Sheet: Output Display

File: Output Display.sch

Sheet: Register XY

File: Register XY.sch

Sheet: Speed Control

File: Speed Control.sch

Sheet: Program Counter

File: Program Counter.sch

Sheet: Clock

File: Clock.sch

Sheet: Instruction Fetch

File: Instruction Fetch.sch

Sheet: Power-on Reset

File: Power-on Reset.sch

Sheet: Data RAM

File: Data RAM.sch

Sheet: Instruction Decode

File: Instruction Decode.sch

Sheet: Register A

File: Register A.sch

Sheet: Serial

File: Serial.sch

Sheet: Execute

File: Execute.sch

Sheet: Register B

File: Register B.sch

Sheet: Register D

File: Register D.sch

Sheet: Bus Display

File: Bus Display.sch

Sheet: ALU

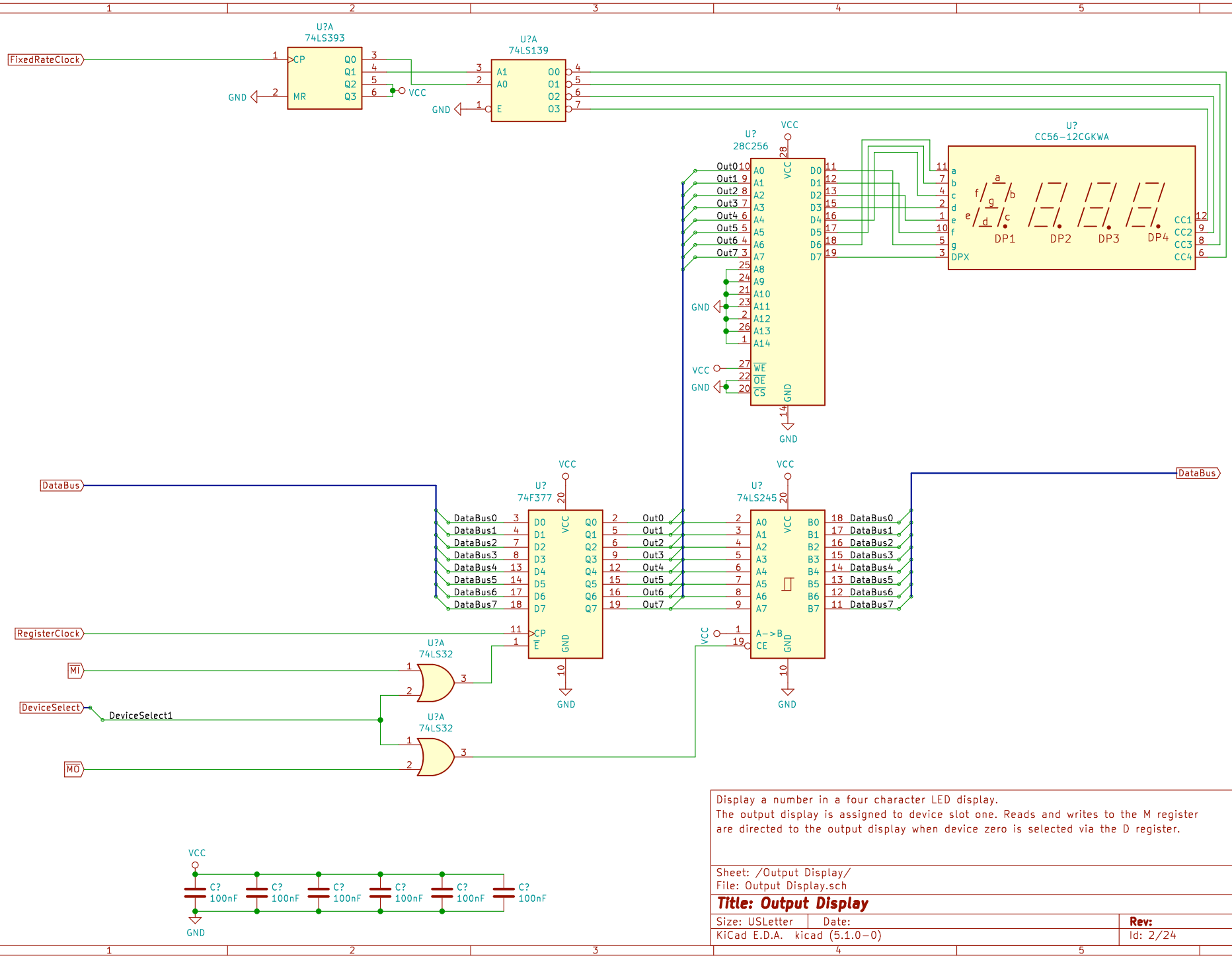
File: ALU.sch

TTL microcomputer built from 74xx series logic chips.

Sheet: /
File: TurtleTTL.sch

Title: Turtle TTL

Size: A	Date:	**Rev:**
KiCad E.D.A. kicad (5.1.0-0)		Id: 1/24

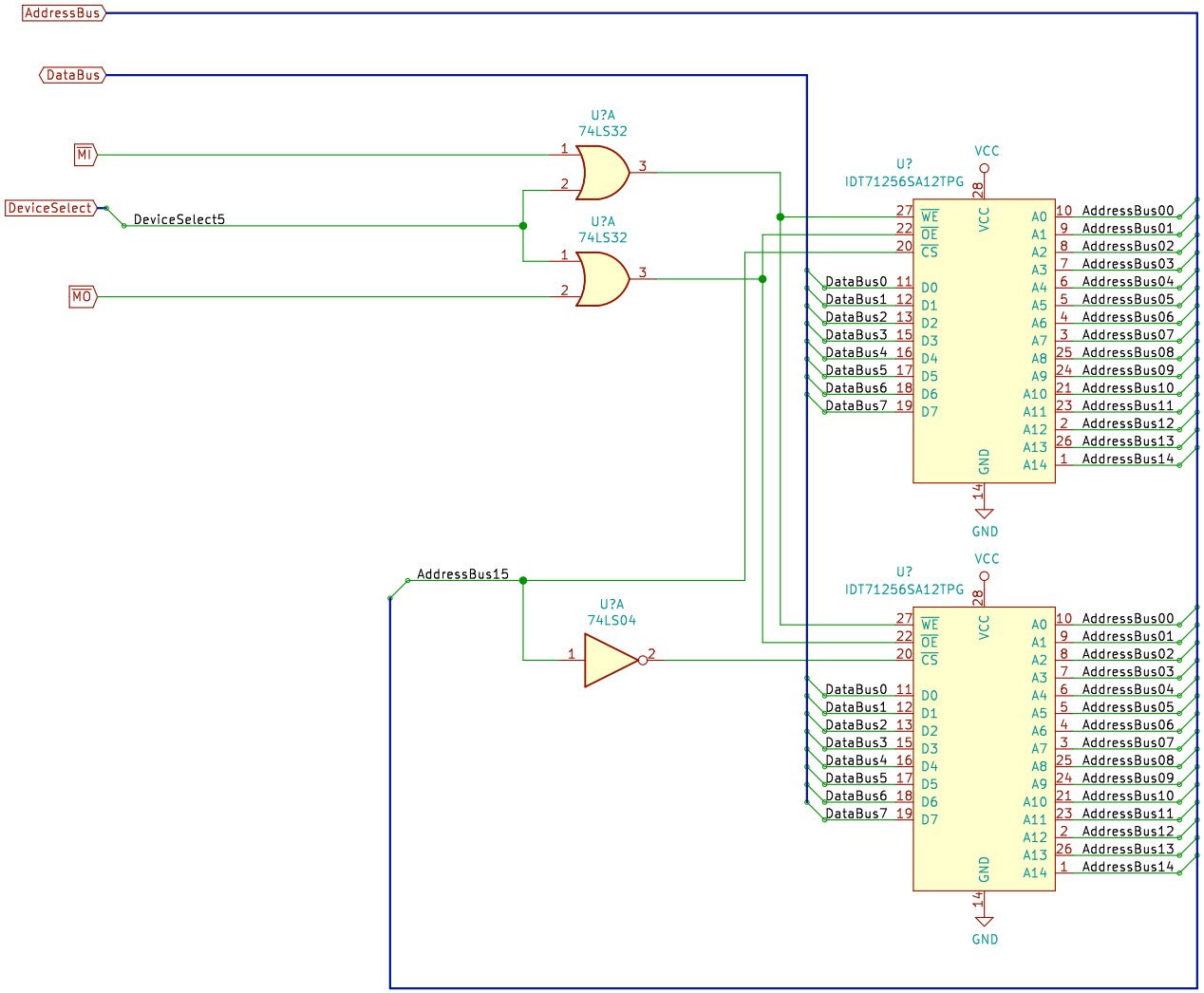


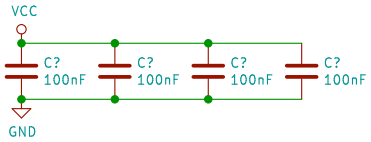
Display a number in a four character LED display.
The output display is assigned to device slot one. Reads and writes to the M register are directed to the output display when device zero is selected via the D register.

Sheet: /Output Display/
File: Output Display.sch

Title: Output Display

Size: USLetter	Date:	Rev:
KiCad E.D.A. kicad (5.1.0-0)		Id: 2/24



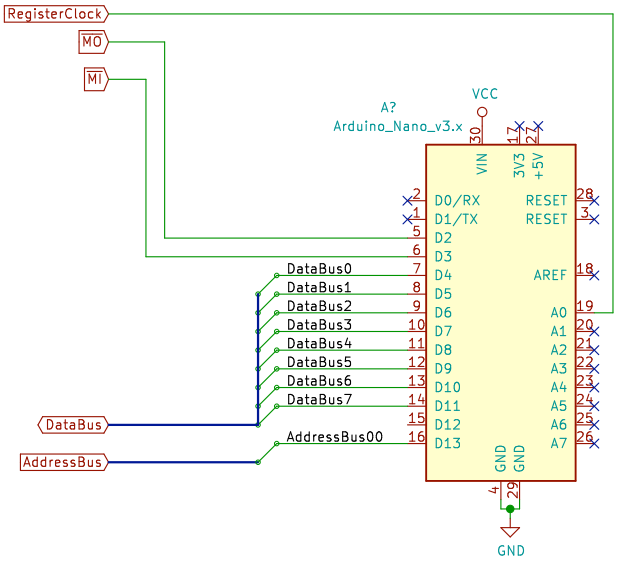


64KB Data RAM is connected to device slot five.
Reads and writes to the M register are directed to Data RAM when device 5 is selected via the D register.
The MSB of the address selects between two 32KB SRAM ICs.

Sheet: /Data RAM/
File: Data RAM.sch

Title: Data RAM

Size: USLetter	Date:	**Rev:**
KiCad E.D.A. kicad (5.1.0-0)		Id: 3/24

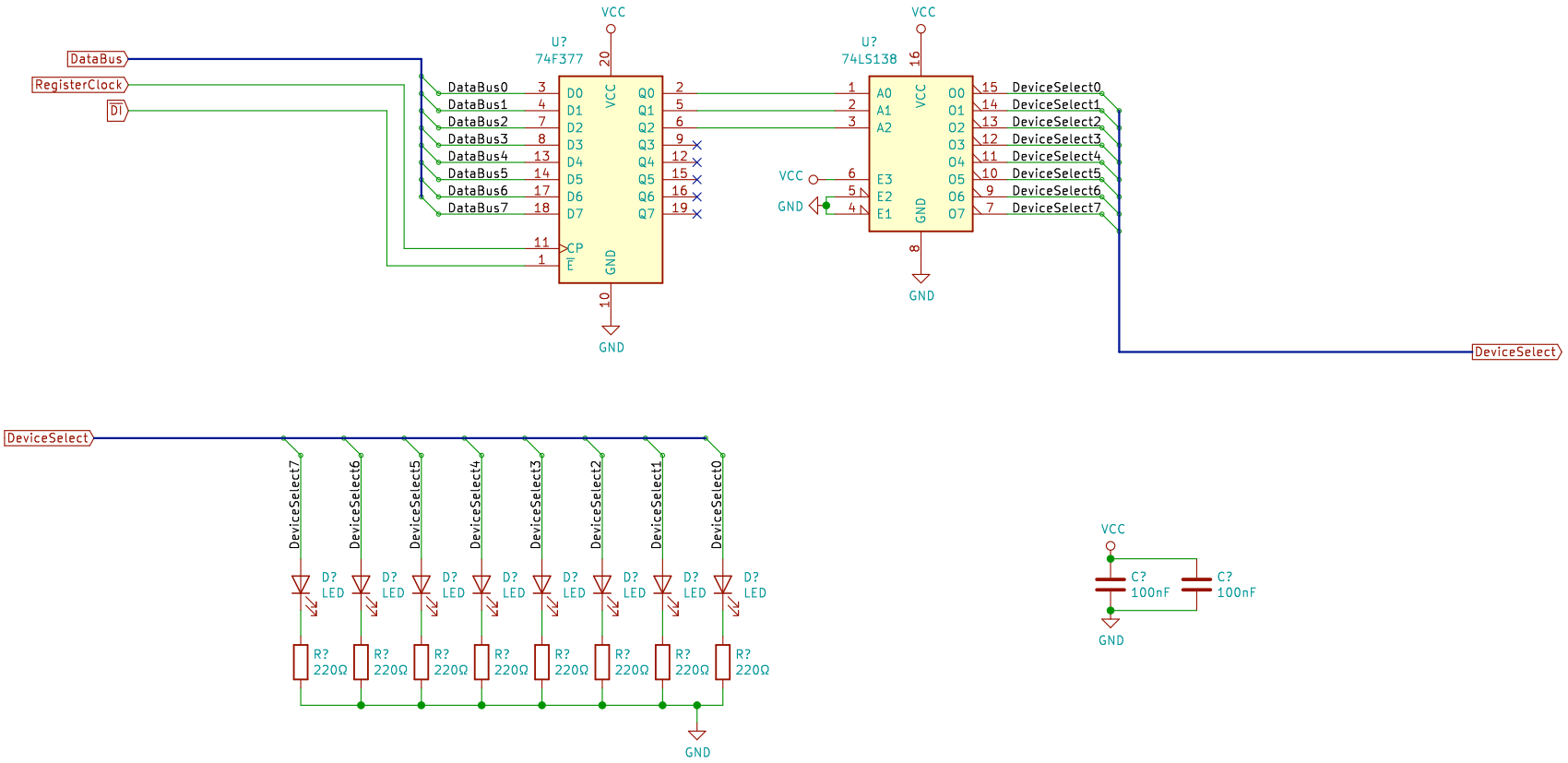


An Arduino Nano serves as a serial interface module.

Sheet: /Serial/
File: Serial.sch

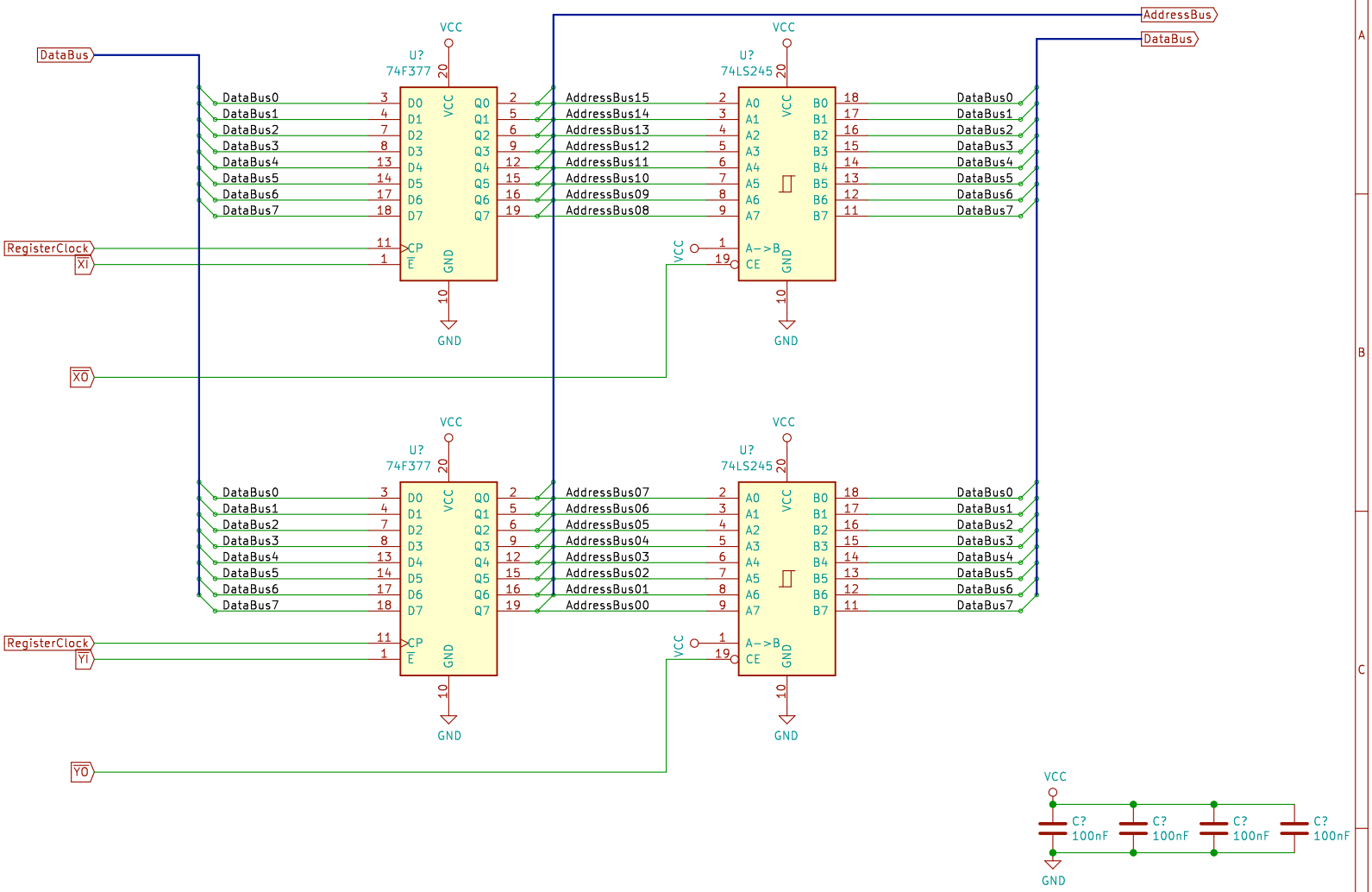
Title:

Size: USLetter	Date:	**Rev:**
KiCad E.D.A. kicad (5.1.0-0)		Id: 4/24



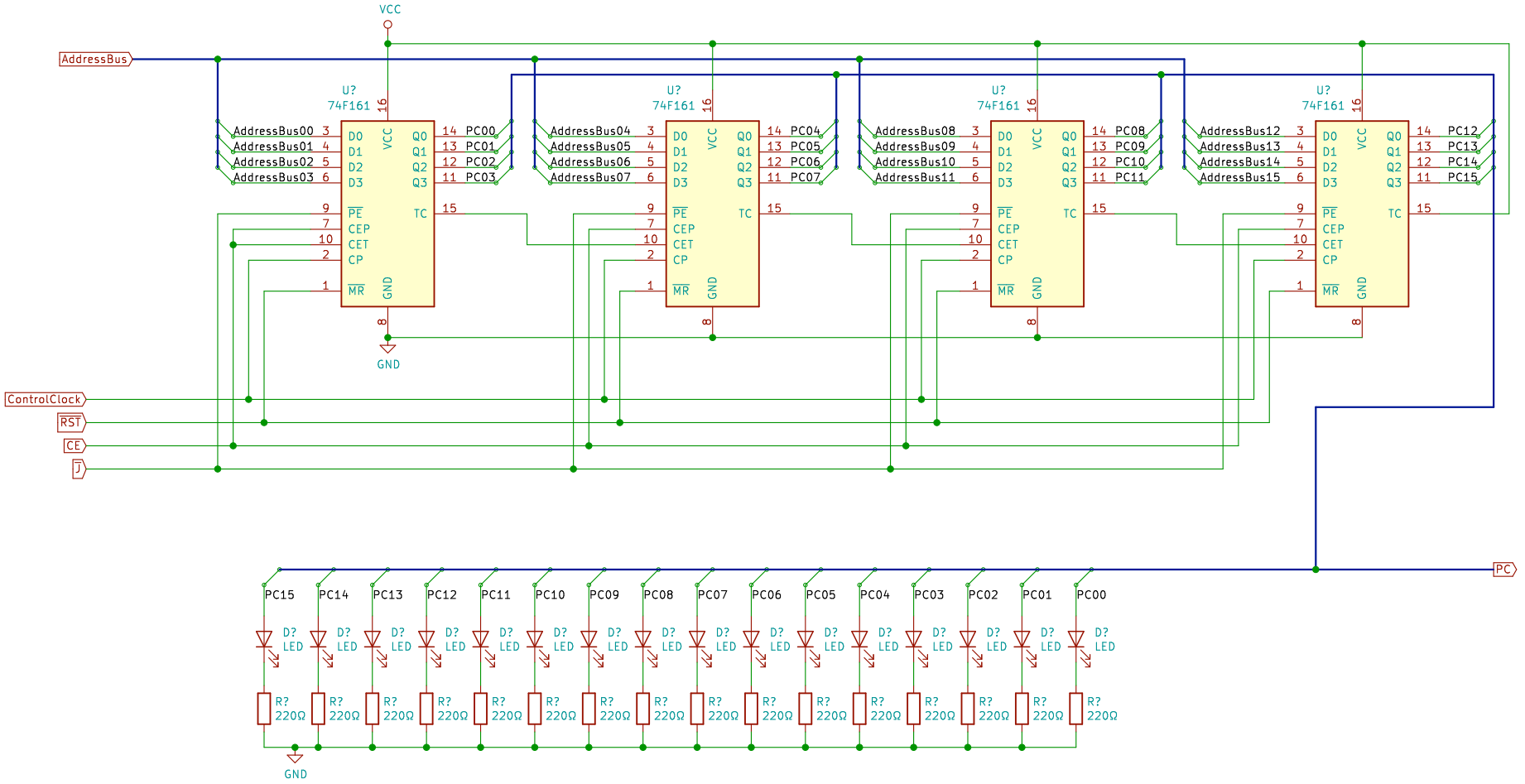
Register D controls the device select lines for peripherals and memory.

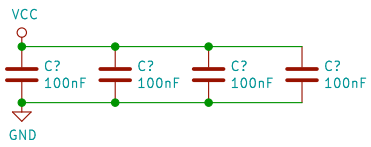
Sheet: /Register D/	
File: Register D.sch	
Title: Register D	
Size: A	Date:
KiCad E.D.A. kicad (5.1.0-0)	Id: 5/24



The XY register pair sets the value on the sixteen-bit Address Bus.

Sheet: /Register XY/		
File: Register XY.sch		
Title:		
Size: USLetter	Date:	**Rev:**
KiCad E.D.A. kicad (5.1.0-0)		Id: 6/24





Sheet: /Program Counter/	
File: Program Counter.sch	
Title: Program Counter	
Size: USLetter	Date:
KiCad E.D.A. kicad (5.1.0-0)	Rev: Id: 7/24

Sheet: PC/IF

File: PC_IF.sch

Sheet: Instruction RAM Address

File: Instruction RAM Address.sch

Sheet: Instruction ROM

File: Instruction ROM.sch

Sheet: Instruction RAM

File: Instruction RAM.sch

Sheet: Instruction Register

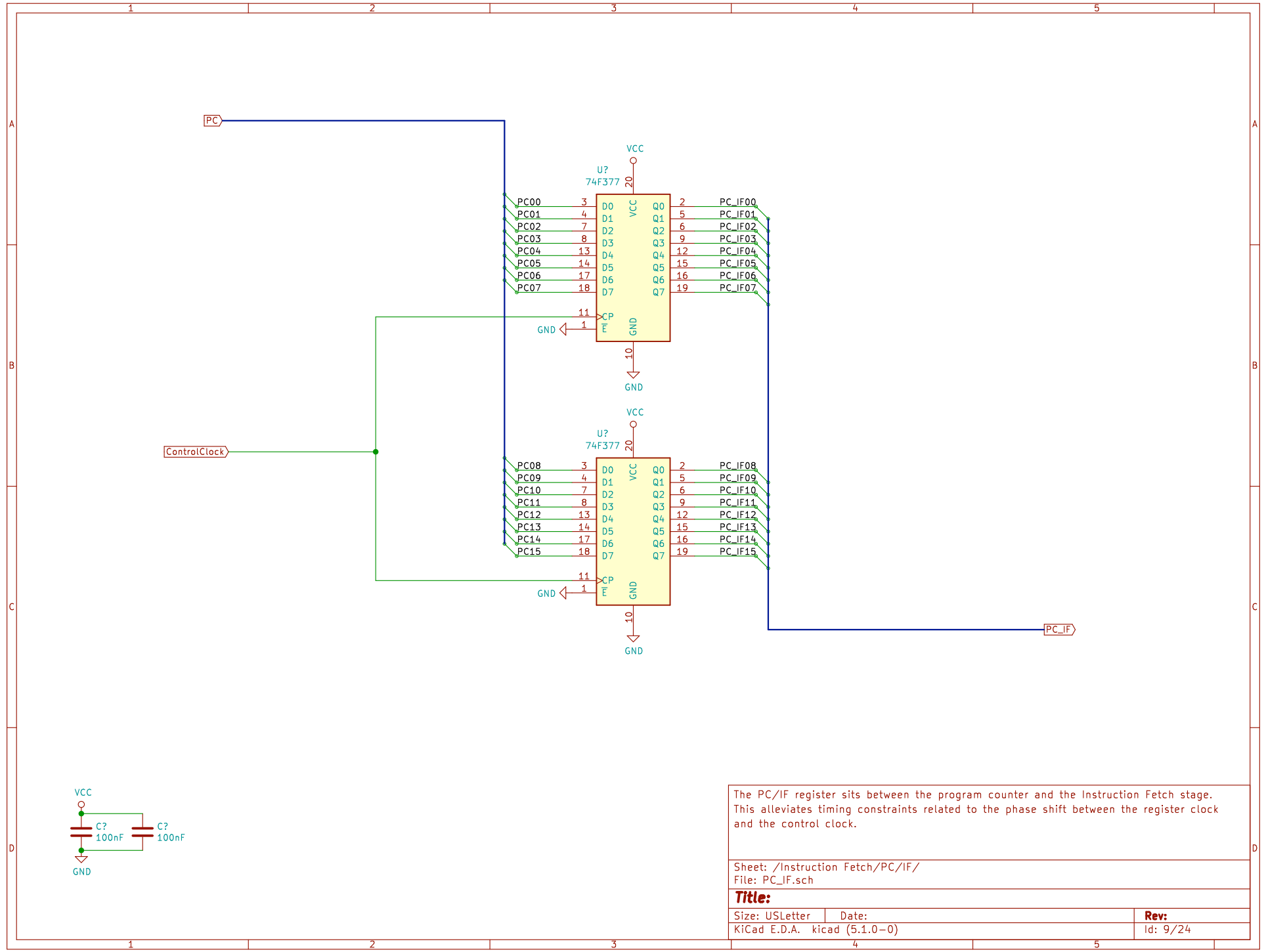
File: Instruction Register.sch

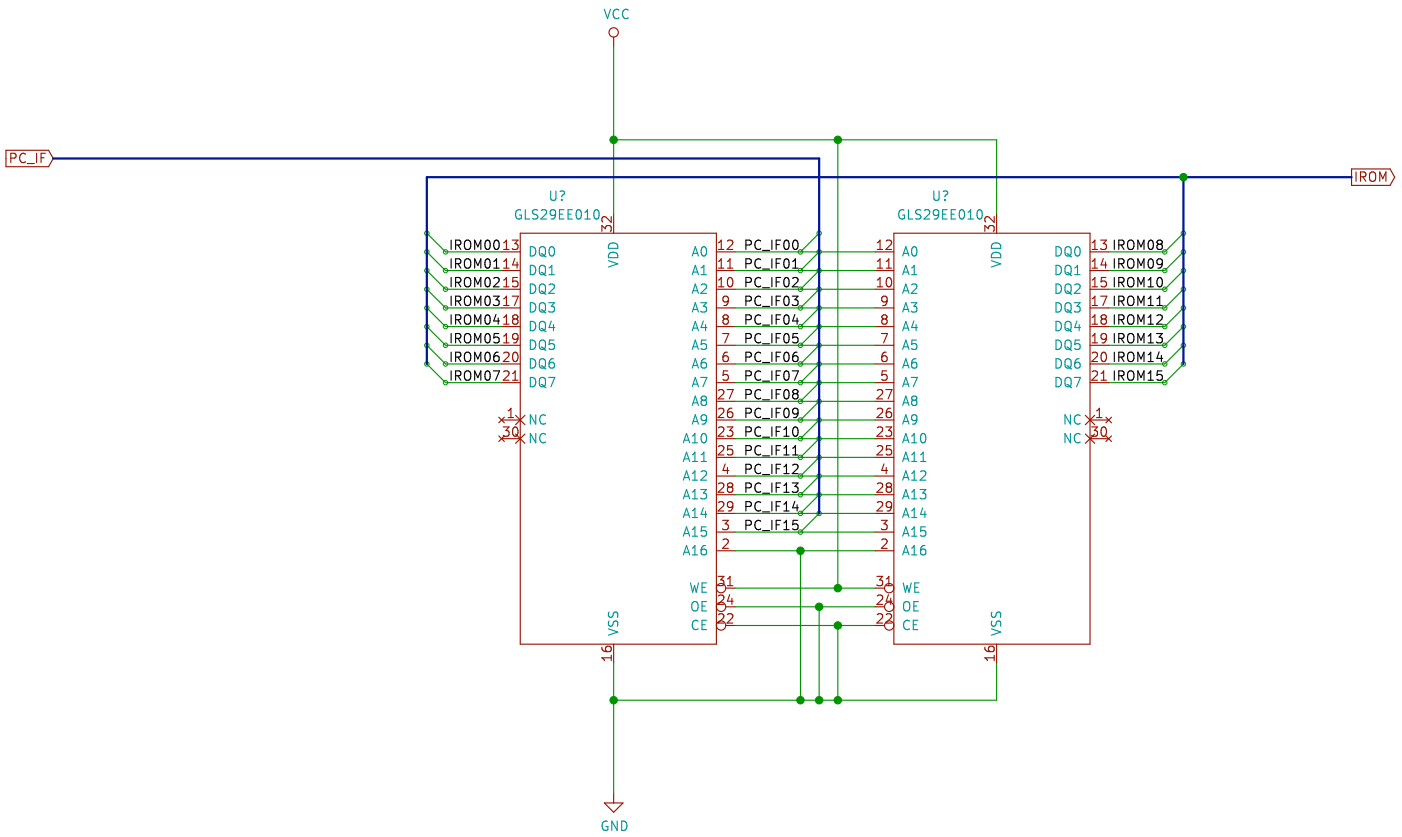
Instructions can be fetched from either Instruction ROM or Instruction RAM.
The lower 32KB of the address space is mapped to ROM, the remainder to RAM.

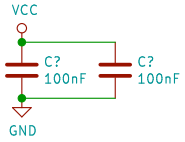
Sheet: /Instruction Fetch/
File: Instruction Fetch.sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. kicad (5.1.0-0)		Id: 8/24


PC
ControlClock
VCC
U?
74F377
20
PC00
PC01
PC02
PC03
PC04
PC05
PC06
PC07
3
4
7
8
13
14
17
18
D0
D1
D2
D3
D4
D5
D6
D7
VCC
CP
E
GND
11
1
10
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
2
5
6
9
12
15
16
19
PC_IF00
PC_IF01
PC_IF02
PC_IF03
PC_IF04
PC_IF05
PC_IF06
PC_IF07
PC08
PC09
PC10
PC11
PC12
PC13
PC14
PC15
PC_IF08
PC_IF09
PC_IF10
PC_IF11
PC_IF12
PC_IF13
PC_IF14
PC_IF15
PC_IF
C?
100nF
The PC/IF register sits between the program counter and the Instruction Fetch stage.
This alleviates timing constraints related to the phase shift between the register clock
and the control clock.
Sheet: /Instruction Fetch/PC/IF/
File: PC_IF.sch
Title:
Size: USLetter
Date:
Rev:
KiCad E.D.A. kicad (5.1.0-0)
Id: 9/24



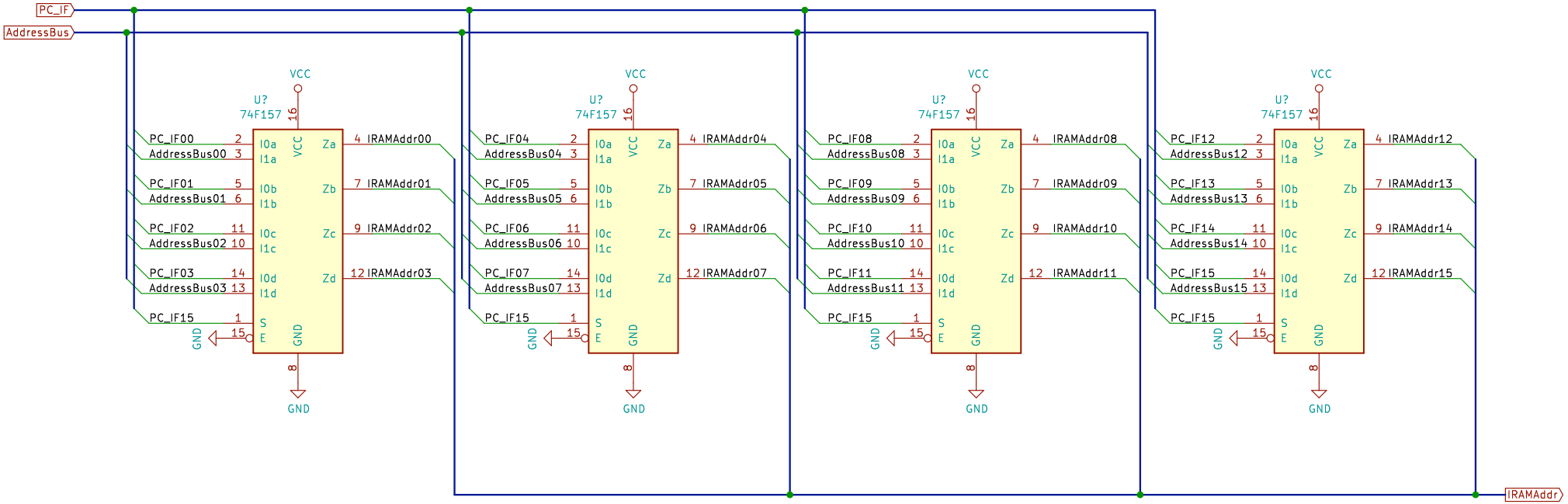


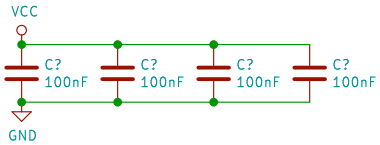
16-bit instructions are read from ROM at the address specified by the Program Counter.

Sheet: /Instruction Fetch/Instruction ROM/
File: Instruction ROM.sch

Title: Instruction ROM

Size: USLetter	Date:	**Rev:**
KiCad E.D.A. kicad (5.1.0-0)		Id: 10/24



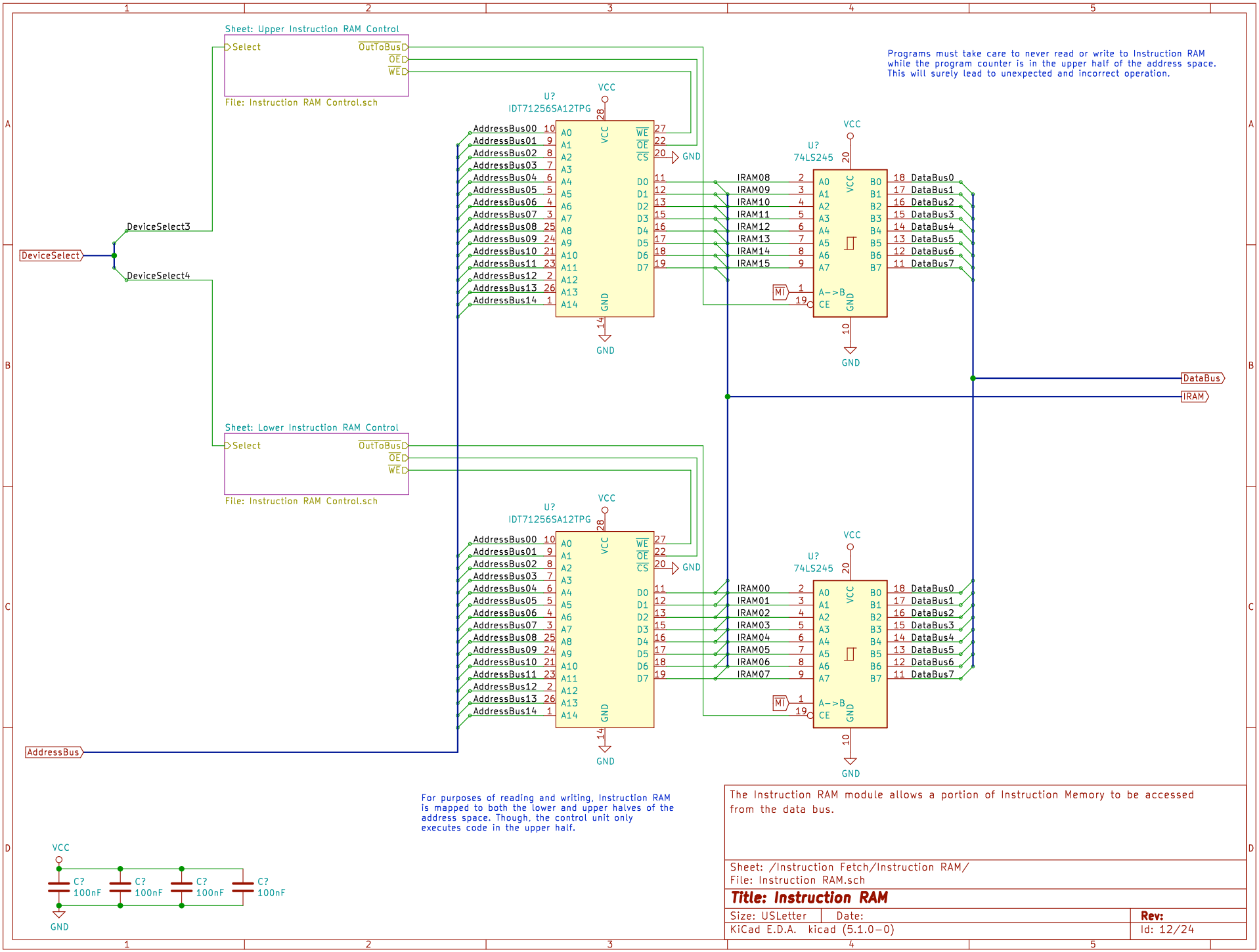


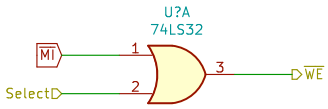
Instruction RAM uses an address from either the program counter or the address bus.
If the program counter is in the upper 32KB of the address space then use the program counter as we're running in Instruction RAM.

Sheet: /Instruction Fetch/Instruction RAM Address/
File: Instruction RAM Address.sch

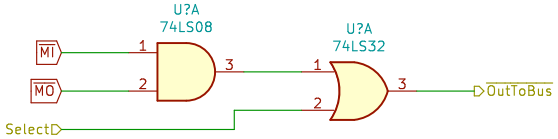
Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. kicad (5.1.0-0)		Id: 11/24


Sheet: Upper Instruction RAM Control
Select
OutToBus
OE
WE
File: Instruction RAM Control.sch
Programs must take care to never read or write to Instruction RAM while the program counter is in the upper half of the address space. This will surely lead to unexpected and incorrect operation.
DeviceSelect
DeviceSelect3
DeviceSelect4
U?
IDT71256SA12TPG
VCC
GND
AddressBus00
AddressBus01
AddressBus02
AddressBus03
AddressBus04
AddressBus05
AddressBus06
AddressBus07
AddressBus08
AddressBus09
AddressBus10
AddressBus11
AddressBus12
AddressBus13
AddressBus14
U?
74LS245
IRAM08
IRAM09
IRAM10
IRAM11
IRAM12
IRAM13
IRAM14
IRAM15
DataBus0
DataBus1
DataBus2
DataBus3
DataBus4
DataBus5
DataBus6
DataBus7
MI
A->B
CE
DataBus
IRAM
Sheet: Lower Instruction RAM Control
Select
OutToBus
OE
WE
File: Instruction RAM Control.sch
U?
IDT71256SA12TPG
U?
74LS245
IRAM00
IRAM01
IRAM02
IRAM03
IRAM04
IRAM05
IRAM06
IRAM07
AddressBus
For purposes of reading and writing, Instruction RAM is mapped to both the lower and upper halves of the address space. Though, the control unit only executes code in the upper half.
C?
100nF
The Instruction RAM module allows a portion of Instruction Memory to be accessed from the data bus.
Sheet: /Instruction Fetch/Instruction RAM/
File: Instruction RAM.sch
Title: Instruction RAM
Size: USLetter
Date:
Rev:
KiCad E.D.A. kicad (5.1.0-0)
Id: 12/24

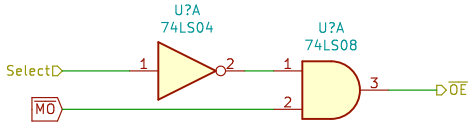


Only write to the Instruction SRAM when the MI signal is active and the appropriate plane of Instruction RAM is selected.



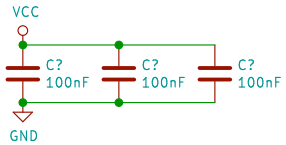
The bus transceiver should connect the Instruction SRAM to the Data Bus when the plane is selected and wither MO or MI is active.

The direction of the bus transceiver must also be set appropriately. This is done elsewhere.



Always output from Instruction SRAM when the plane of Instruction RAM is deselected. This allows RAM to be read for use in IF.

Else, if the plane is selected then only read from SRAM when the MO signal is active.

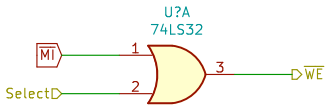


Logic for the control signals which drive Instruction RAM.

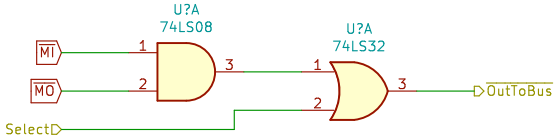
Sheet: /Instruction Fetch/Instruction RAM/Lower Instruction RAM Control/
File: Instruction RAM Control.sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. kicad (5.1.0-0)		Id: 13/24

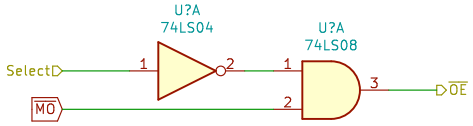


Only write to the Instruction SRAM when the MI signal is active and the appropriate plane of Instruction RAM is selected.



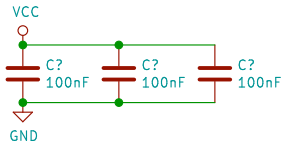
The bus transceiver should connect the Instruction SRAM to the Data Bus when the plane is selected and wither MO or MI is active.

The direction of the bus transceiver must also be set appropriately. This is done elsewhere.



Always output from Instruction SRAM when the plane of Instruction RAM is deselected. This allows RAM to be read for use in IF.

Else, if the plane is selected then only read from SRAM when the MO signal is active.

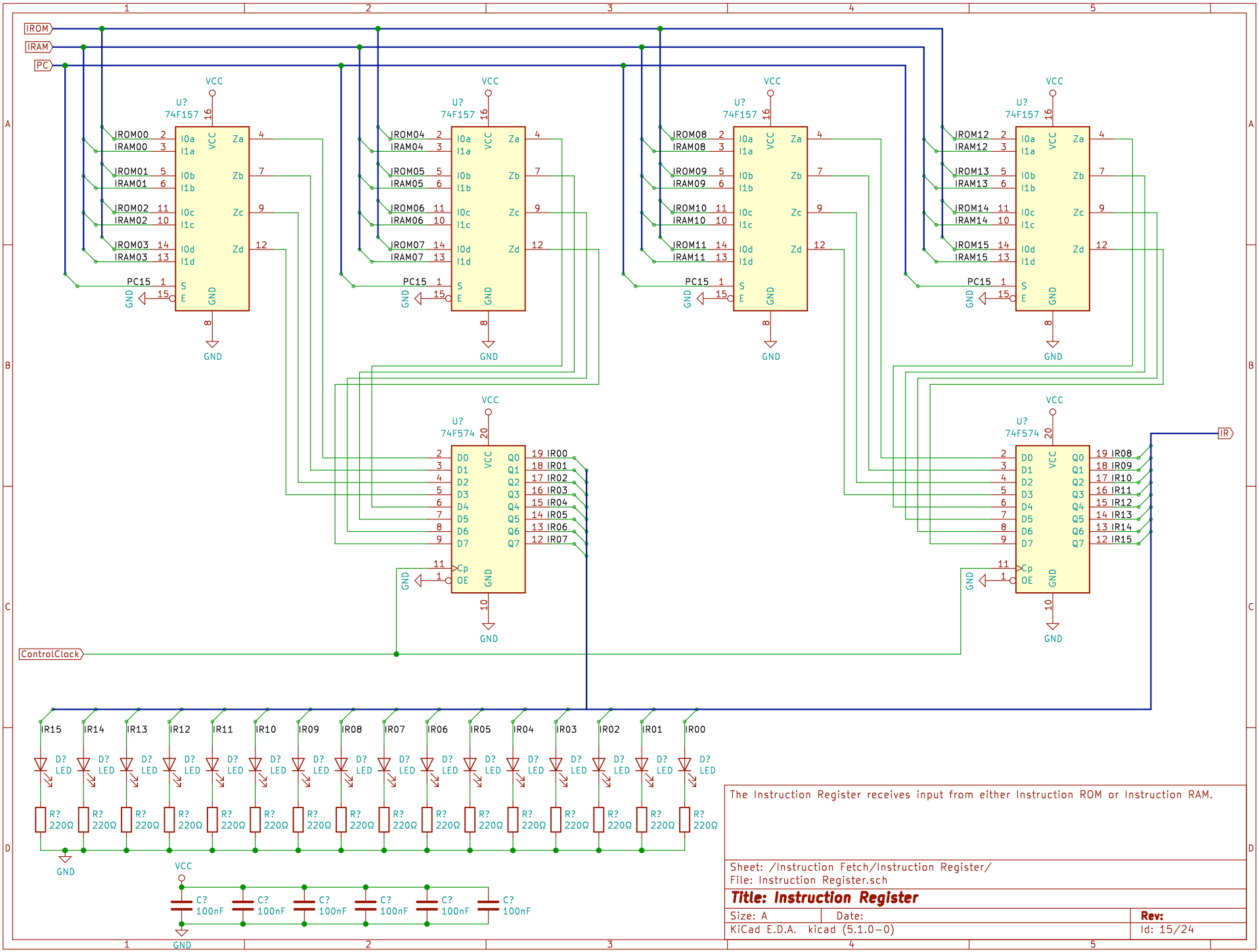


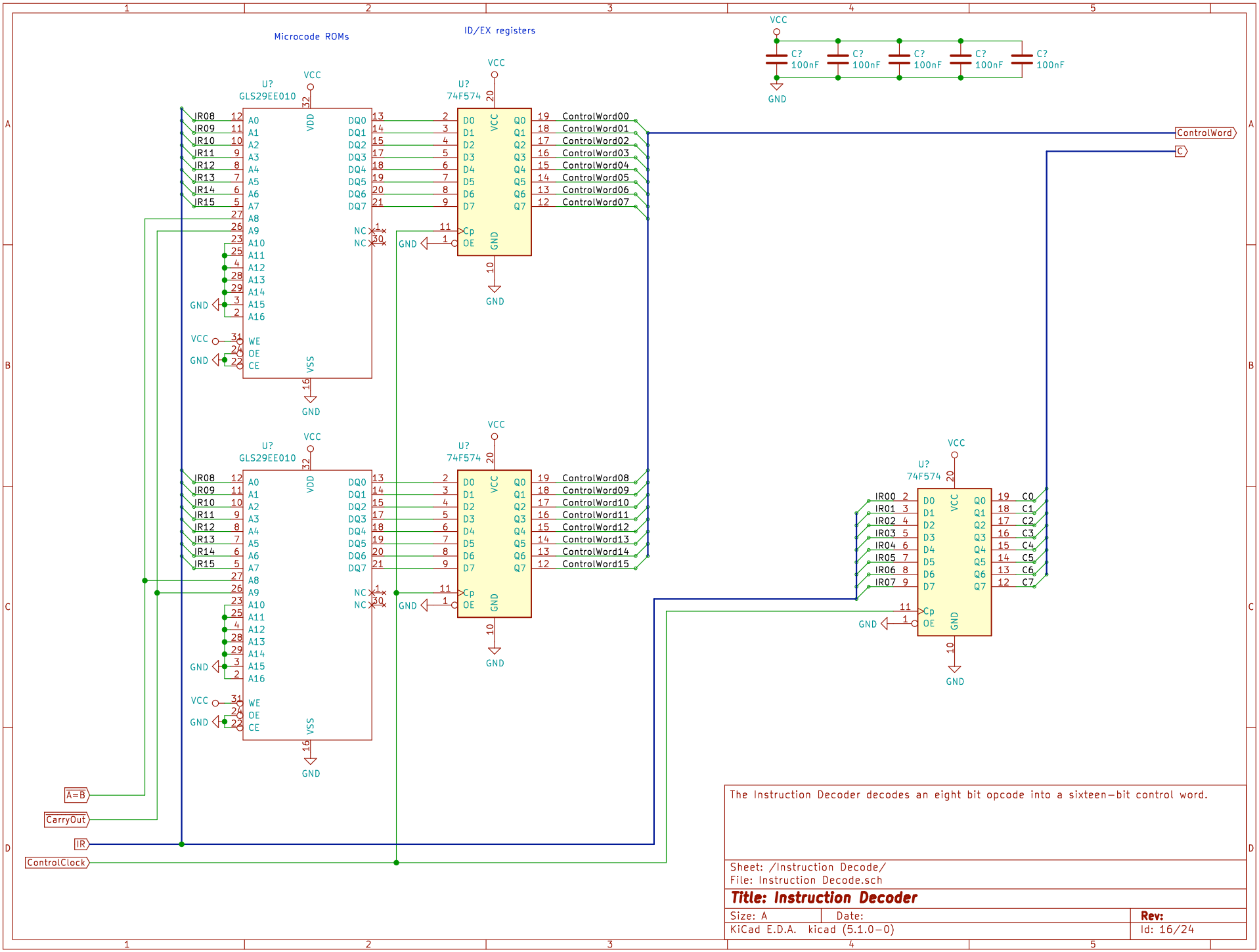
Logic for the control signals which drive Instruction RAM.

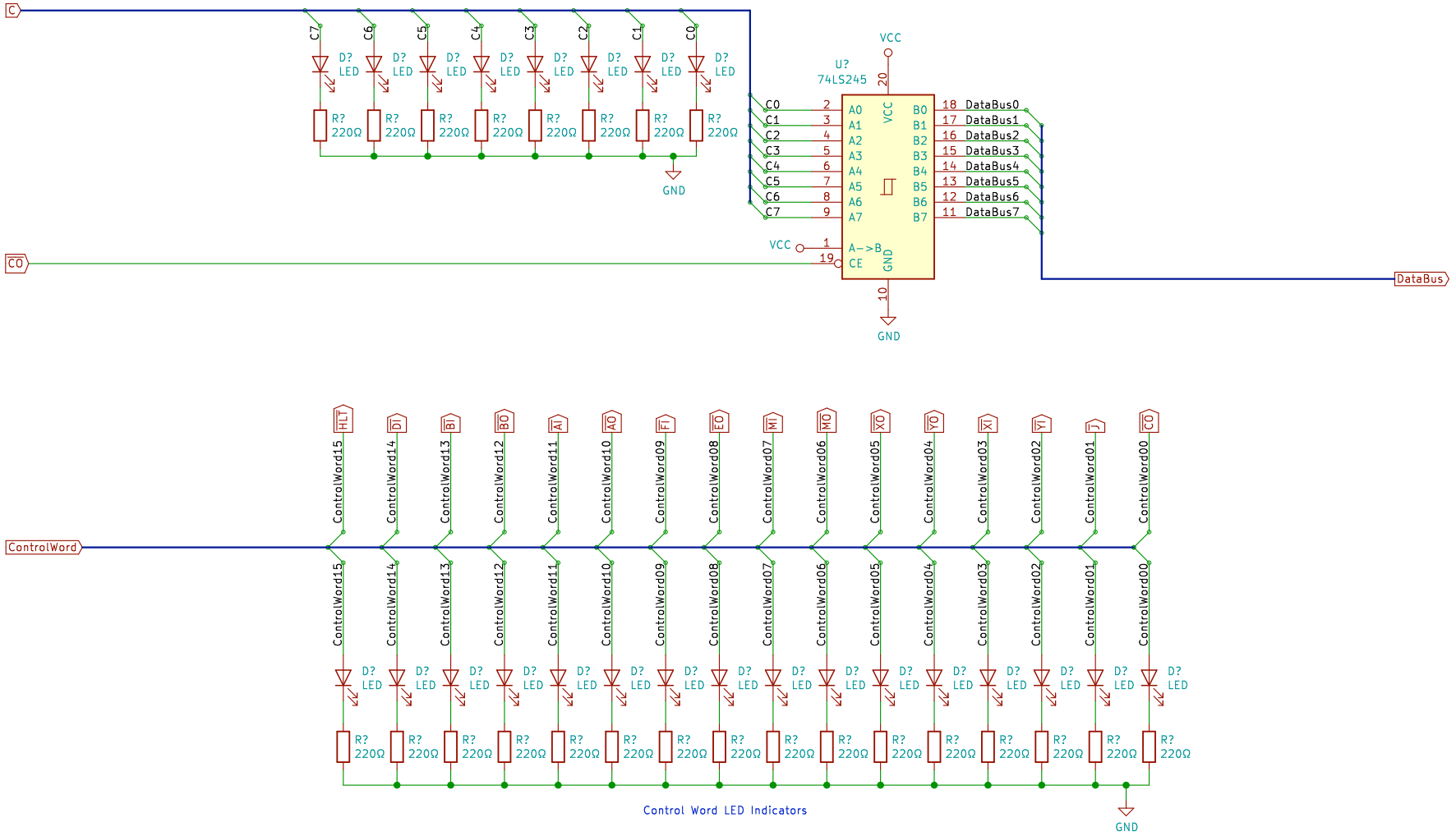
Sheet: /Instruction Fetch/Instruction RAM/Upper Instruction RAM Control/
File: Instruction RAM Control.sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. kicad (5.1.0-0)		Id: 14/24

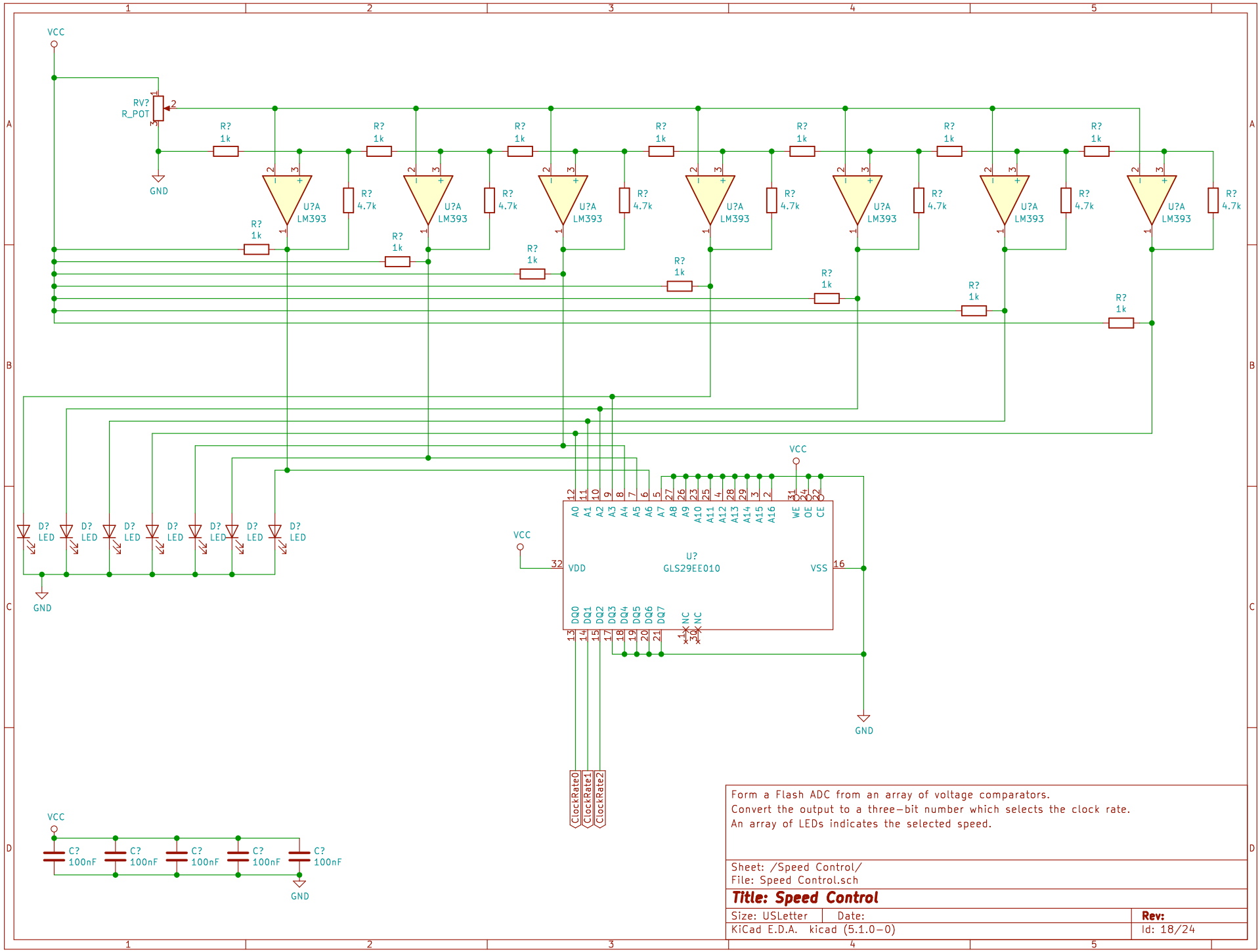

IROM
IRAM
PC
U? 74F157
IROM00 IRAM00 IROM01 IRAM01 IROM02 IRAM02 IROM03 IRAM03
IROM04 IRAM04 IROM05 IRAM05 IROM06 IRAM06 IROM07 IRAM07
IROM08 IRAM08 IROM09 IRAM09 IROM10 IRAM10 IROM11 IRAM11
IROM12 IRAM12 IROM13 IRAM13 IROM14 IRAM14 IROM15 IRAM15
PC15
VCC
GND
U? 74F574
IR00 IR01 IR02 IR03 IR04 IR05 IR06 IR07
IR08 IR09 IR10 IR11 IR12 IR13 IR14 IR15
IR
ControlClock
D? LED
R? 220Ω
C? 100nF
The Instruction Register receives input from either Instruction ROM or Instruction RAM.
Sheet: /Instruction Fetch/Instruction Register/
File: Instruction Register.sch
Title: Instruction Register
Size: A
Date:
Rev:
KiCad E.D.A. kicad (5.1.0-0)
Id: 15/24

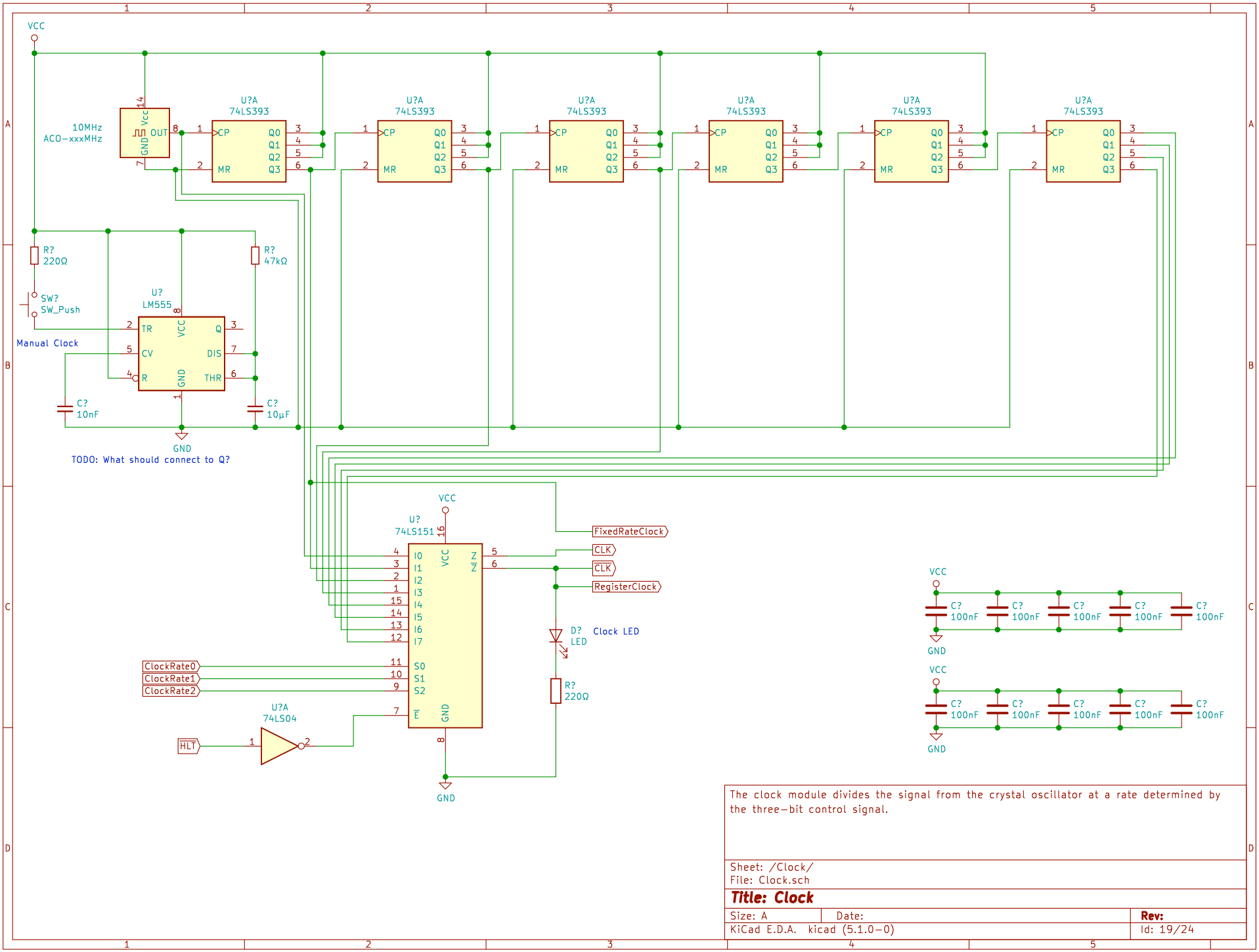

Microcode ROMs
ID/EX registers
VCC
GND
C? 100nF
U? GLS29EE010
U? 74F574
IR08
IR09
IR10
IR11
IR12
IR13
IR14
IR15
ControlWord00
ControlWord01
ControlWord02
ControlWord03
ControlWord04
ControlWord05
ControlWord06
ControlWord07
ControlWord08
ControlWord09
ControlWord10
ControlWord11
ControlWord12
ControlWord13
ControlWord14
ControlWord15
ControlWord
C
IR00
IR01
IR02
IR03
IR04
IR05
IR06
IR07
C0
C1
C2
C3
C4
C5
C6
C7
A=B
CarryOut
IR
ControlClock
The Instruction Decoder decodes an eight bit opcode into a sixteen-bit control word.
Sheet: /Instruction Decode/
File: Instruction Decode.sch
Title: Instruction Decoder
Size: A
Date:
Rev:
KiCad E.D.A. kicad (5.1.0-0)
Id: 16/24

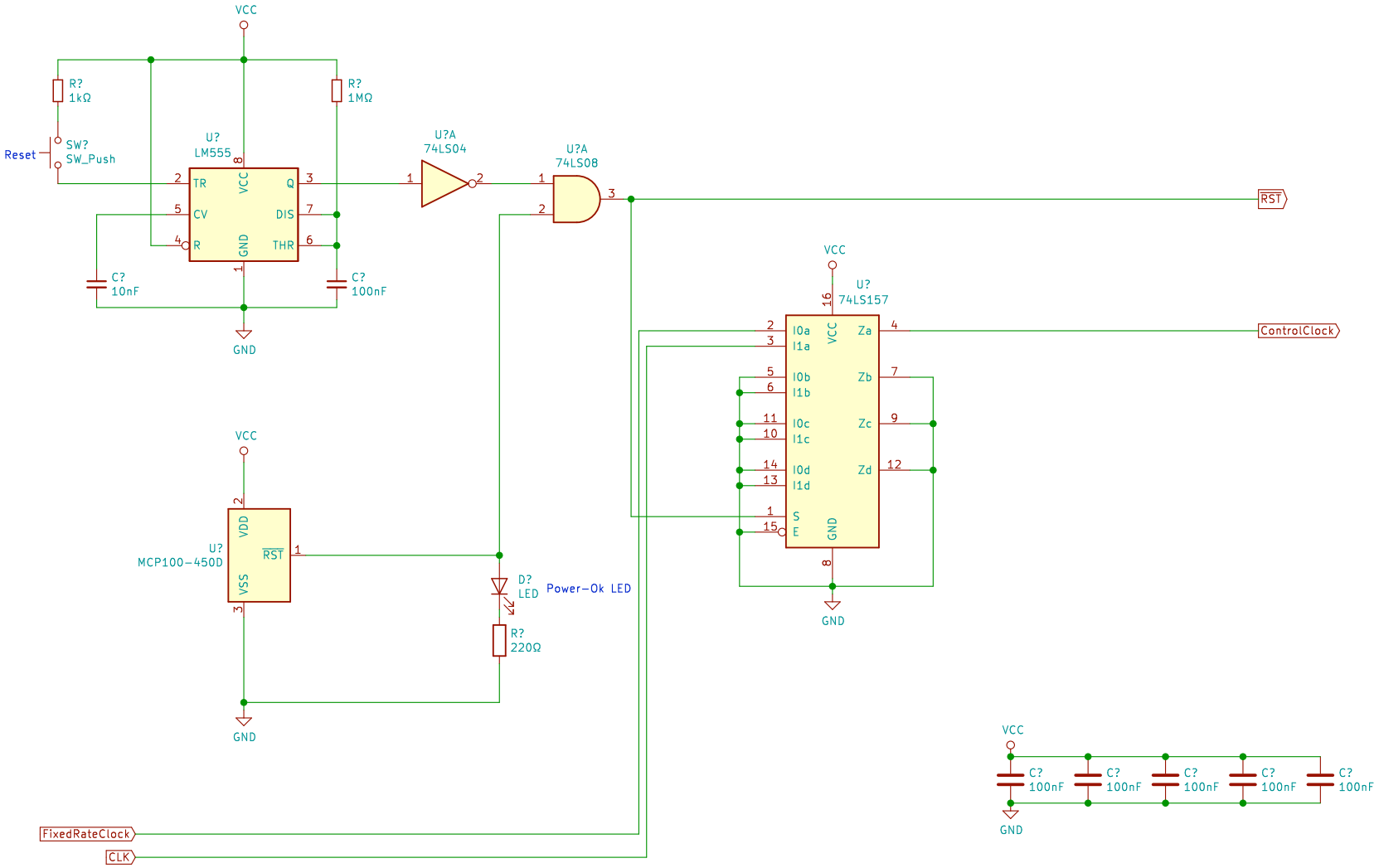


The lower eight bits of the instruction register can be expressed onto the data bus as a constant value.

Sheet: /Execute/	
File: Execute.sch	
Title: Execute	
Size: A	Date:
KiCad E.D.A. kicad (5.1.0-0)	Id: 17/24


VCC
RV?
R_POT
GND
R?
1k
4.7k
U?A
LM393
D?
LED
U?
GLS29EE010
VDD
VSS
WE
OE
CE
NC
ClockRate0
ClockRate1
ClockRate2
C?
100nF
Form a Flash ADC from an array of voltage comparators.
Convert the output to a three-bit number which selects the clock rate.
An array of LEDs indicates the selected speed.
Sheet: /Speed Control/
File: Speed Control.sch
Title: Speed Control
Size: USLetter
Date:
Rev:
KiCad E.D.A. kicad (5.1.0-0)
Id: 18/24


VCC
10MHz
ACO-xxxMHz
U?A
74LS393
R?
220Ω
R?
47kΩ
SW?
SW_Push
Manual Clock
U?
LM555
C?
10nF
C?
10µF
GND
TODO: What should connect to Q?
U?
74LS151
FixedRateClock
CLK
CLK
RegisterClock
D?
LED
Clock LED
ClockRate0
ClockRate1
ClockRate2
U?A
74LS04
HLT
C?
100nF
The clock module divides the signal from the crystal oscillator at a rate determined by the three-bit control signal.
Sheet: /Clock/
File: Clock.sch
Title: Clock
Size: A
Date:
Rev:
KiCad E.D.A. kicad (5.1.0-0)
Id: 19/24

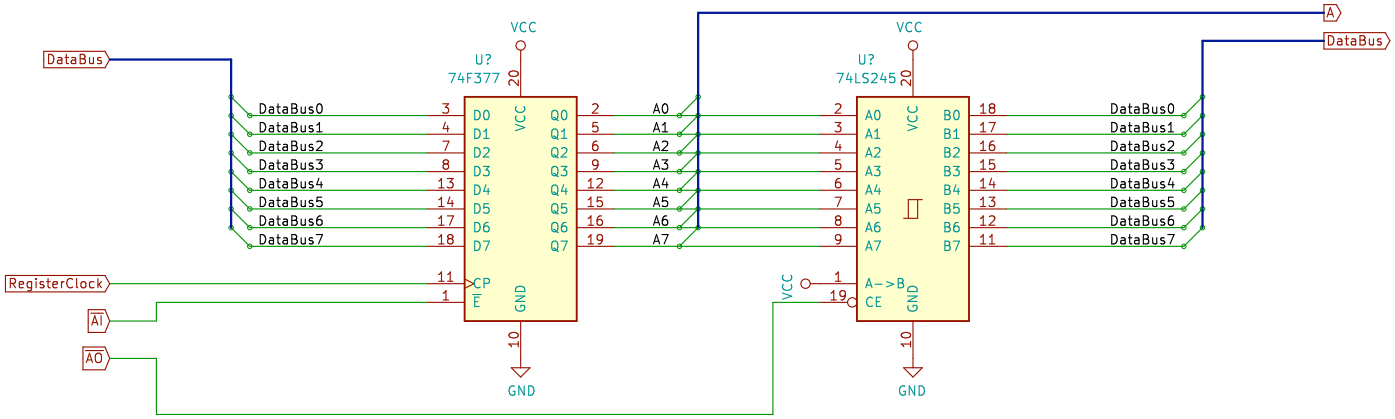


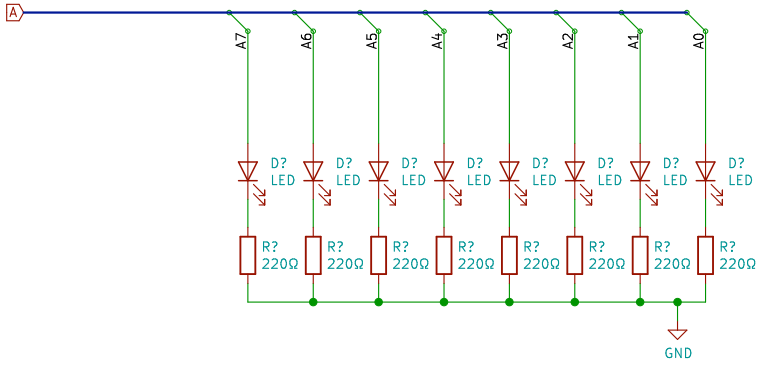
The MCP100 provides Power-on Reset functionality.
A button is also provided to manually reset the machine.
During reset, the control clock is pulsed repeatedly to flush the pipeline.

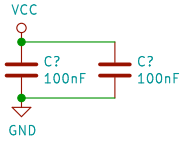
Sheet: /Power-on Reset/
File: Power-on Reset.sch

Title: Power-on Reset

Size: A	Date:	Rev:
KiCad E.D.A. kicad (5.1.0-0)		Id: 20/24





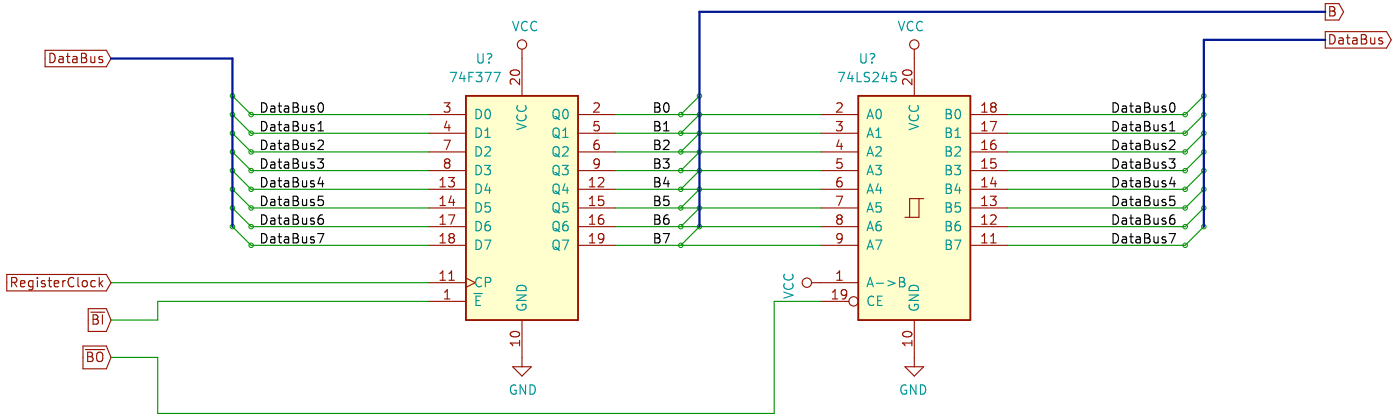


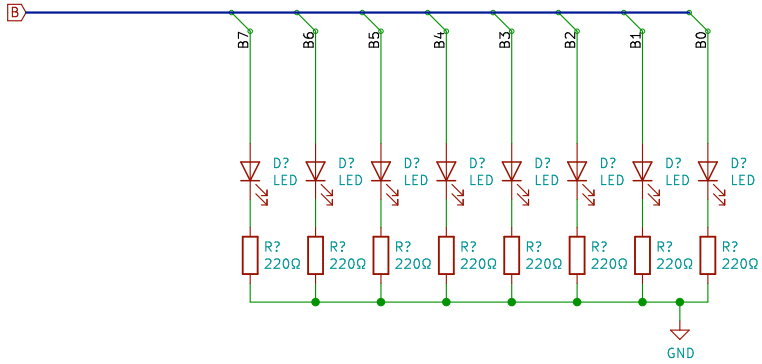
Register A is wired to the ALU's A operand.

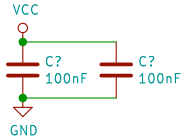
Sheet: /Register A/
File: Register A.sch

Title: Register A

Size: A	Date:	**Rev:**
KiCad E.D.A. kicad (5.1.0-0)		Id: 21/24





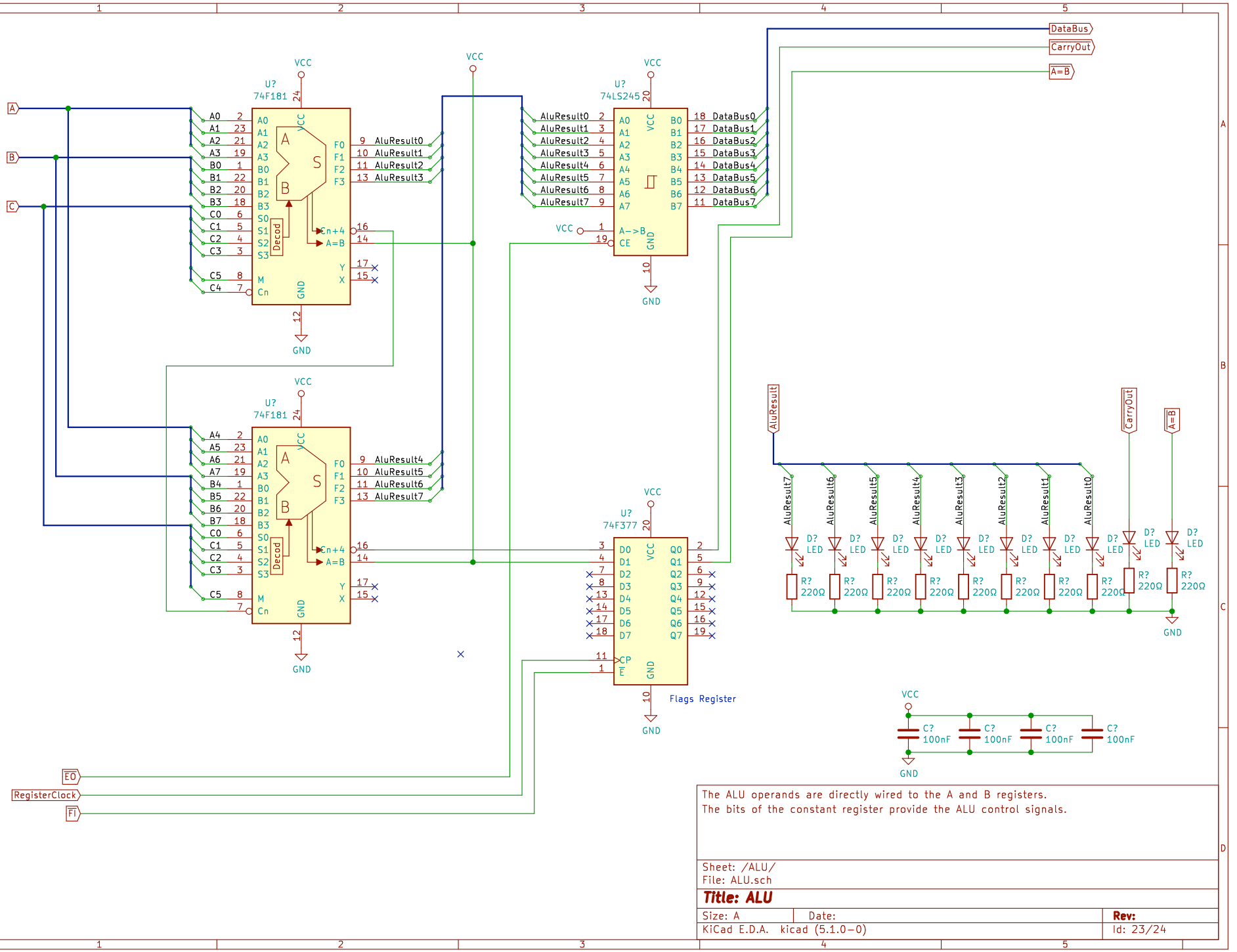


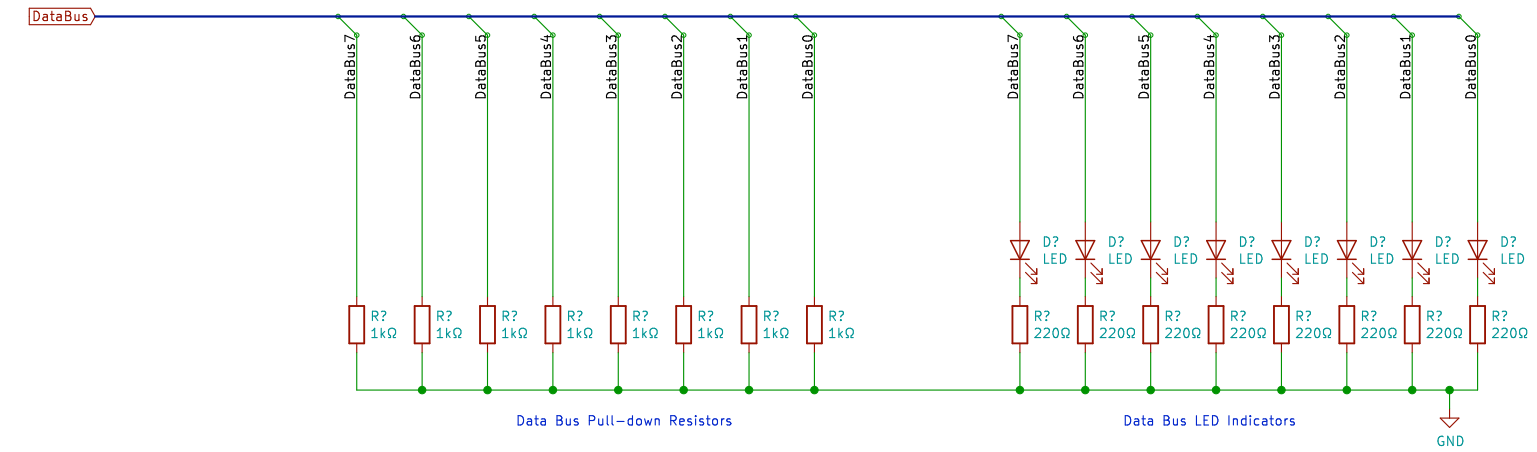
Register B is wired to the ALU's B operand.

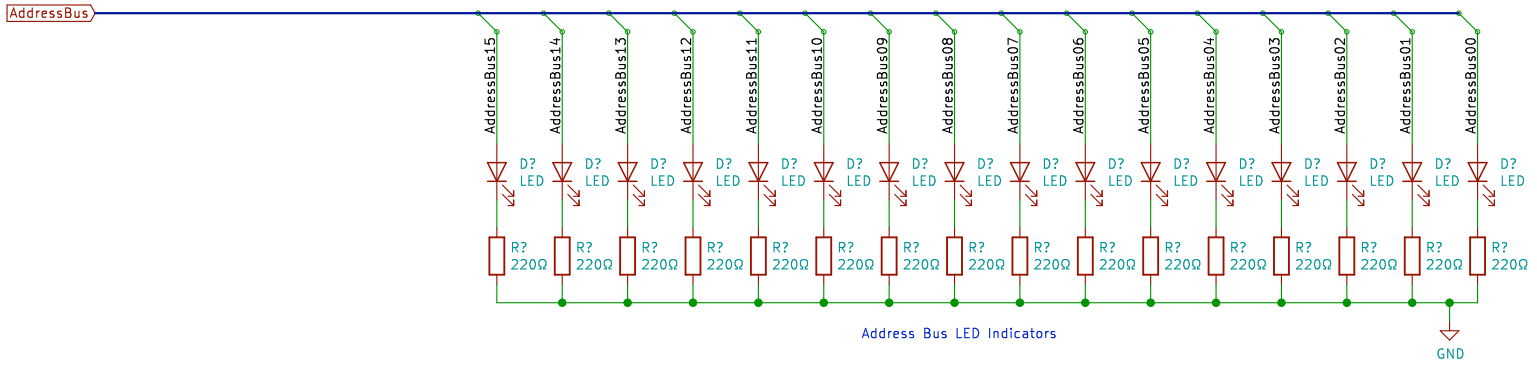
Sheet: /Register B/
File: Register B.sch

Title: Register B

Size: A	Date:	**Rev:**
KiCad E.D.A. kicad (5.1.0-0)		Id: 22/24


DataBus
CarryOut
A=B
U?
74F181
U?
74LS245
U?
74F377
Flags Register
AluResult
EO
RegisterClock
FI
VCC
GND
The ALU operands are directly wired to the A and B registers.
The bits of the constant register provide the ALU control signals.
Sheet: /ALU/
File: ALU.sch
Title: ALU
Size: A
Date:
Rev:
KiCad E.D.A. kicad (5.1.0-0)
Id: 23/24





Sheet: /Bus Display/		
File: Bus Display.sch		
Title: Bus Display		
Size: A	Date:	**Rev:**
KiCad E.D.A. kicad (5.1.0-0)		Id: 24/24