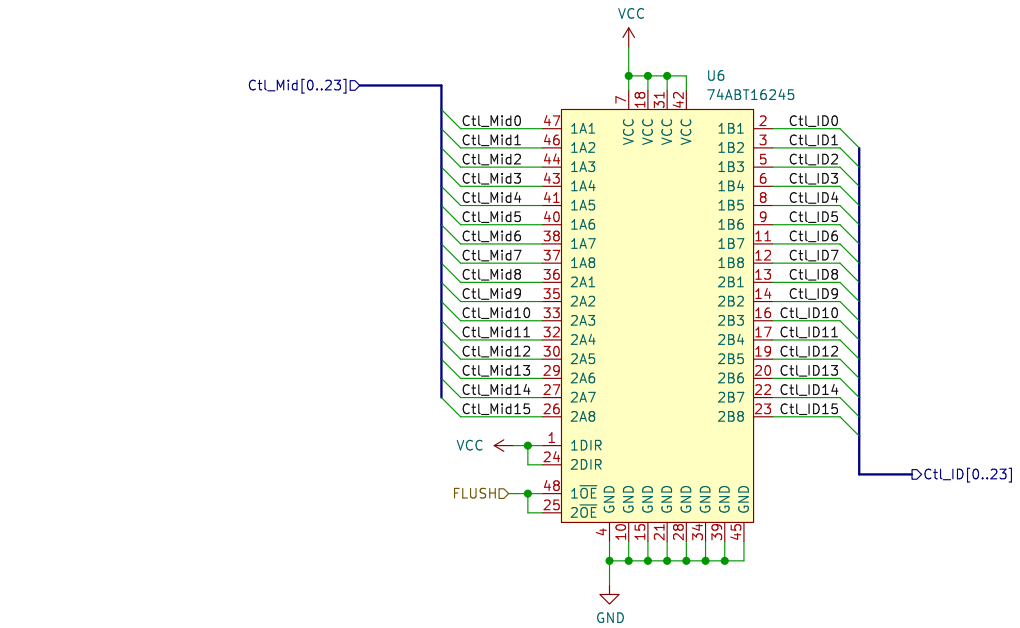


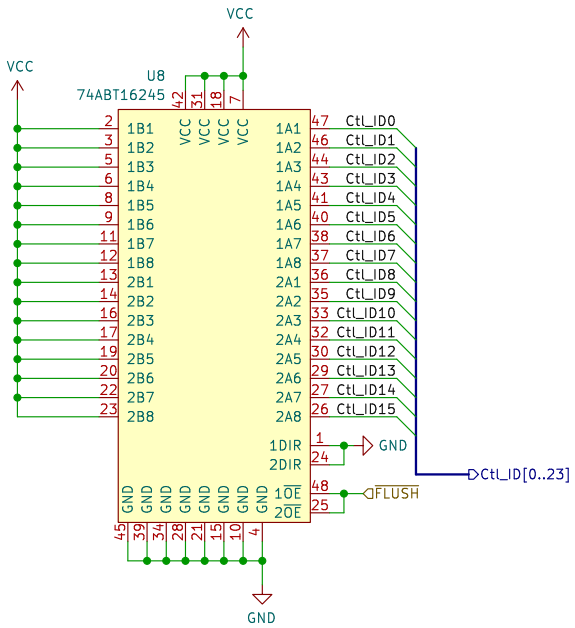



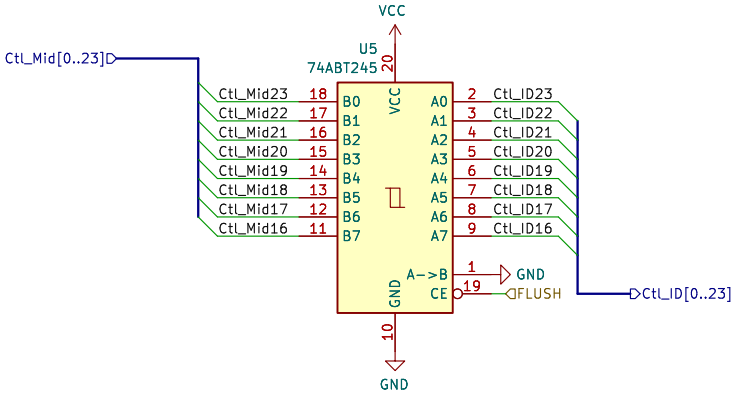

Sheet: /MEM/sheet5FD643DA/
File: Ctl_15_23_Register.kicad_sch

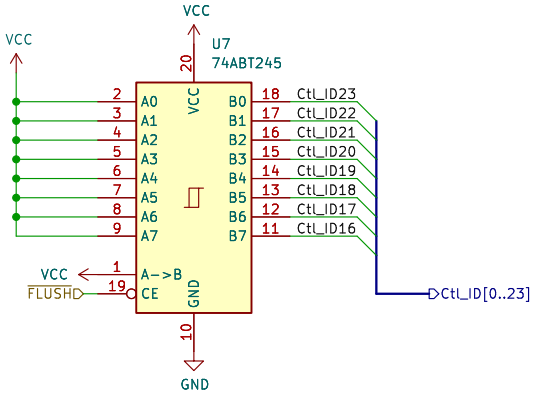
**Title: MEM/WB: Control Word Register**

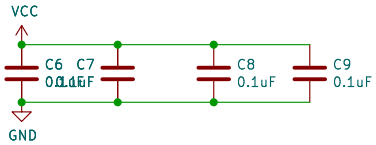
| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: 7/37 |



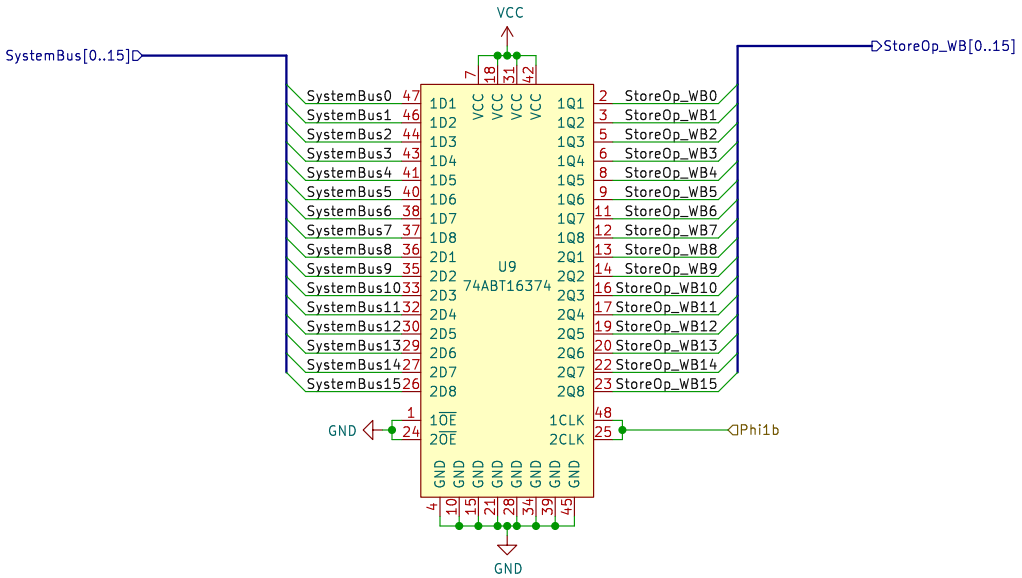




If the $\overline{\text{FLUSH}}$ signal is asserted then the control signal set presented to the interstage registers must be all ones, which represents a NOP.

Sheet: /ID/ID Flush/
File: IDFlush.kicad_sch

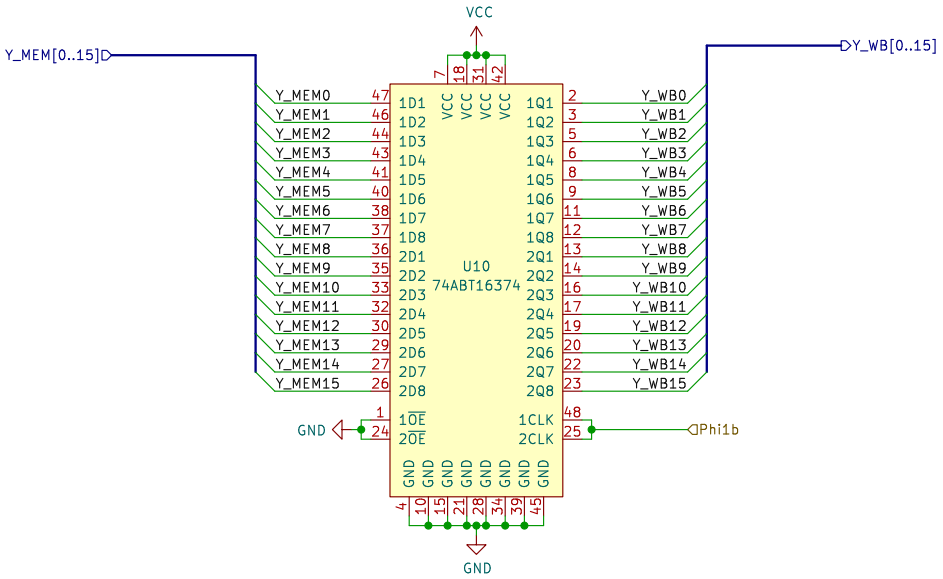
**Title:**

| Size: A4 | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: 8/37 |





| | |
|---|---|
| Sheet: /MEM/sheet5FD56BF4/ | |
| File: StoreOperandRegister3.kicad_sch | |
| **Title: MEM/WB: Store Operand Register** | |
| Size: USLetter \| Date: 2023-11-25 | **Rev: A** |
| KiCad E.D.A. kicad 7.0.8 | Id: 8/37 |

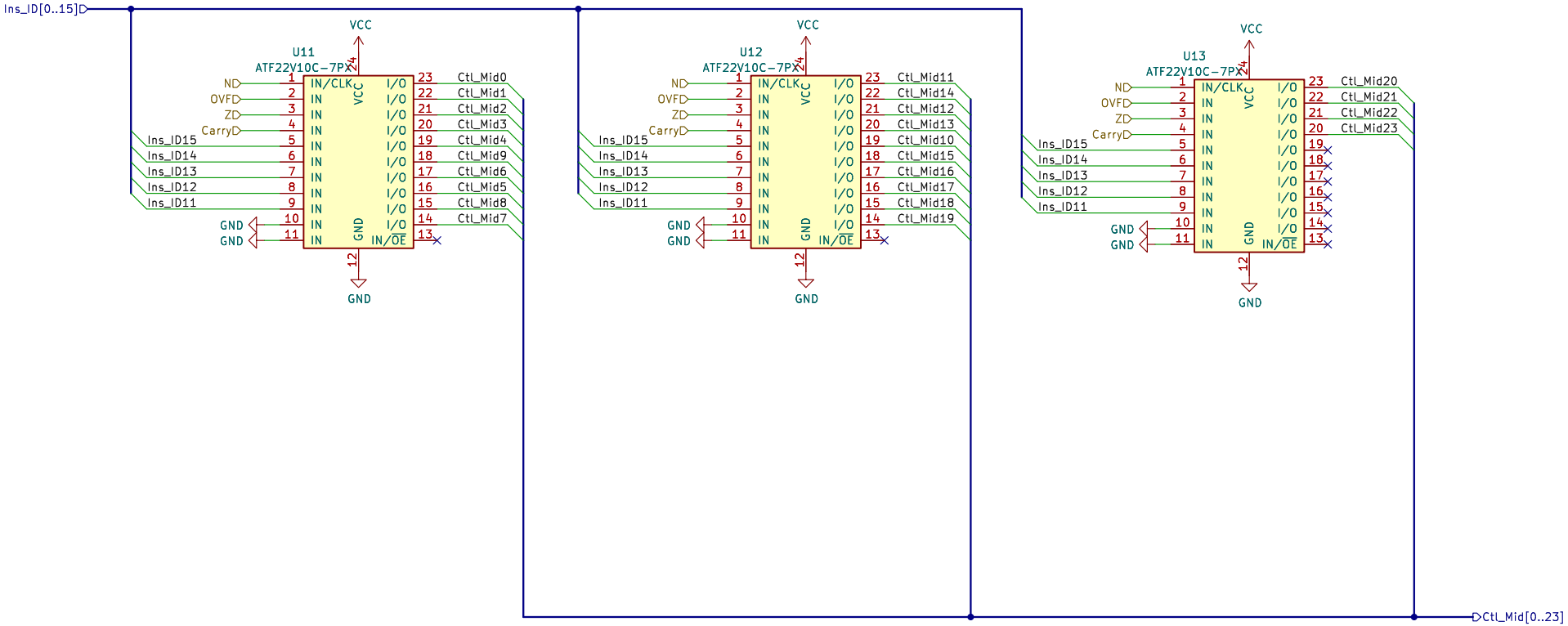


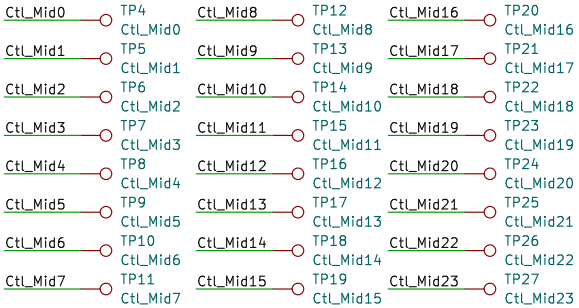


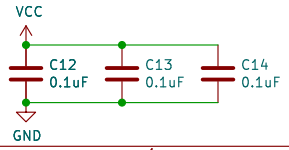
Sheet: /MEM/Sheet5FD9EFDA/
File: ALUResultRegister_MEM_WB.kicad_sch

**Title: ALU Result Register MEM/WB**

| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: 10/37 |




Decodes an instruction opcode into an array of control signals.

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

**Title: Instruction Decoder**

| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: 28/37 |

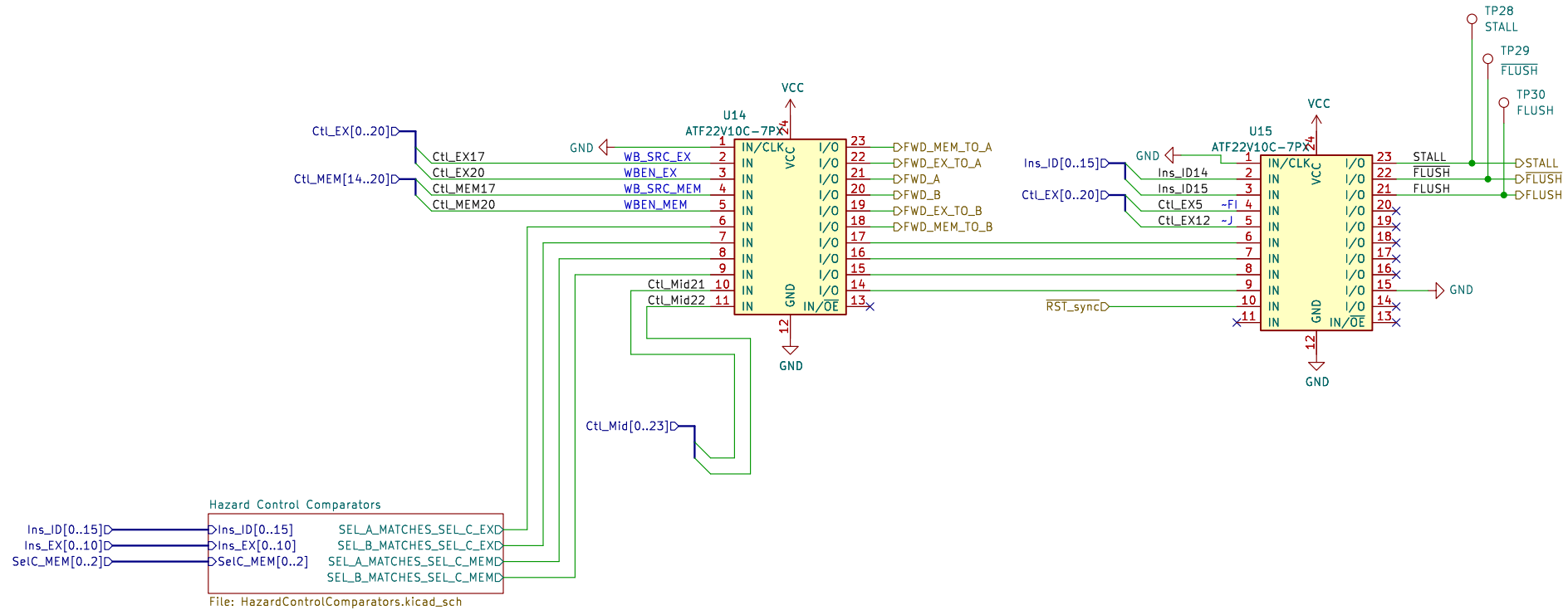
Hazard Control detects several types of hazards and resolves each one either by stalling the pipeline, or by forwarding an operand from a later pipeline stage.

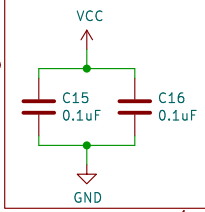
On a jump, the program counter must be allowed to load a new value. At the same time, the ID and IF stages must both be flushed.

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.



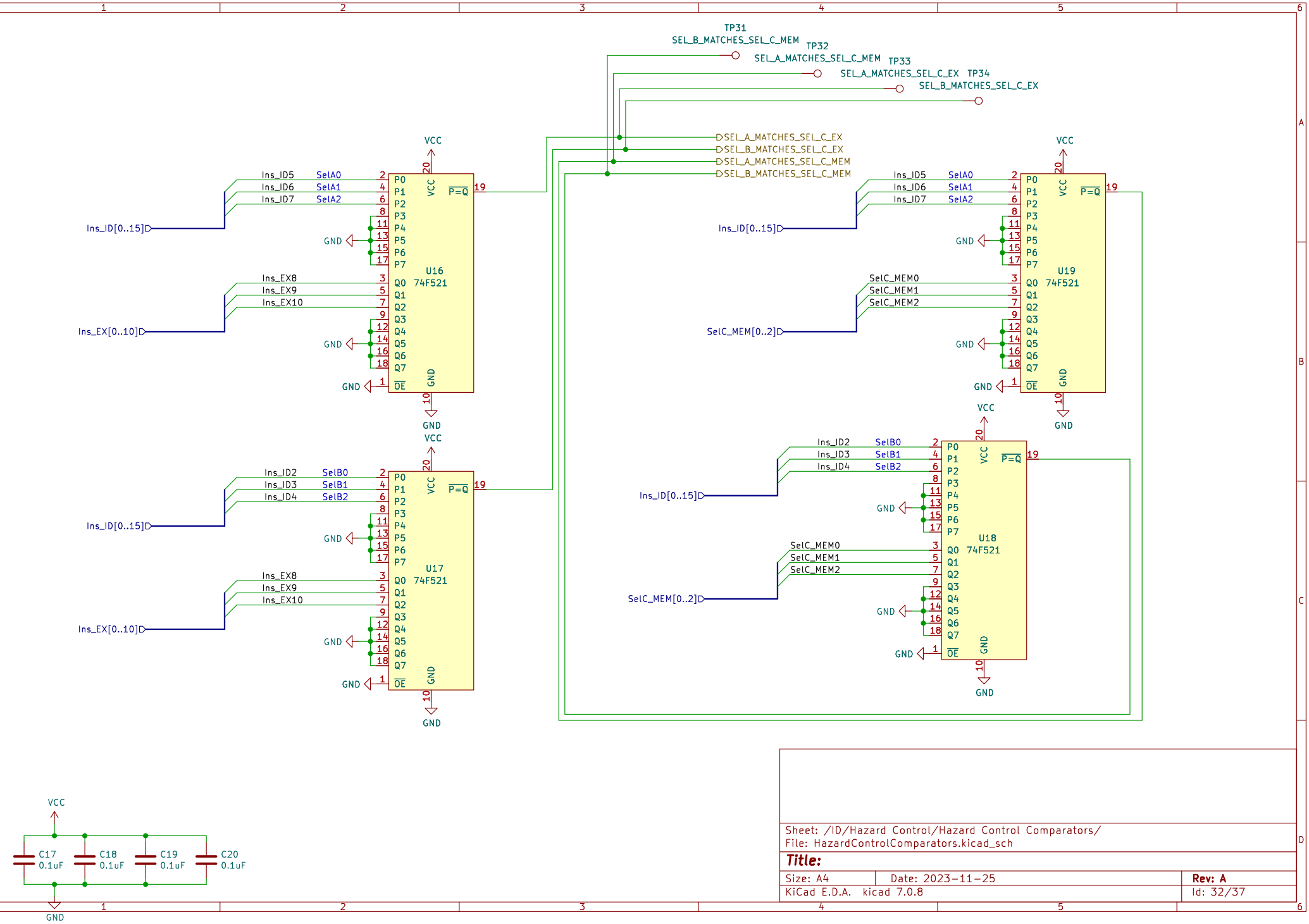



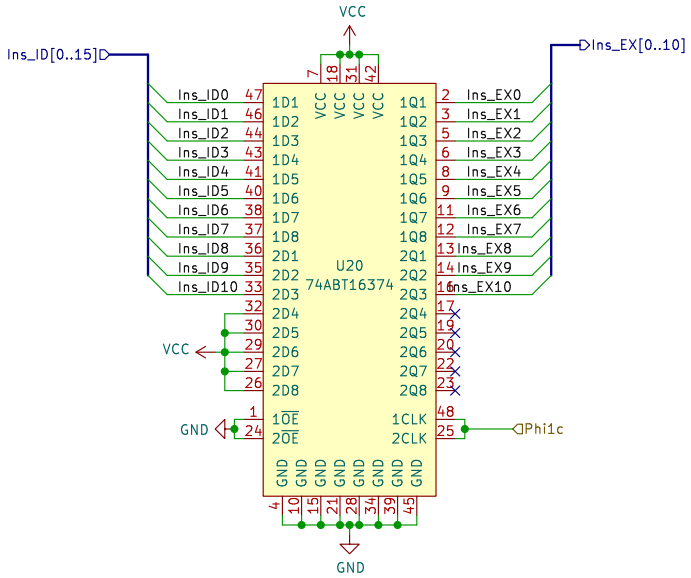

Control logic for dealing with pipeline hazards
This may stall the pipeline on a hazard. For RAW hazards, it produces signals to control operand forwarding.

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

**Title: Hazard Control**

| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: 31/37 |


TP31
SEL_B_MATCHES_SEL_C_MEM
TP32
SEL_A_MATCHES_SEL_C_MEM
TP33
SEL_A_MATCHES_SEL_C_EX
TP34
SEL_B_MATCHES_SEL_C_EX
SEL_A_MATCHES_SEL_C_EX
SEL_B_MATCHES_SEL_C_EX
SEL_A_MATCHES_SEL_C_MEM
SEL_B_MATCHES_SEL_C_MEM
U16
U17
U18
U19
74F521
VCC
GND
Ins_ID[0..15]
Ins_EX[0..10]
SelC_MEM[0..2]
Ins_ID5
Ins_ID6
Ins_ID7
SelA0
SelA1
SelA2
Ins_EX8
Ins_EX9
Ins_EX10
Ins_ID2
Ins_ID3
Ins_ID4
SelB0
SelB1
SelB2
SelC_MEM0
SelC_MEM1
SelC_MEM2
C17
C18
C19
C20
0.1uF
Sheet: /ID/Hazard Control/Hazard Control Comparators/
File: HazardControlComparators.kicad_sch
Title:
Size: A4
Date: 2023-11-25
Rev: A
KiCad E.D.A. kicad 7.0.8
Id: 32/37

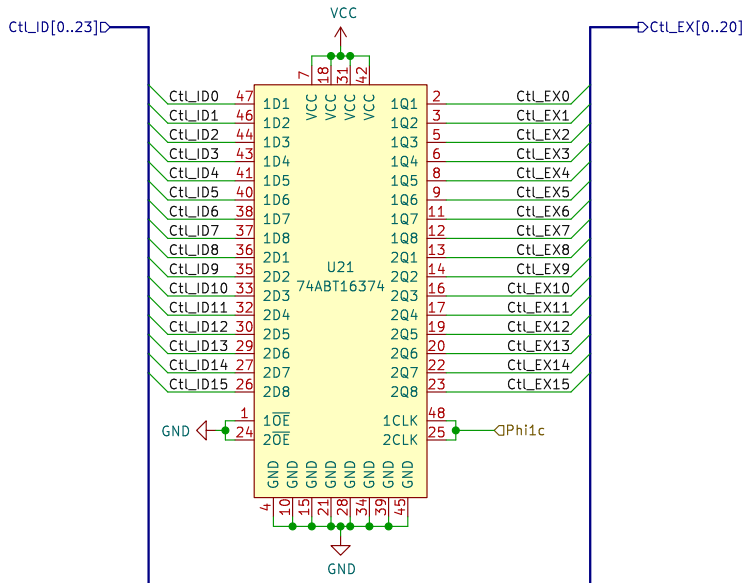


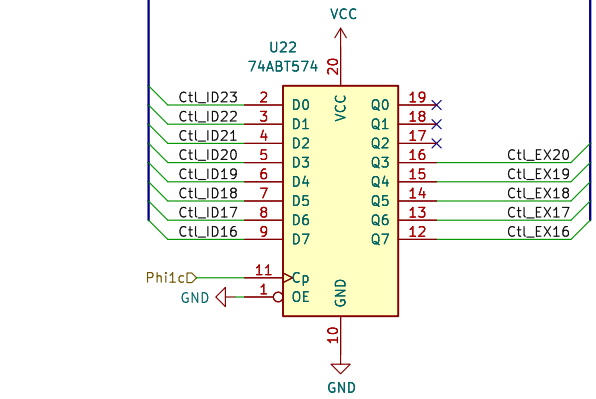


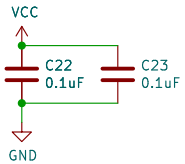
Sheet: /ID/sheet60693B99/
File: ID_EX_InstructionWord.kicad_sch

**Title:**

| | | |
|---|---|---|
| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
| KiCad E.D.A. kicad 7.0.8 | | Id: 33/37 |



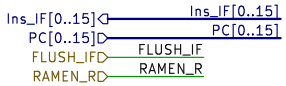



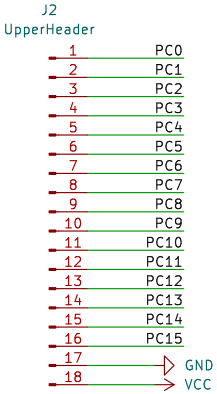

Sheet: /ID/sheet60693B98/
File: ID_EX_ControlWord.kicad_sch

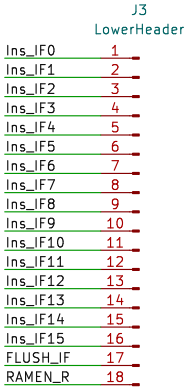
**Title: ID/EX Control Register**

Size: USLetter | Date: 2023-11-25 | **Rev: A**

KiCad E.D.A. kicad 7.0.8 | Id: 34/37




RAMEN_R enables the ROM.
The right side of the RAM is used for Instruction Fetch. If the IF stage is not being fed by RAM then it is being fed by this ROM.

Sheet: /Memory/ROM/
File: ROM.kicad_sch

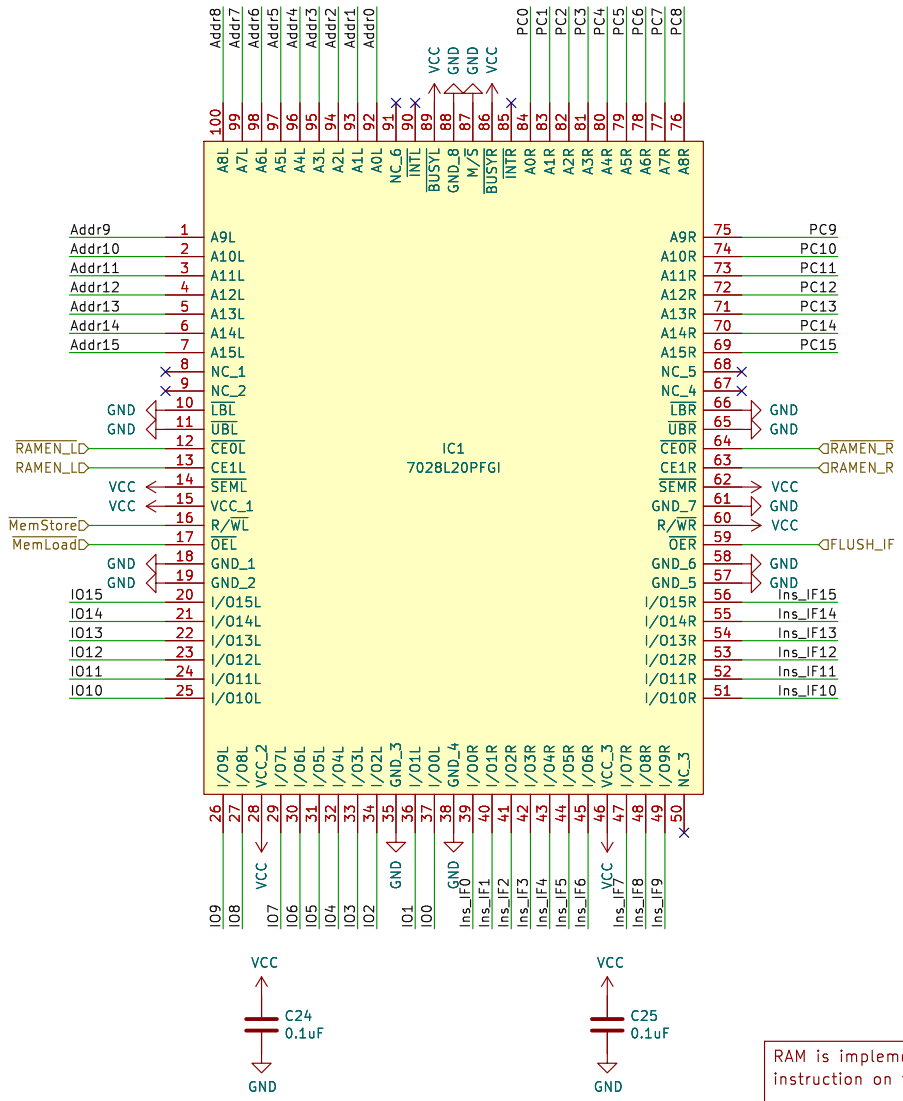
**Title:**

| | | |
|---|---|---|
| Size: A4 | Date: 2023-11-25 | **Rev: A** |
| KiCad E.D.A. kicad 7.0.8 | | Id: 14/37 |

The RAM is configured in $\overline{S}$ mode to disable the onboard write arbitration logic.

IO[0..15]
Addr[0..15]
Ins_IF[0..15]
PC[0..15]

If the IF stage is not being fed by ROM then it is being fed by RAM.





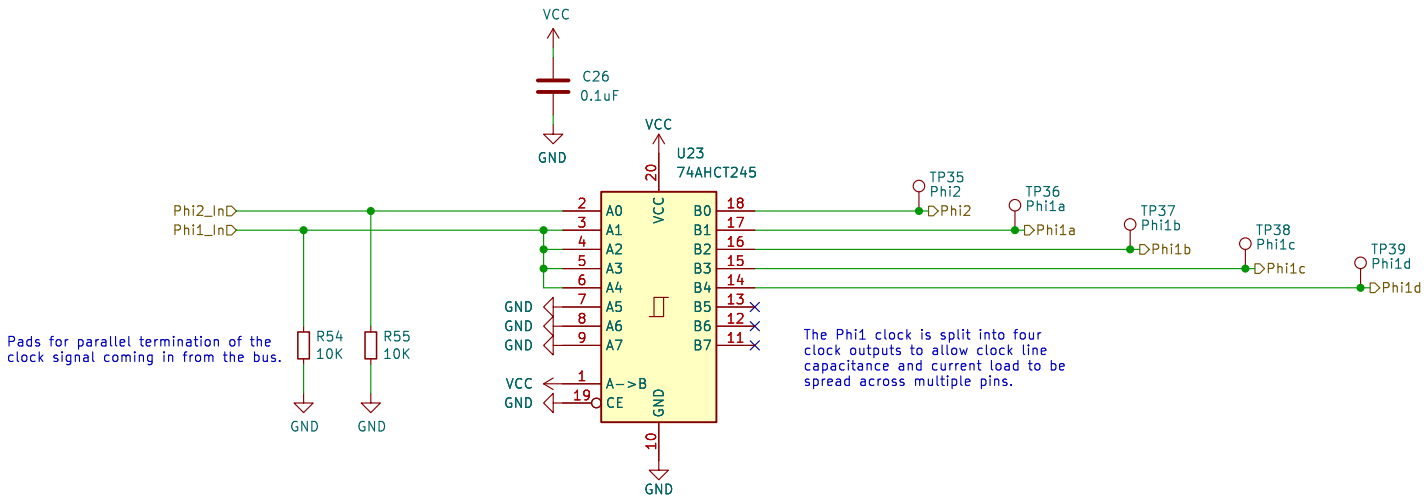


RAM is implemented with a 64K x 16 dual port SRAM so that the IF stage can fetch an instruction on the second read port.

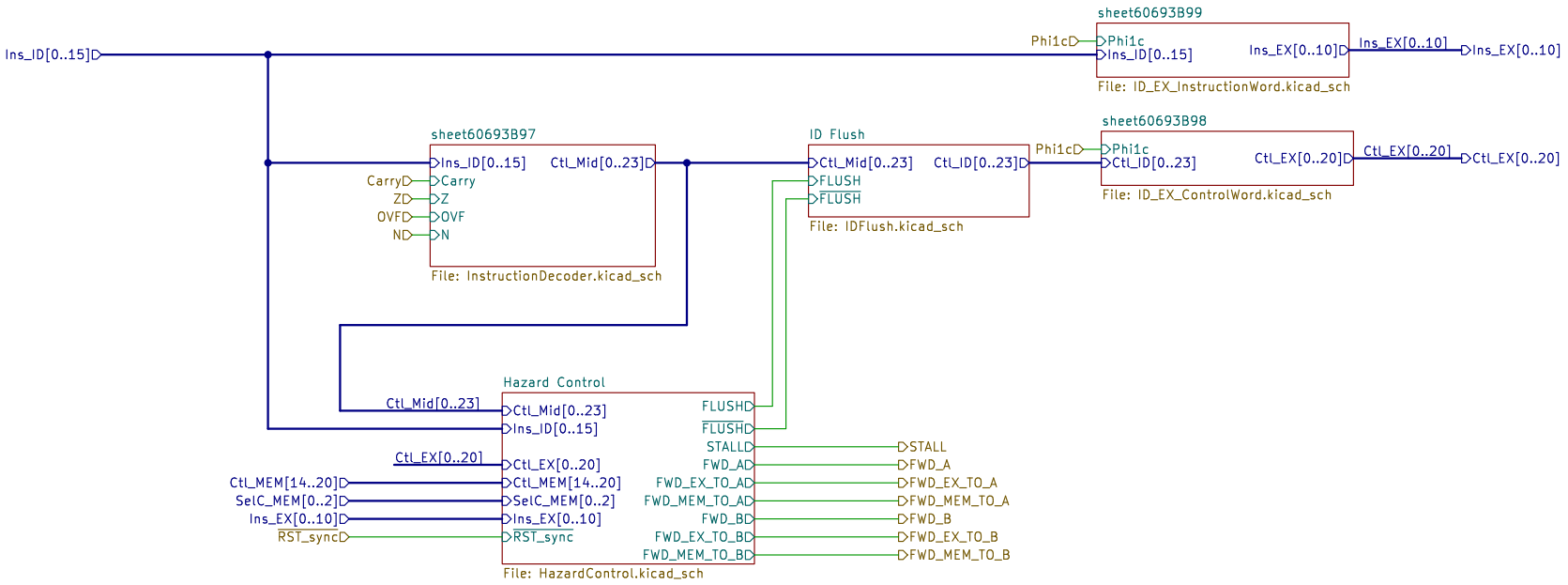
Sheet: /Memory/RAM/
File: RAM.kicad_sch

**Title: Turtle16: RAM**

| | | |
|---|---|---|
| Size: A4 | Date: 2023-11-25 | **Rev: A** |
| KiCad E.D.A. kicad 7.0.8 | | Id: 15/37 |




| | |
|---|---|
| Sheet: /Clock Distribution/ | |
| File: ClockDistribution.kicad_sch | |
| **Title: Turtle16: Clock Distribution** | |
| Size: USLetter \| Date: 2023-11-25 | **Rev: A** |
| KiCad E.D.A. kicad 7.0.8 | Id: #/37 |



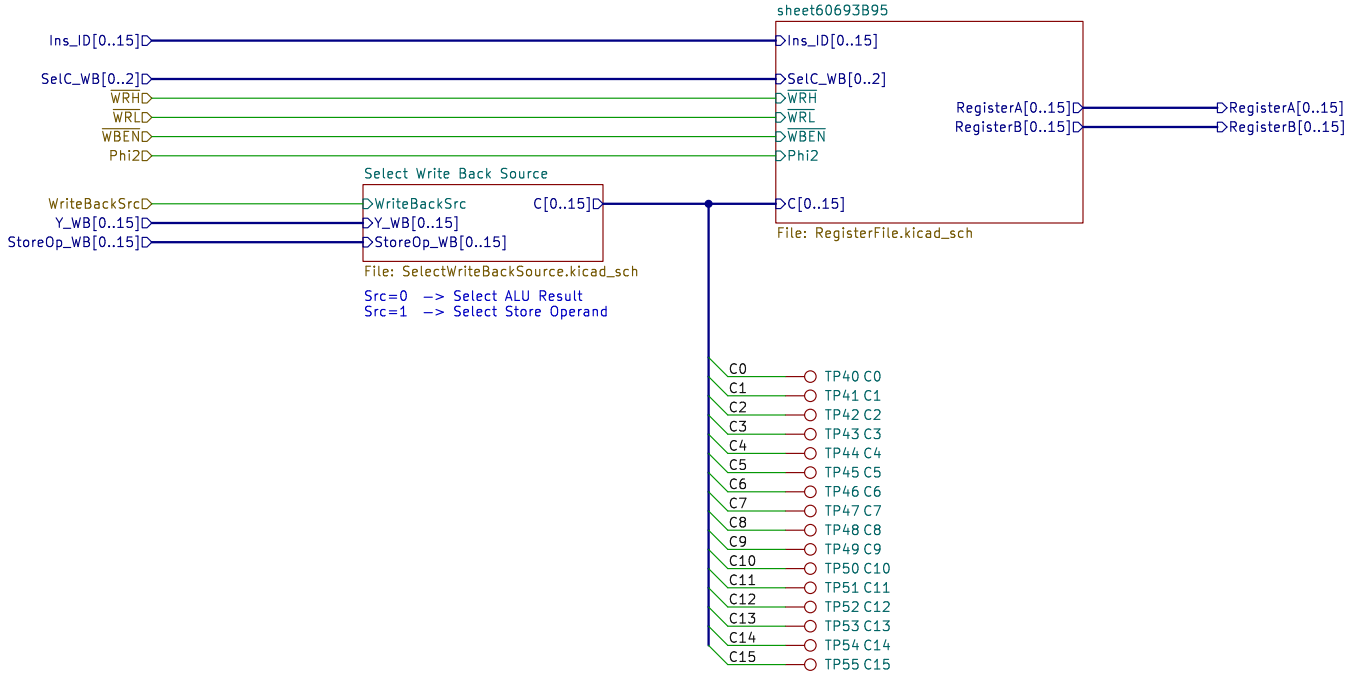

| # | Mnemonic | Description |
|---|---|---|
| 0 | /HLT | Halt Clock |
| 1 | SelStoreOpA | Select Store Operand 0 |
| 2 | SelStoreOpB | Select Store Operand 1 |
| 3 | SelRightOpA | Select Right Operand 1 |
| 4 | SelRightOpB | Select Right Operand 2 |
| 5 | /FI | Flags Register In |
| 6 | CarryIn | ALU 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 |
| 21 | LUNUSED | Left operand is unused |
| 22 | RUNUSED | Right operand is unused |

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.kicad_sch

**Title: ID**

| | | |
|---|---|---|
| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |

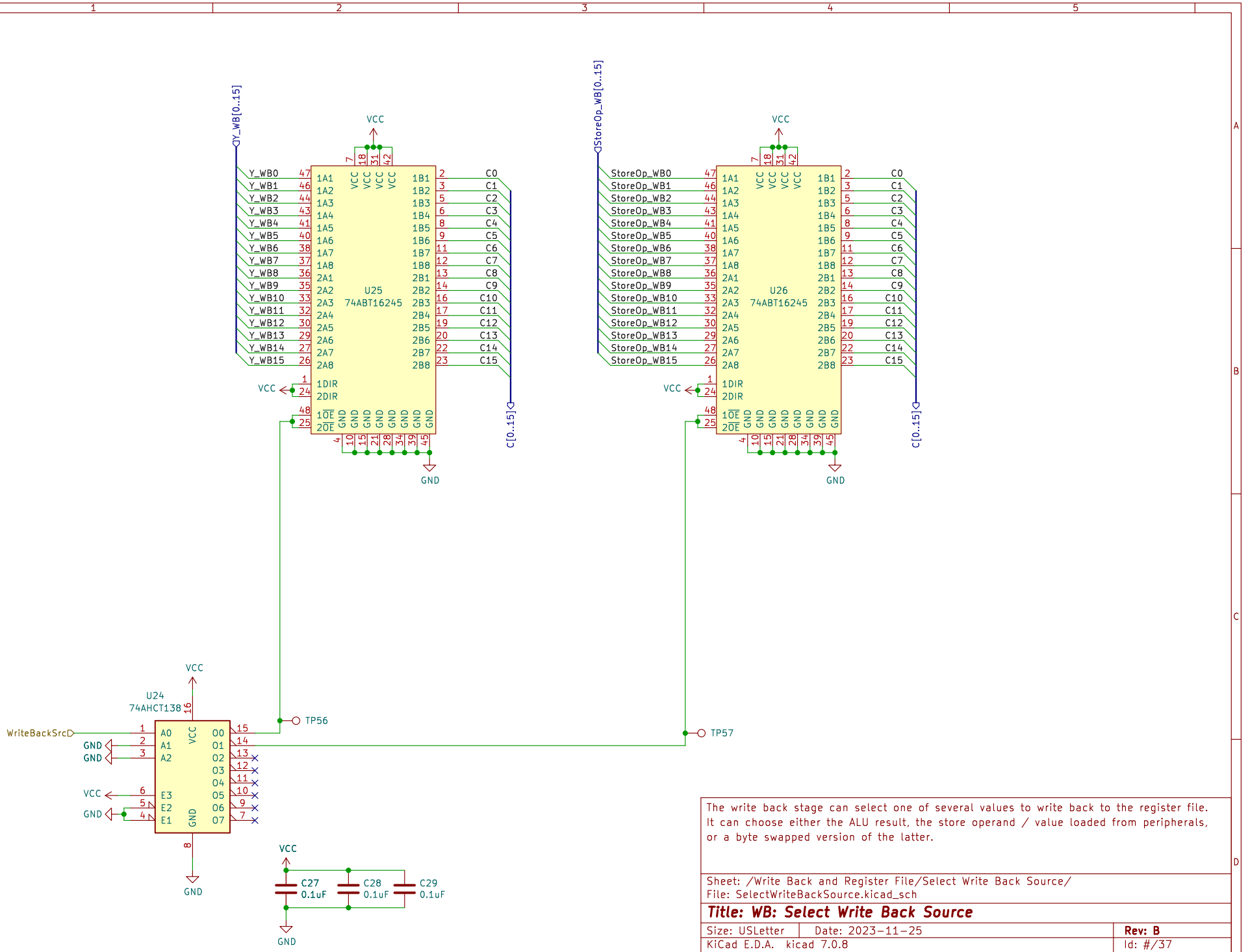



The Write Back Stage and Register File are combined on one PCB.

Sheet: /Write Back and Register File/
File: WBReg.kicad_sch

**Title:**

| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |



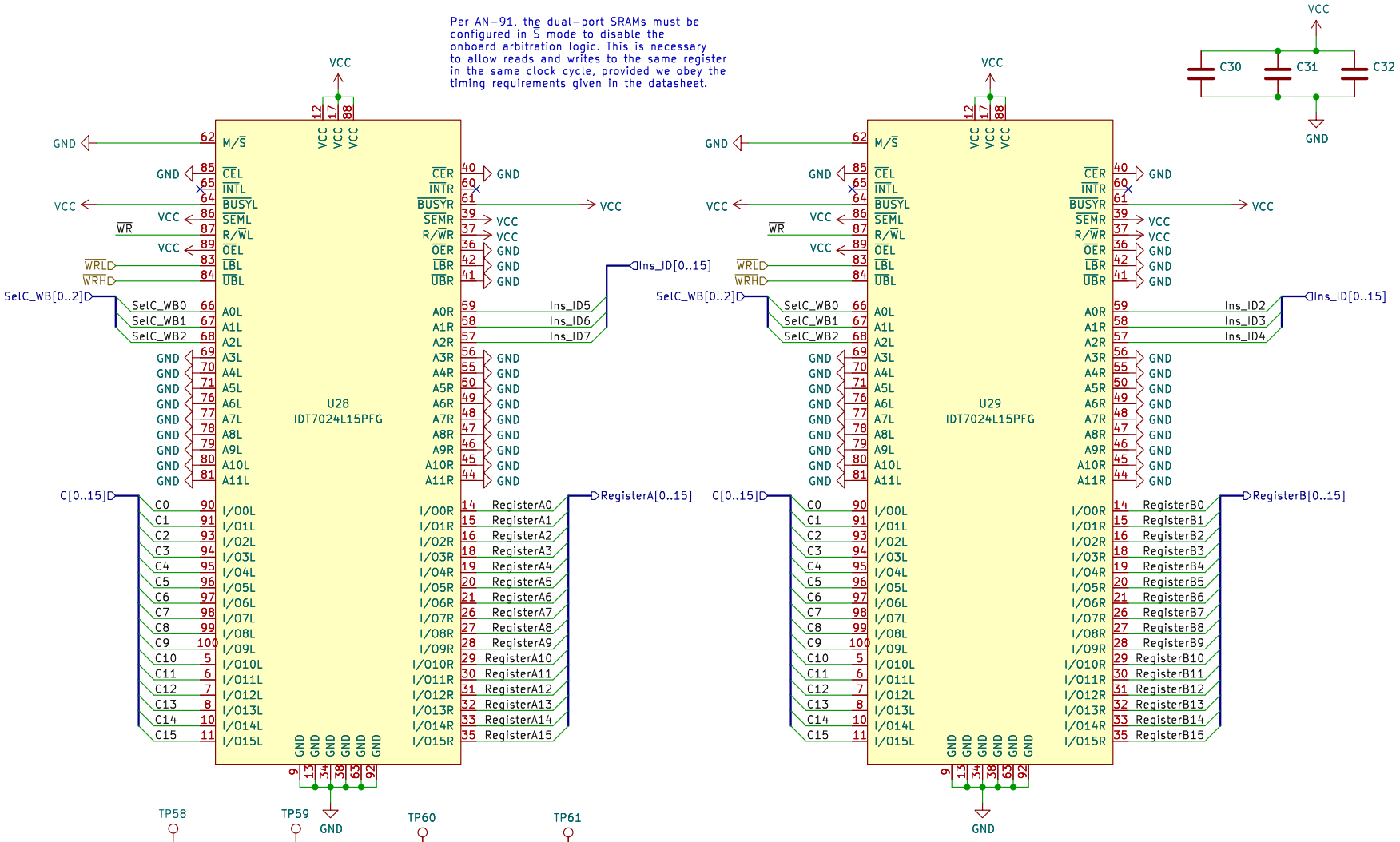

The write back stage can select one of several values to write back to the register file. It can choose either the ALU result, the store operand / value loaded from peripherals, or a byte swapped version of the latter.

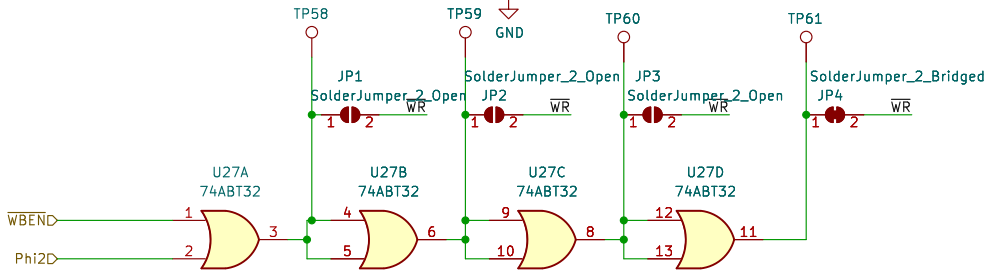
Sheet: /Write Back and Register File/Select Write Back Source/
File: SelectWriteBackSource.kicad_sch

**Title: WB: Select Write Back Source**

| Size: USLetter | Date: 2023-11-25 | **Rev: B** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |

Per AN-91, the dual-port SRAMs must be configured in S mode to disable the onboard arbitration logic. This is necessary to allow reads and writes to the same register in the same clock cycle, provided we obey the timing requirements given in the datasheet.




Ensure the $\overline{WR}$ signal goes high well before the next rising edge of Phi1.

0.1uF

Delay the $\overline{WR}$ signal by up to 8ns per gate. The three OR-gates cause the $\overline{WR}$ signal to appear on the dual-port SRAM about 24ns after the other inputs. This is necessary to ensure correct values are written to the SRAM.

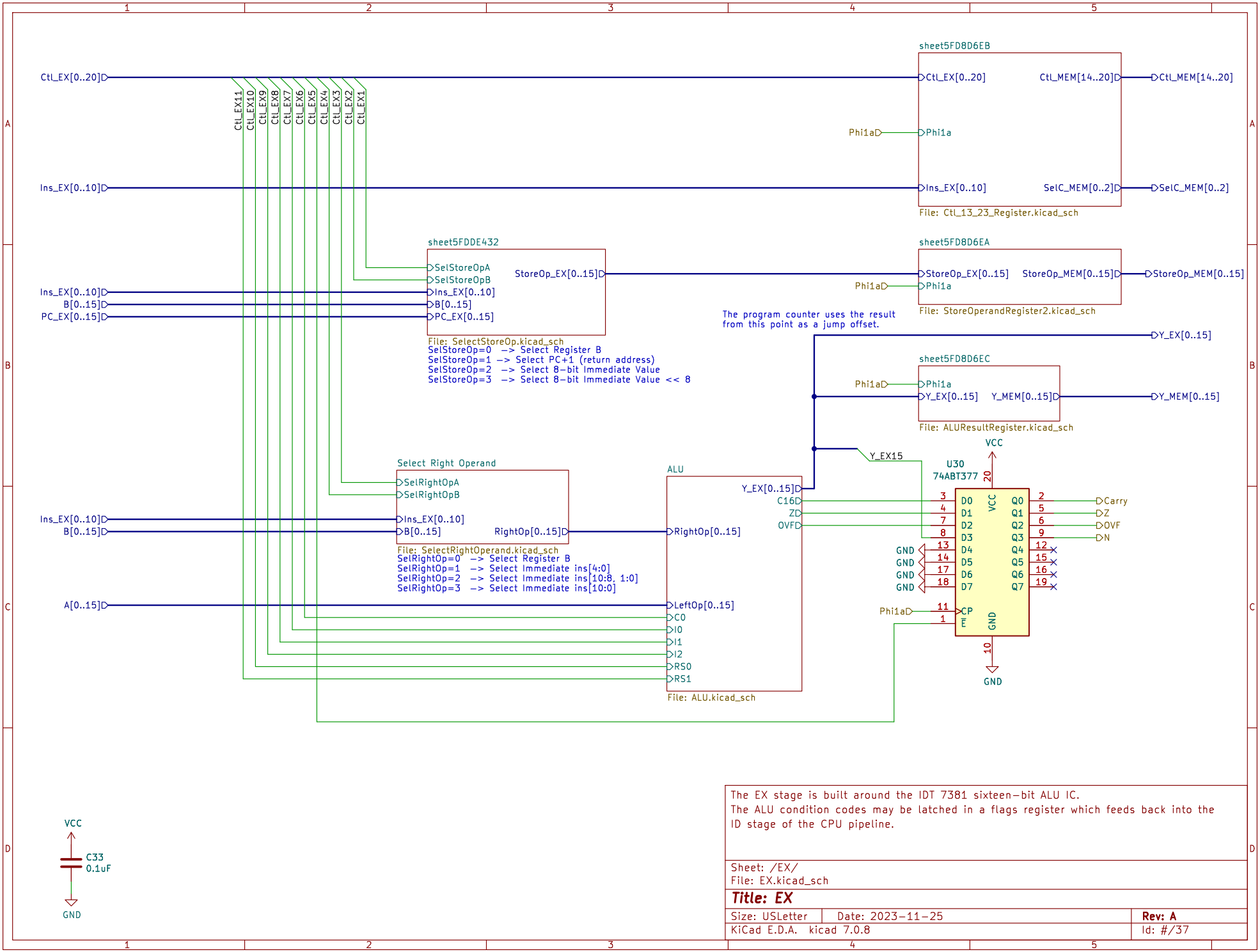
The data sheet notes that any of various control signals, including $\overline{WR}$ must remain high while address bits are changing.

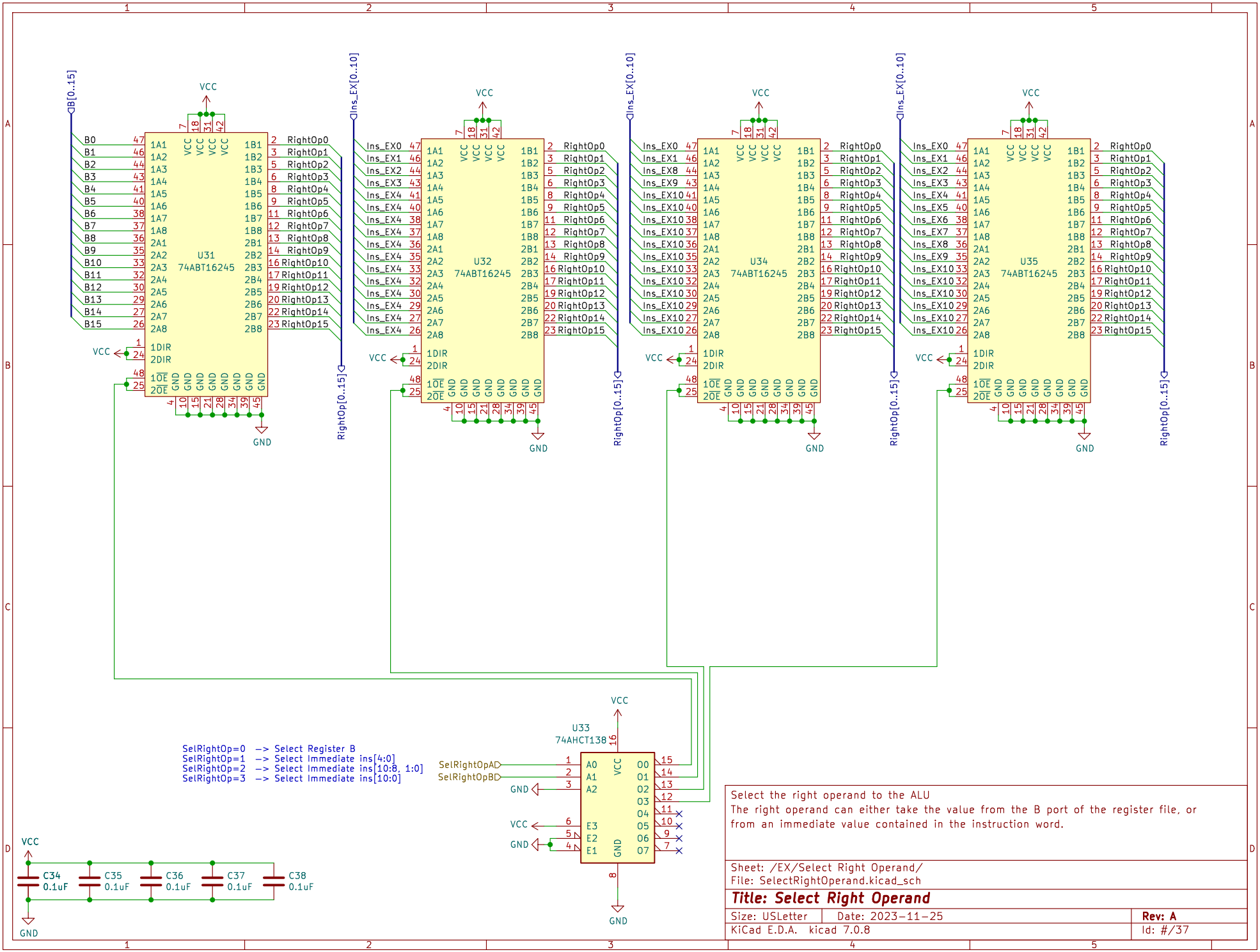
Triple Port Register File build from dual port SRAM

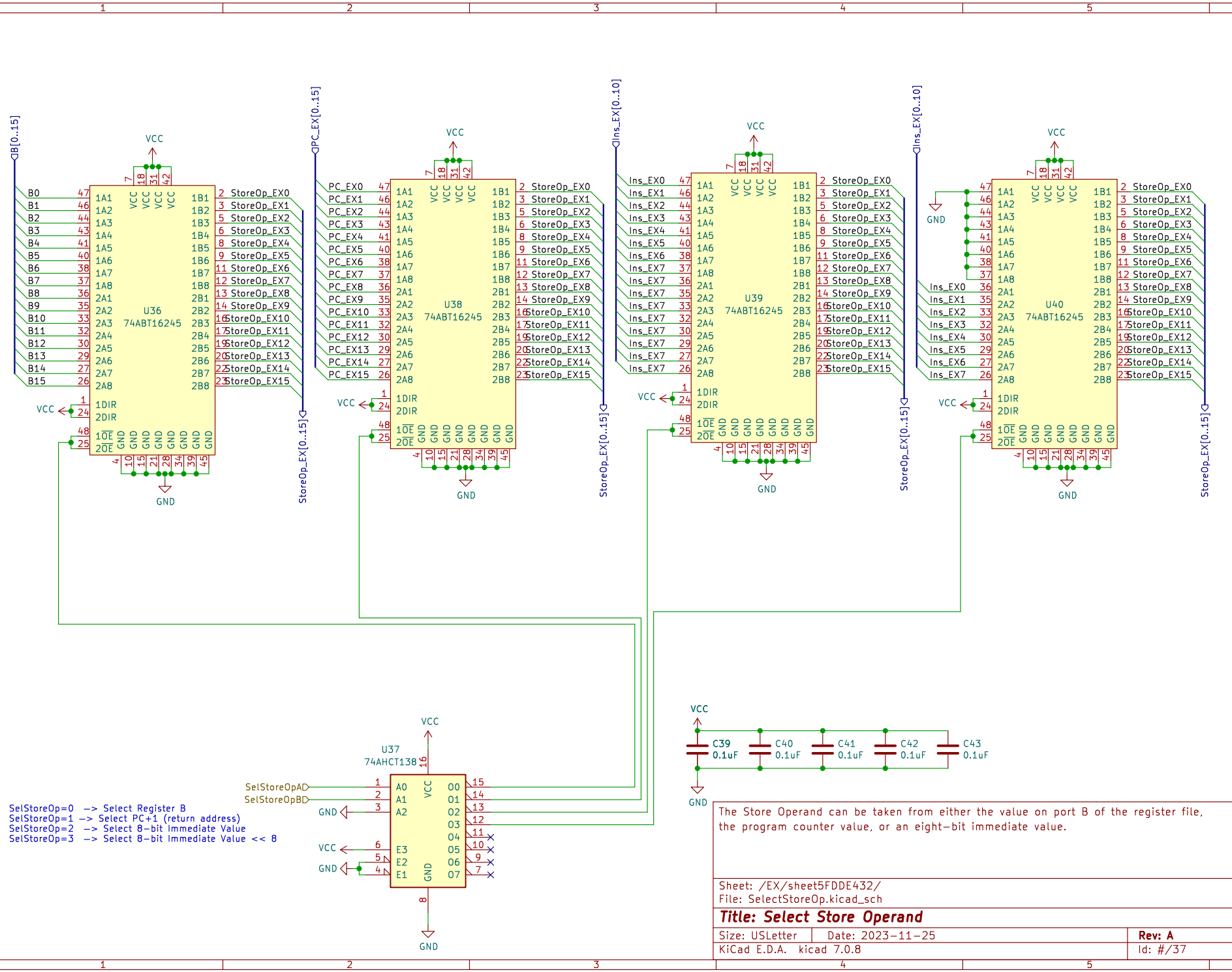
Sheet: /Write Back and Register File/sheet60693B95/
File: RegisterFile.kicad_sch

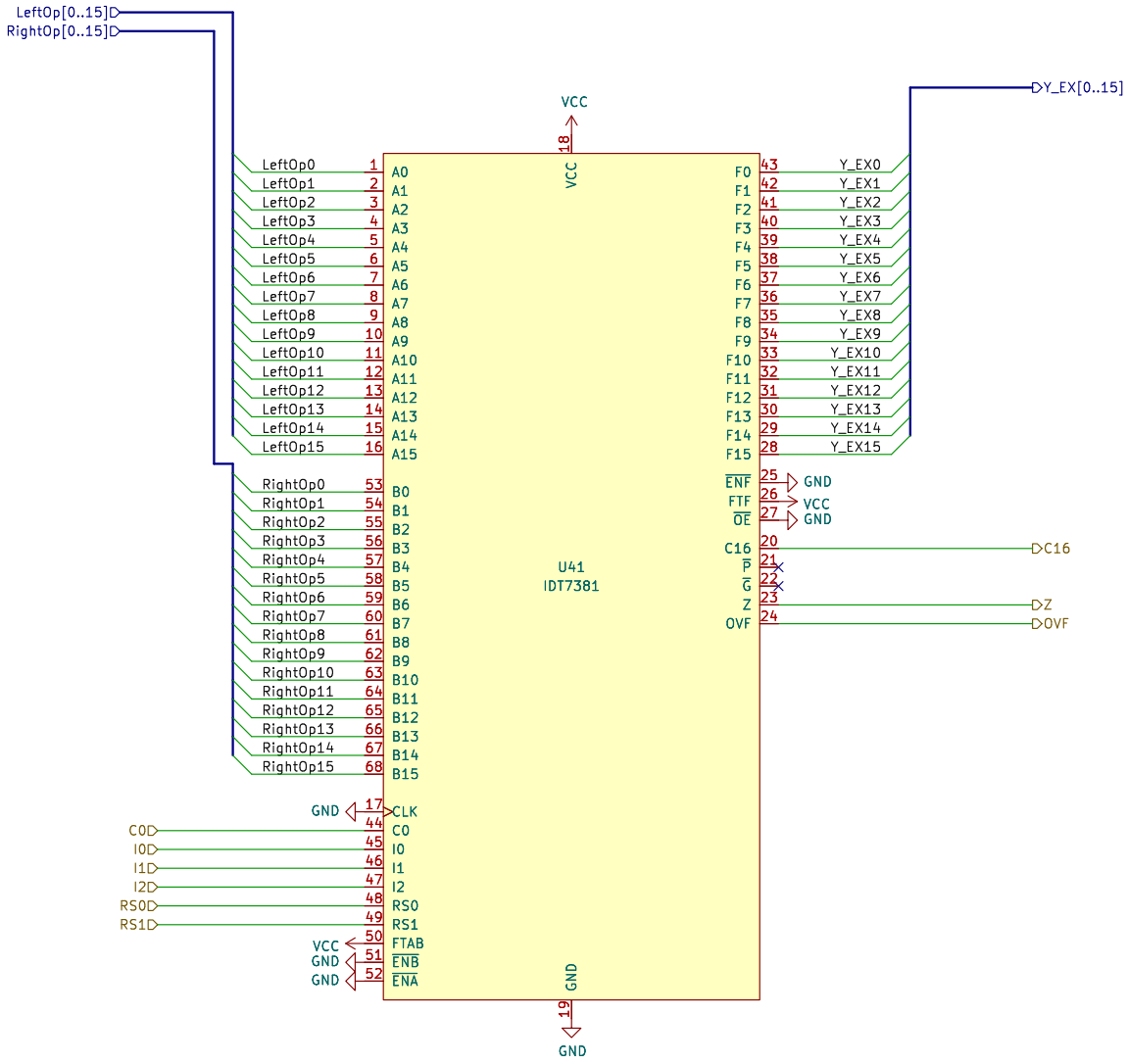
**Title: Register File**

| Size: USLetter | Date: 2023-11-25 | **Rev: B** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |


sheet5FD8D6EB
Ctl_EX[0..20]
Ctl_EX[0..20]
Ctl_MEM[14..20]
Ctl_MEM[14..20]
Phi1a
Phi1a
Phi1a
Ins_EX[0..10]
Ins_EX[0..10]
SelC_MEM[0..2]
SelC_MEM[0..2]
File: Ctl_13_23_Register.kicad_sch
Ctl_EX11
Ctl_EX10
Ctl_EX9
Ctl_EX8
Ctl_EX7
Ctl_EX6
Ctl_EX5
Ctl_EX4
Ctl_EX3
Ctl_EX2
Ctl_EX1
sheet5FDDE432
SelStoreOpA
SelStoreOpB
StoreOp_EX[0..15]
Ins_EX[0..10]
B[0..15]
PC_EX[0..15]
Ins_EX[0..10]
B[0..15]
PC_EX[0..15]
File: SelectStoreOp.kicad_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]
Phi1a
Phi1a
File: StoreOperandRegister2.kicad_sch
The program counter uses the result
from this point as a jump offset.
Y_EX[0..15]
sheet5FD8D6EC
Phi1a
Phi1a
Y_EX[0..15]
Y_MEM[0..15]
Y_MEM[0..15]
File: ALUResultRegister.kicad_sch
Y_EX15
VCC
U30
74ABT377
20
3
4
7
8
13
14
17
18
11
1
10
D0
D1
D2
D3
D4
D5
D6
D7
CP
E
VCC
GND
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
2
5
6
9
12
15
16
19
Carry
Z
OVF
N
GND
GND
GND
GND
Phi1a
GND
Select Right Operand
SelRightOpA
SelRightOpB
Ins_EX[0..10]
B[0..15]
RightOp[0..15]
Ins_EX[0..10]
B[0..15]
File: SelectRightOperand.kicad_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.kicad_sch
VCC
C33
0.1uF
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.kicad_sch
Title: EX
Size: USLetter
Date: 2023-11-25
Rev: A
KiCad E.D.A. kicad 7.0.8
Id: #/37


B[0..15]
Ins_EX[0..10]
RightOp[0..15]
U31
U32
U34
U35
74ABT16245
VCC
GND
U33
74AHCT138
SelRightOpA
SelRightOpB
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]
C34 0.1uF
C35 0.1uF
C36 0.1uF
C37 0.1uF
C38 0.1uF
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.kicad_sch
Title: Select Right Operand
Size: USLetter
Date: 2023-11-25
Rev: A
KiCad E.D.A. kicad 7.0.8
Id: #/37


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]
U36
74ABT16245
U38
74ABT16245
U39
74ABT16245
U40
74ABT16245
U37
74AHCT138
VCC
GND
SelStoreOpA
SelStoreOpB
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
C39 0.1uF
C40 0.1uF
C41 0.1uF
C42 0.1uF
C43 0.1uF
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.kicad_sch
Title: Select Store Operand
Size: USLetter
Date: 2023-11-25
Rev: A
KiCad E.D.A. kicad 7.0.8
Id: #/37

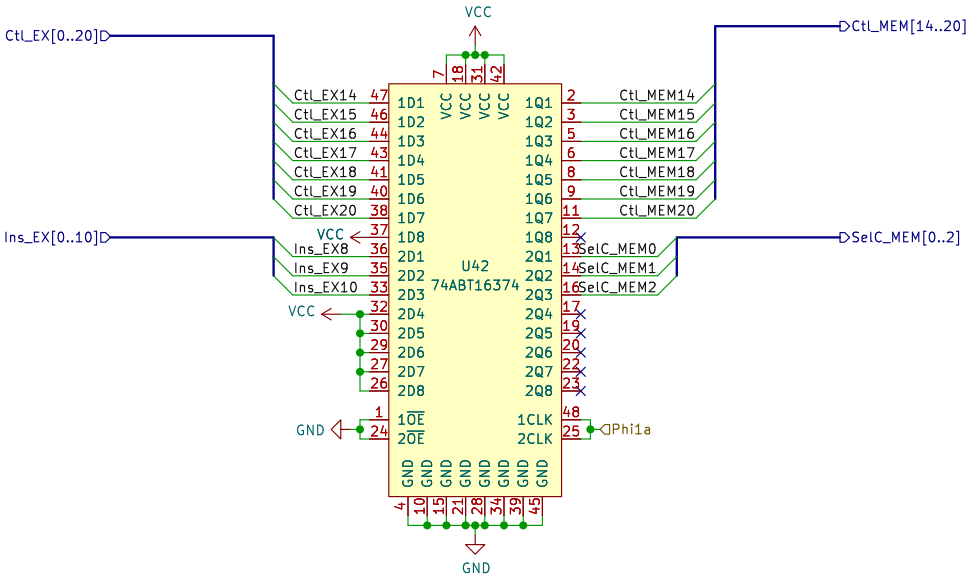



The ALU is built around the IDT 7381 sixteen-bit monolithic ALU IC.
This is configured for pure combinatorial operation where operands and outputs are latched in registers in other parts of the circuit.

Sheet: /EX/ALU/
File: ALU.kicad_sch

**Title: ALU**

| | | |
|---|---|---|
| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |

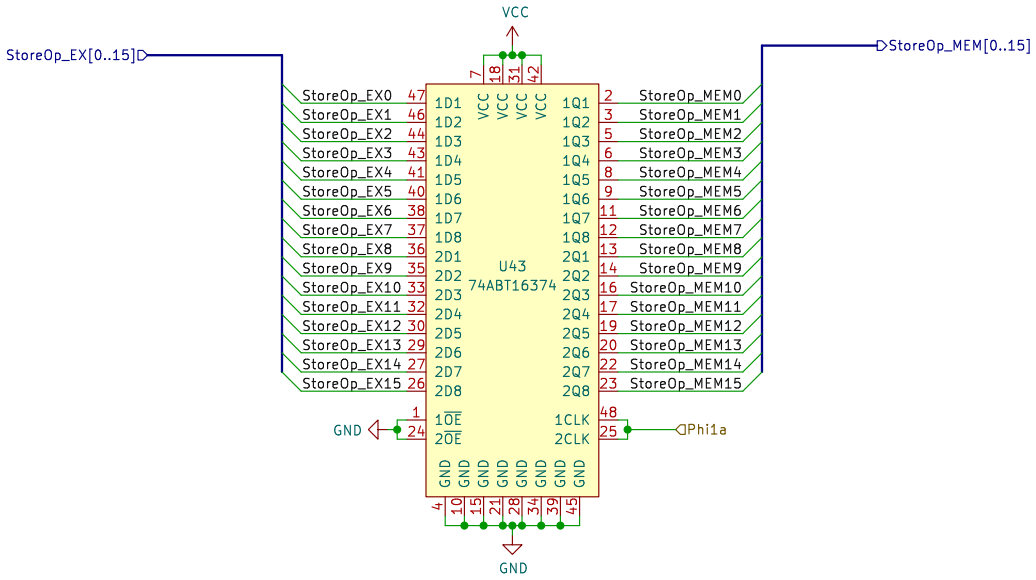




Sheet: /EX/sheet5FD8D6EB/
File: Ctl_13_23_Register.kicad_sch

**Title: EX/MEM: Control Word Register**

| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |

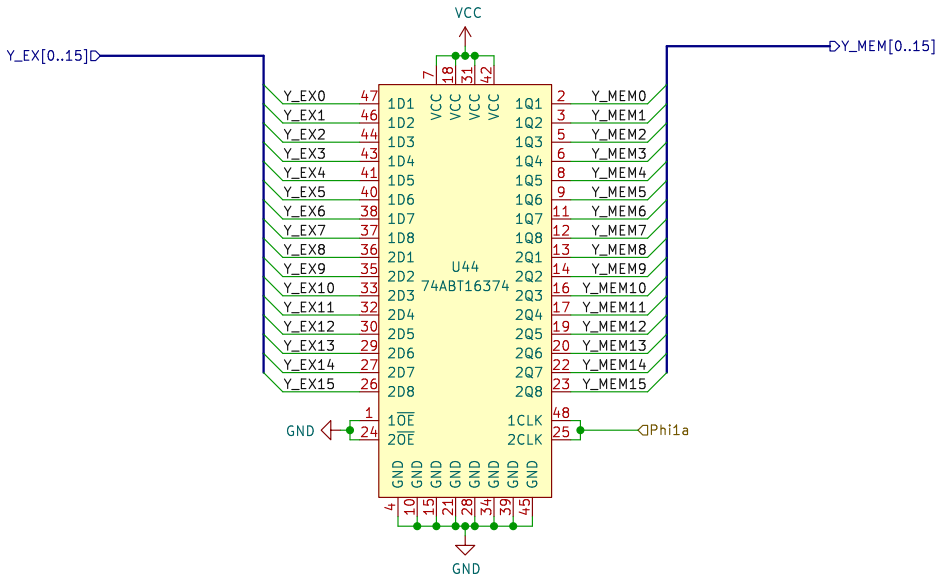




Sheet: /EX/sheet5FD8D6EA/
File: StoreOperandRegister2.kicad_sch

**Title: EX/MEM: Store Operand Register**

| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |

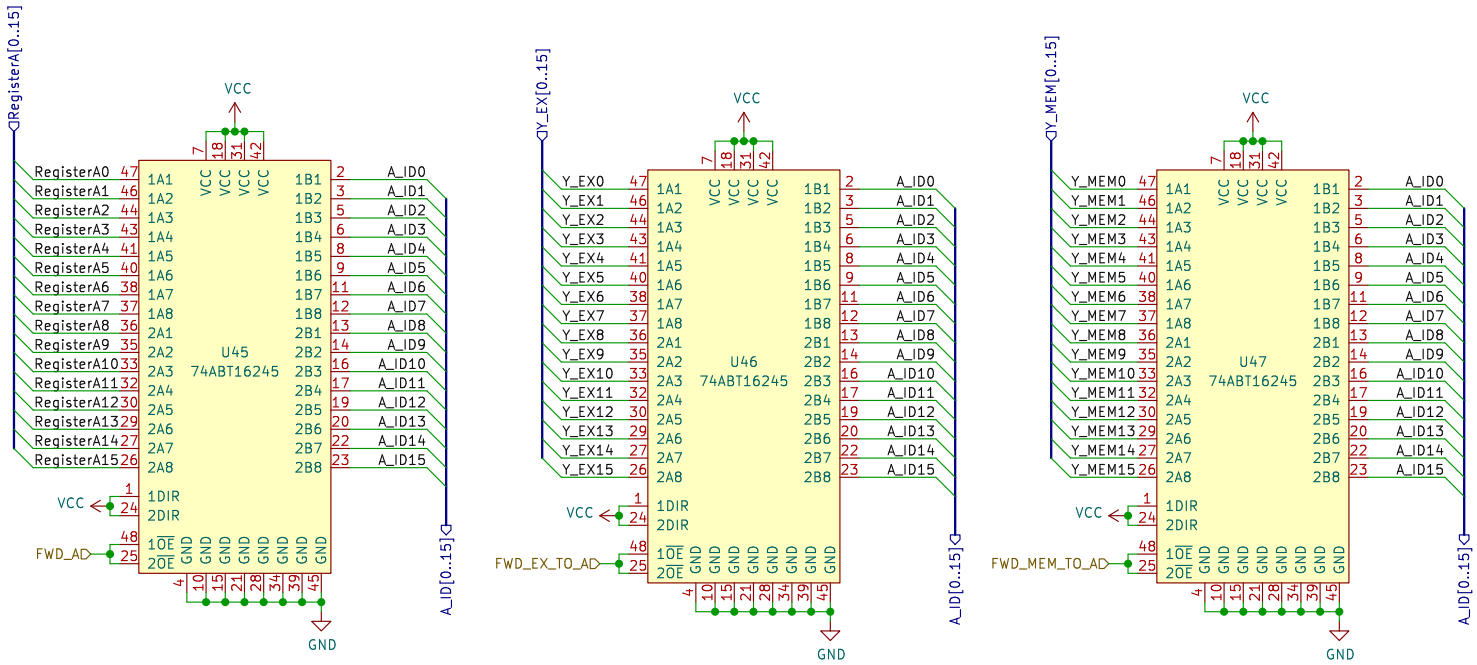


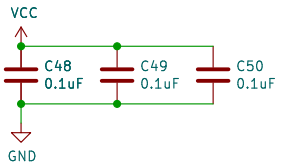


Sheet: /EX/sheet5FD8D6EC/
File: ALUResultRegister.kicad_sch

**Title: EX/MEM: ALUResult Register**

| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |

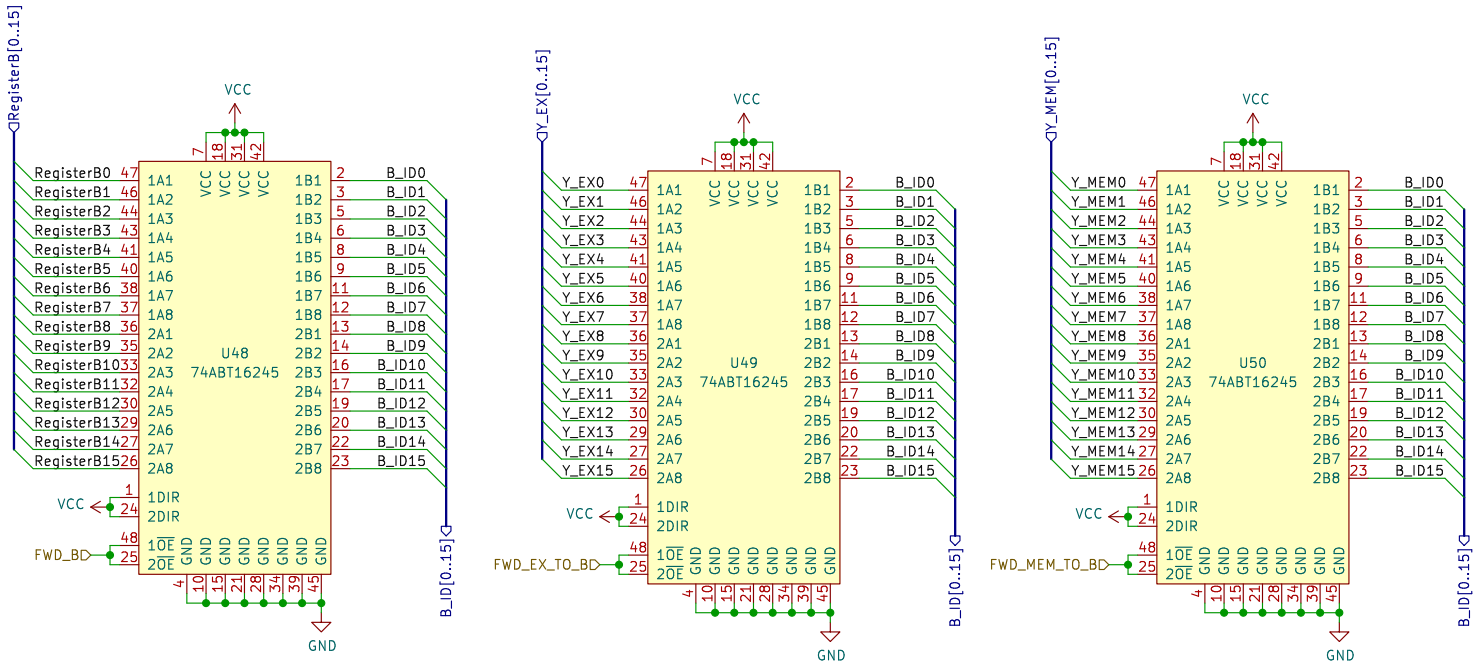


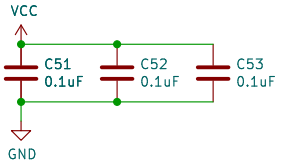




Sheet: /Operand Forwarding A/
File: OperandForwardingA.kicad_sch

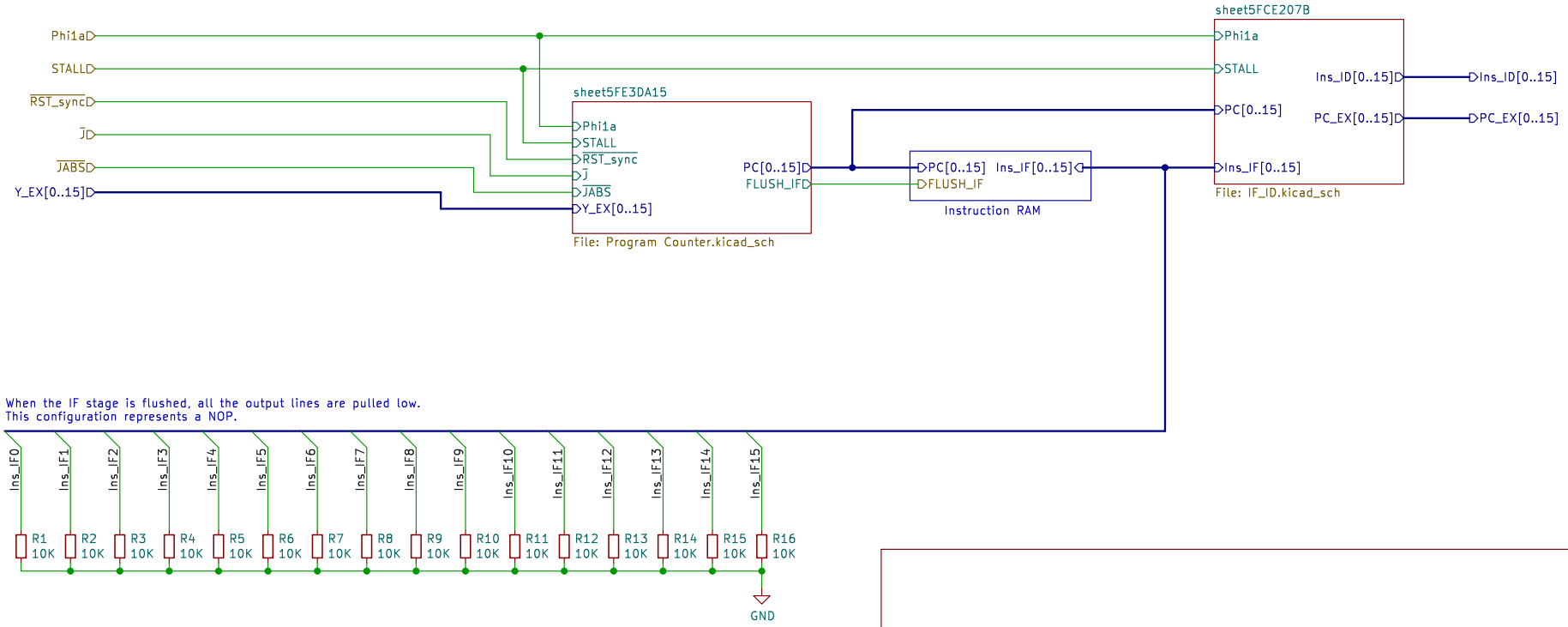
**Title:**

| Size: A4 | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |




| | | |
|---|---|---|
| Sheet: /Operand Forwarding B/ | | |
| File: OperandForwardingB.kicad_sch | | |
| **Title:** | | |
| Size: A4 | Date: 2023-11-25 | **Rev: A** |
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |




Sheet: /IF/
File: IF.kicad_sch

**Title: IF**

| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |

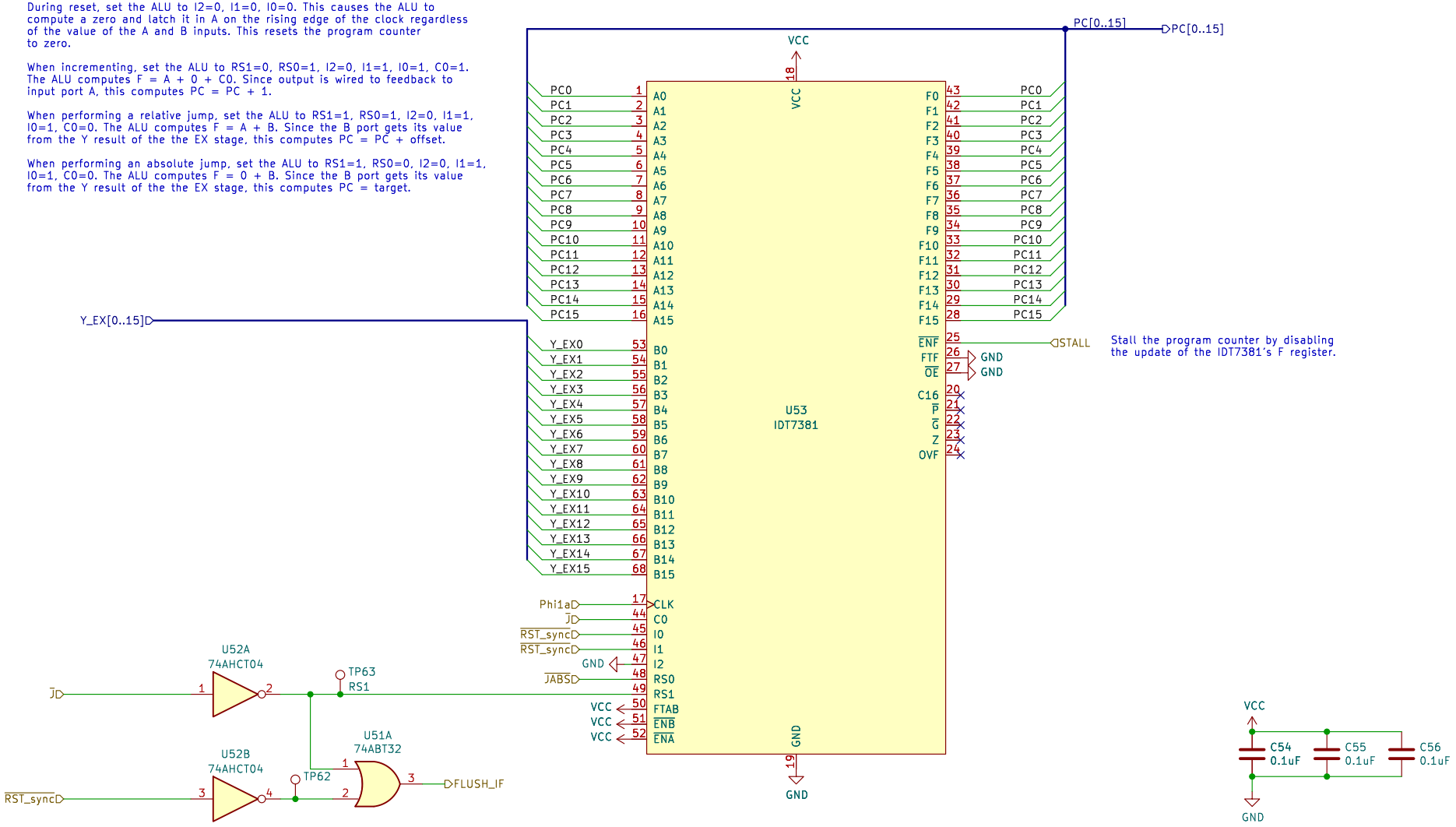
Configure the ALU for FTAB=1 and FTF=0. This causes the A and B registers to by bypassed entirely. The F output updates on the next rising edge of the clock.

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.



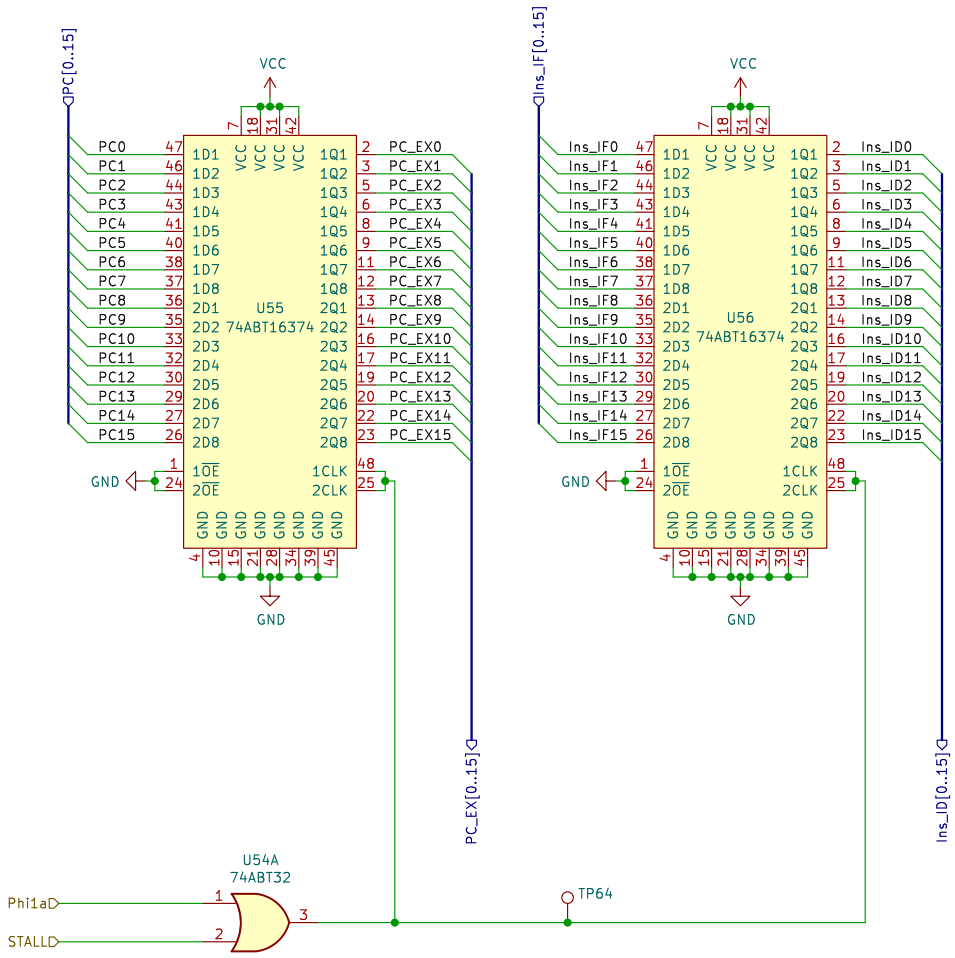

Flush the IF stage on Jump and on Reset. The flush during Reset ensures the pipeline is filled with NOPs.

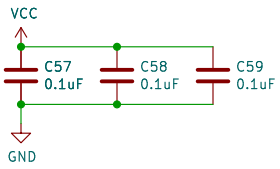
Sixteen-bit program counter will either increment on the clock, add a specified sixteen-bit offset, or else reset to zero.

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

**Title: Program Counter**

| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |



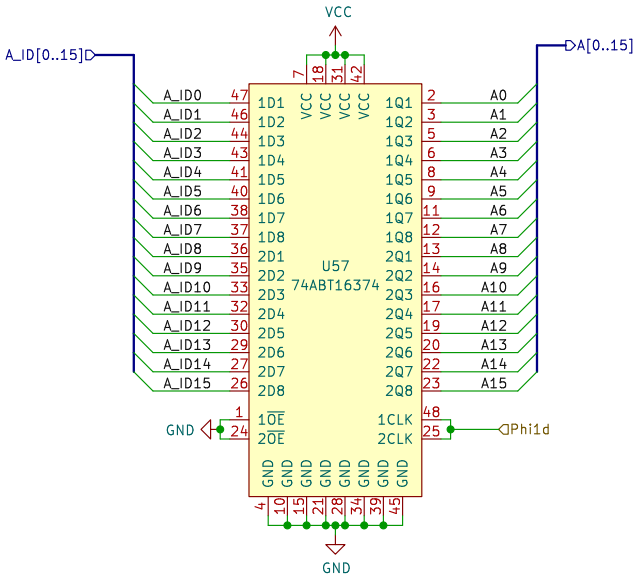




Interstage pipeline registers between IF and ID

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

**Title: IF/ID**

| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |

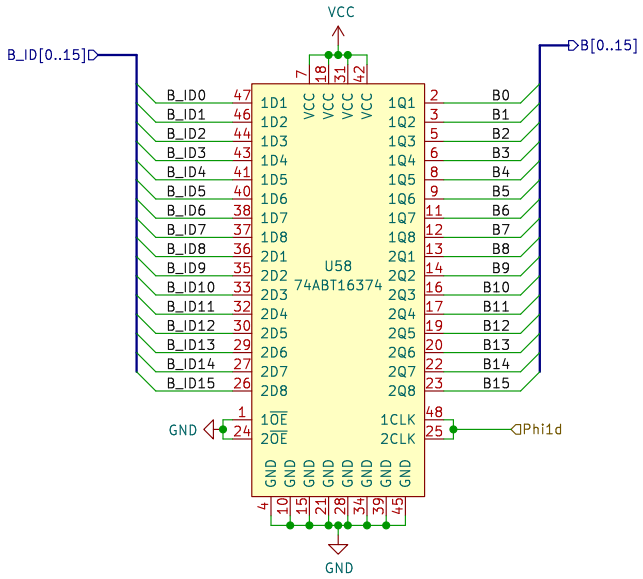




Sheet: /sheet60693B9A/
File: ID_EX_A.kicad_sch

**Title:**

| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |

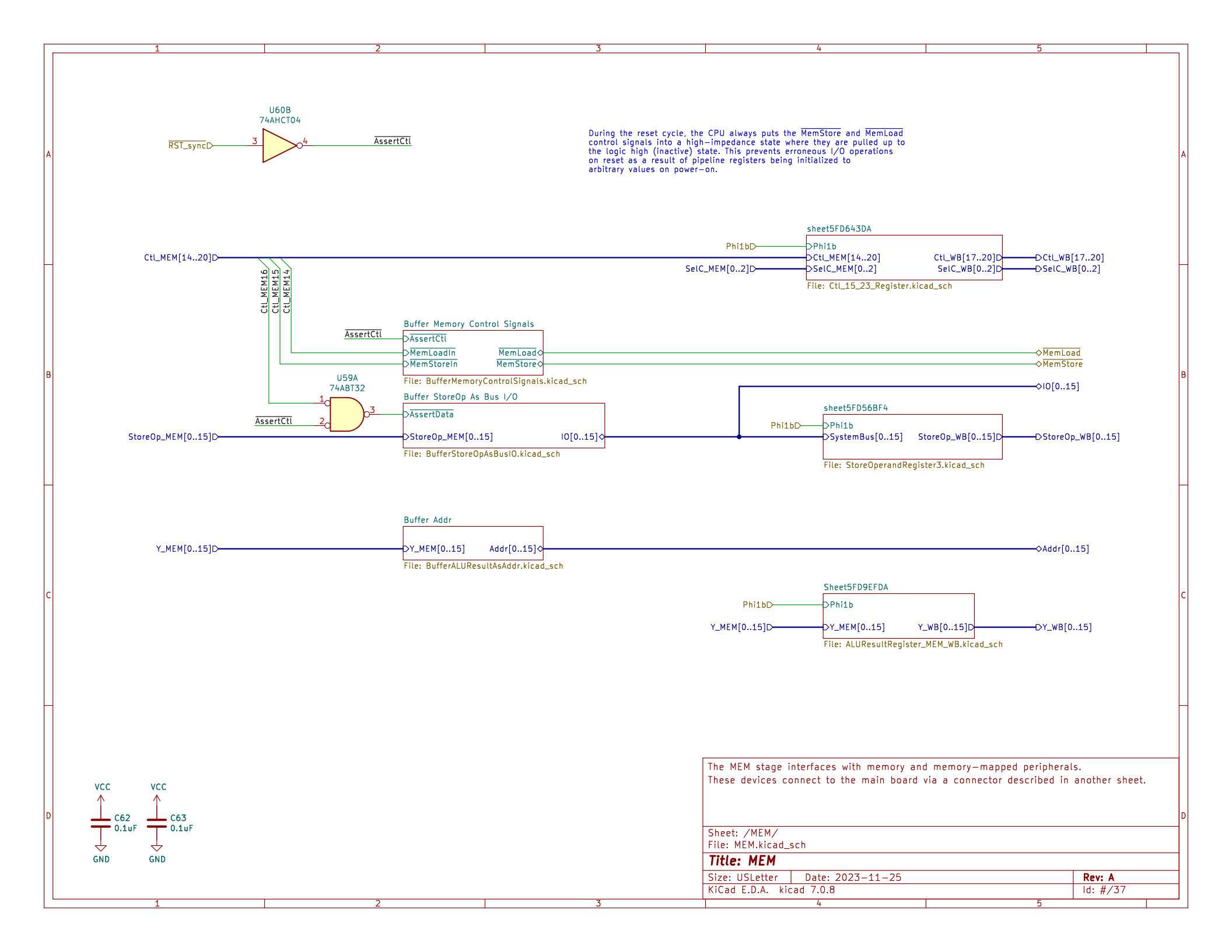




Sheet: /sheet60693B9B/
File: ID_EX_B.kicad_sch

**Title:**

| | | |
|---|---|---|
| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |


U60B
74AHCT04
RST_sync
AssertCtl
During the reset cycle, the CPU always puts the MemStore and MemLoad control signals into a high-impedance state where they are pulled up to the logic high (inactive) state. This prevents erroneous I/O operations on reset as a result of pipeline registers being initialized to arbitrary values on power-on.
sheet5FD643DA
Phi1b
Ctl_MEM[14..20]
SelC_MEM[0..2]
Ctl_WB[17..20]
SelC_WB[0..2]
File: Ctl_15_23_Register.kicad_sch
Ctl_MEM16
Ctl_MEM15
Ctl_MEM14
Buffer Memory Control Signals
AssertCtl
MemLoadIn
MemStoreIn
MemLoad
MemStore
File: BufferMemoryControlSignals.kicad_sch
U59A
74ABT32
Buffer StoreOp As Bus I/O
AssertData
StoreOp_MEM[0..15]
IO[0..15]
File: BufferStoreOpAsBusIO.kicad_sch
sheet5FD56BF4
SystemBus[0..15]
StoreOp_WB[0..15]
File: StoreOperandRegister3.kicad_sch
Buffer Addr
Y_MEM[0..15]
Addr[0..15]
File: BufferALUResultAsAddr.kicad_sch
Sheet5FD9EFDA
Y_WB[0..15]
File: ALUResultRegister_MEM_WB.kicad_sch
VCC
C62
0.1uF
C63
GND
The MEM stage interfaces with memory and memory-mapped peripherals.
These devices connect to the main board via a connector described in another sheet.
Sheet: /MEM/
File: MEM.kicad_sch
Title: MEM
Size: USLetter
Date: 2023-11-25
Rev: A
KiCad E.D.A. kicad 7.0.8
Id: #/37

IO[0..15]
Addr[0..15]
Ins_IF[0..15]
PC[0..15]
Bank[0..2]

The Bank register resets to zero when the computer resets. Set the bank register by writing a word to address 0xffff.

When the bank register is zero, the instruction fetch unit is fed by ROM.

When the bank register is not zero, the instruction fetch unit is fed by The right side of RAM. This allows the computer to be reprogrammed at run time.

When the bank register is zero or one then the left side of RAM is enabled on the system bus.

The PLD allows some other combinations of memory address and bank to be mapped to RAM during load/store over the memory bus. This mapping may be reprogrammed later to accomodate new devices added to the bus. If there are no other devices on the bus then a good default mapping would be to map all banks and memory ranges to RAM.

ROM
PC[0..15]
FLUSH_IF
RAMEN_R
PC[0..15]
FLUSH_IF
RAMEN_R
Ins_IF[0..15]
Ins_IF[0..15]
File: ROM.kicad_sch

RAM
IO[0..15]
Addr[0..15]
PC[0..15]
FLUSH_IF
MemStore
MemLoad
RAMEN_L
RAMEN_L
RAMEN_R
RAMEN_R
IO[0..15]
Addr[0..15]
PC[0..15]
FLUSH_IF
MemStore
MemLoad
RAMEN_L
RAMEN_L
RAMEN_R
RAMEN_R
Ins_IF[0..15]
Ins_IF[0..15]
File: RAM.kicad_sch

VCC
C65
0.1uF
GND

U61
ATF22V10C-7PX
VCC
GND
Addr0 1 IN/CLK
Addr1 2 IN
Addr2 3 IN
Addr3 4 IN
Addr4 5 IN
Addr5 6 IN
Addr6 7 IN
Addr7 8 IN
Addr8 9 IN
Addr9 10 IN
Addr10 11 IN
24 VCC
12 GND
23 I/O RAMEN_L
22 I/O RAMEN_R
21 I/O
20 I/O Bank2
19 I/O Bank1
18 I/O Bank0
17 I/O Addr15
16 I/O Addr14
15 I/O Addr13
14 I/O Addr12
13 I/O Addr11 IN/OE

VCC
C64
0.1uF
GND

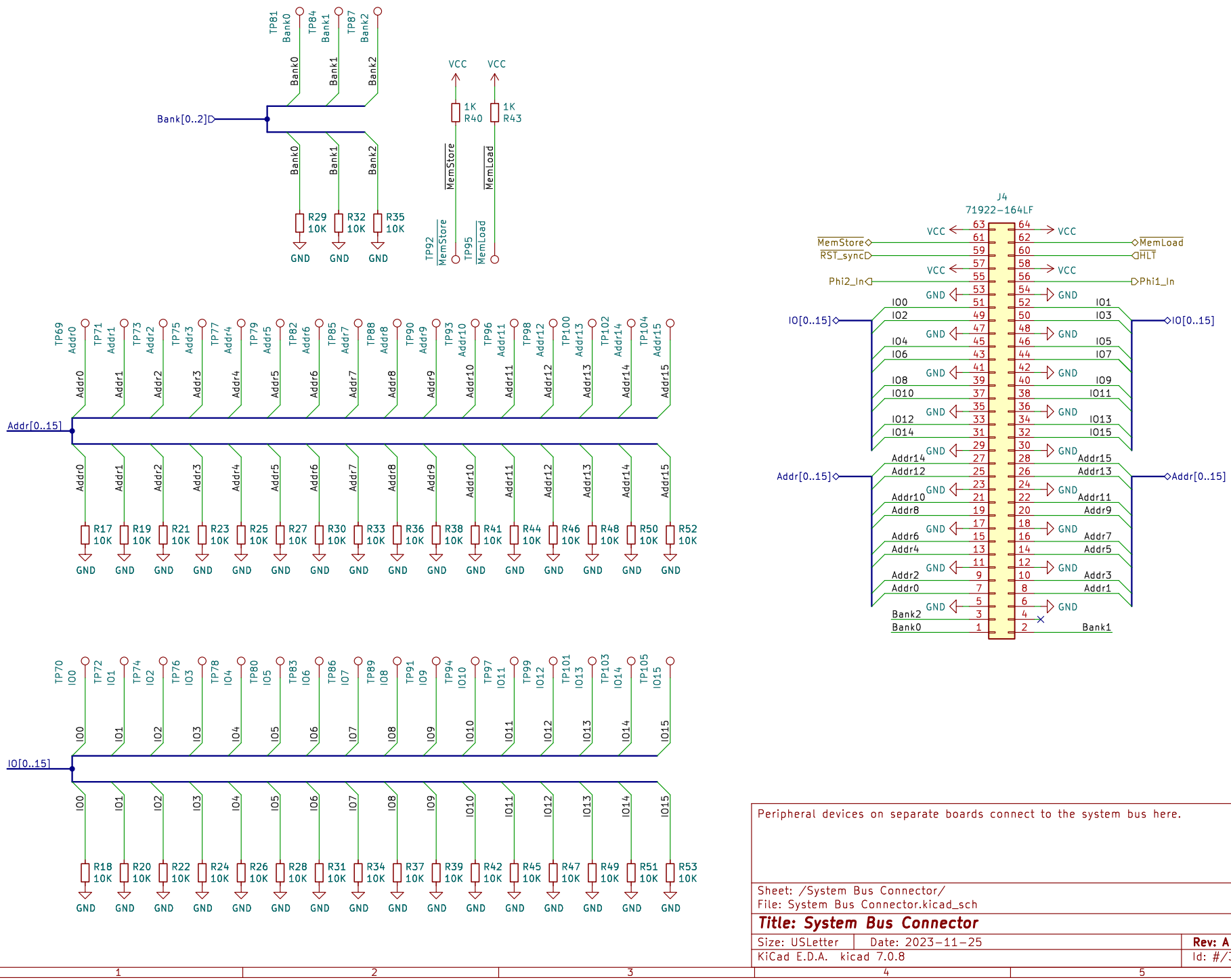
U62
CD74HC173PWR
VCC
GND
IO0 14 D0
IO1 13 D1
IO2 12 D2
IO3 11 D3
16 VCC
8 GND
3 Q0 Bank0
4 Q1 Bank1
5 Q2 Bank2
6 Q3
1 Oe1
2 Oe2
GND
9 E1
10 E2
7 Cp
MemStore
Phi1b
15 Mr
U60E
74AHCT04
11
10
RST_sync

U60C
74AHCT04
RAMEN_L 5 6 RAMEN_L

U60D
74AHCT04
RAMEN_R 9 8 RAMEN_R

TP65
TestPoint TP66
TestPoint TP67
TestPoint TP68
TestPoint
RAMEN_L
RAMEN_L
RAMEN_R
RAMEN_R

Sheet: /Memory/
File: Memory.kicad_sch
Title:
Size: A4 | Date: 2023-11-25 | Rev: A
KiCad E.D.A. kicad 7.0.8 | Id: #/37




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

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

**Title: System Bus Connector**

| Size: USLetter | Date: 2023-11-25 | **Rev: A** |
|---|---|---|
| KiCad E.D.A. kicad 7.0.8 | | Id: #/37 |