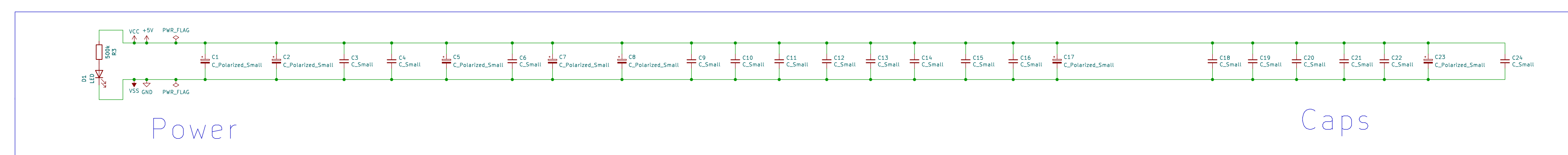
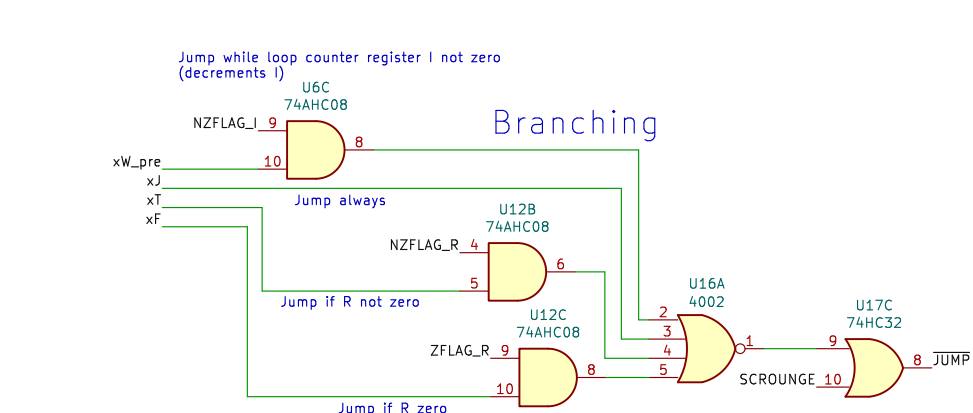


Myth Microcontroller Reference Schematics



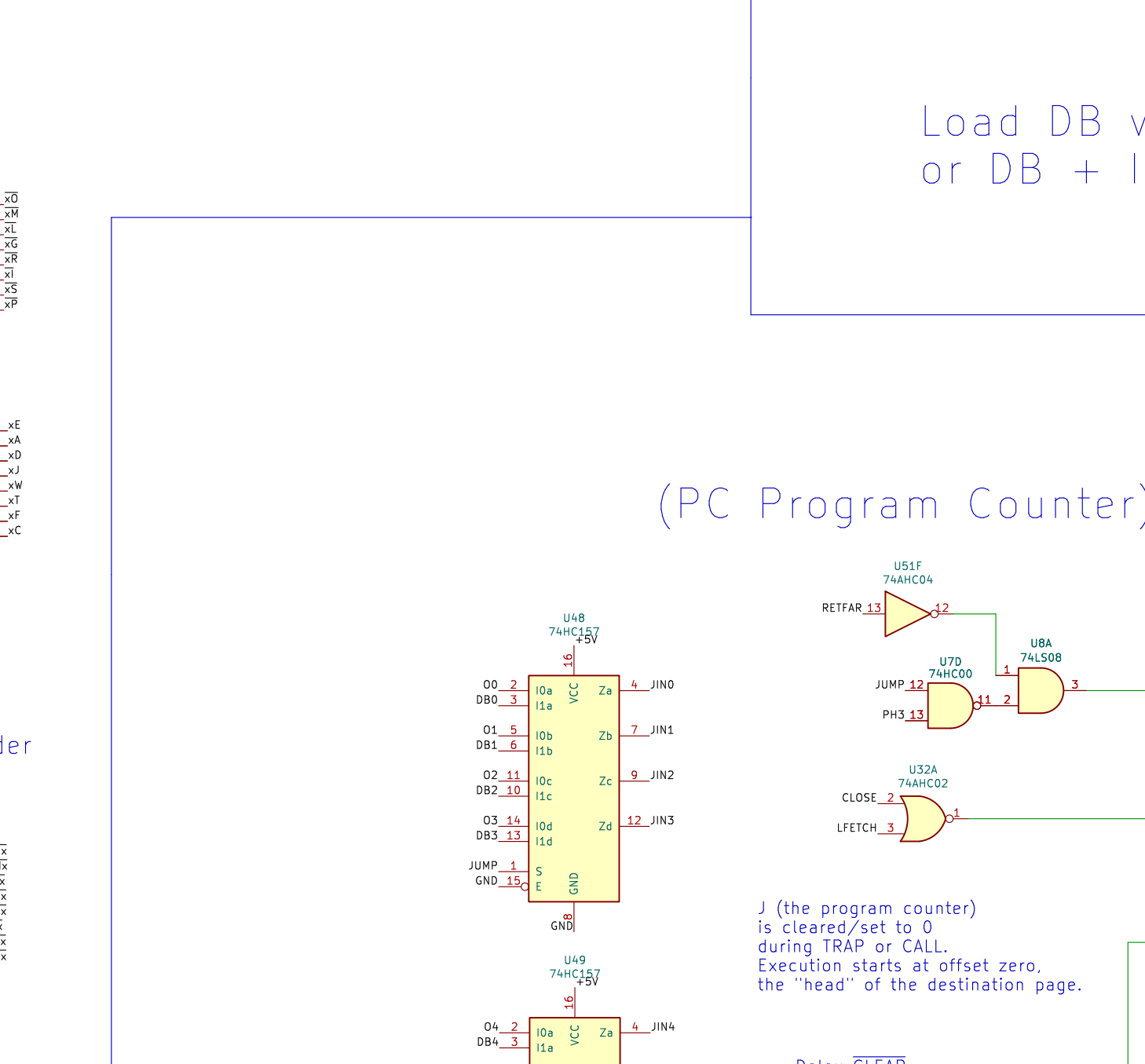
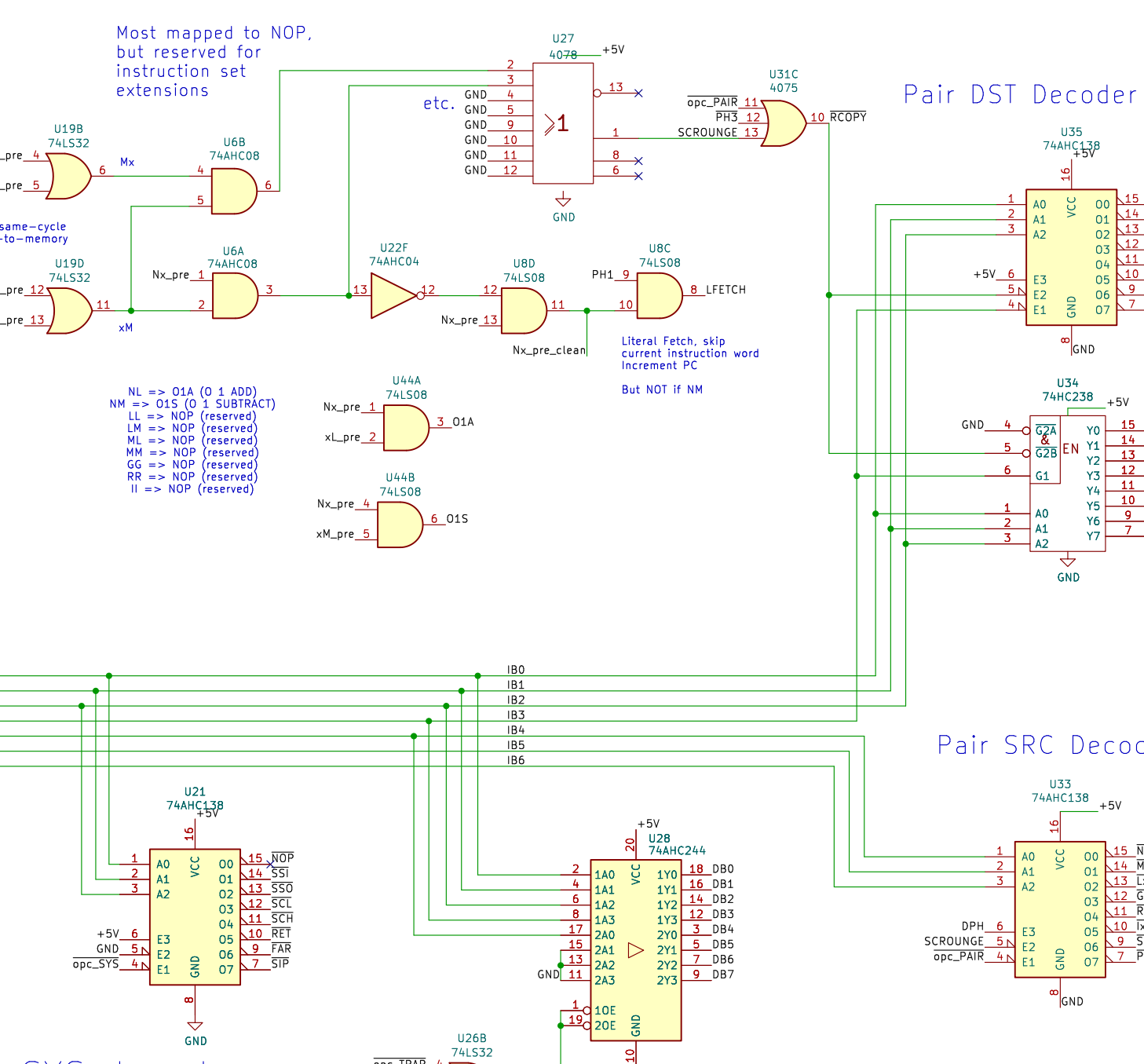
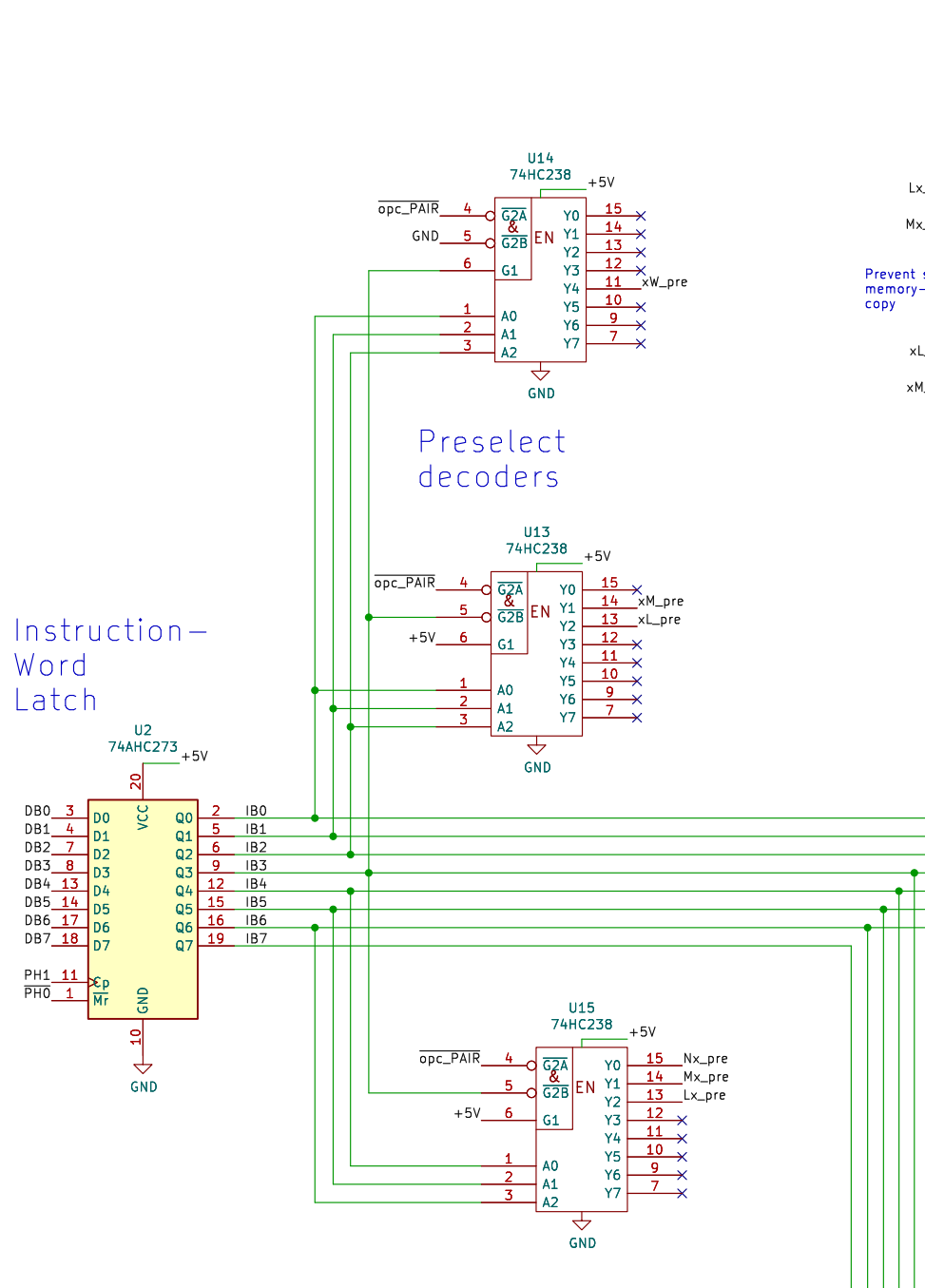
Caps

Instruction Decoder



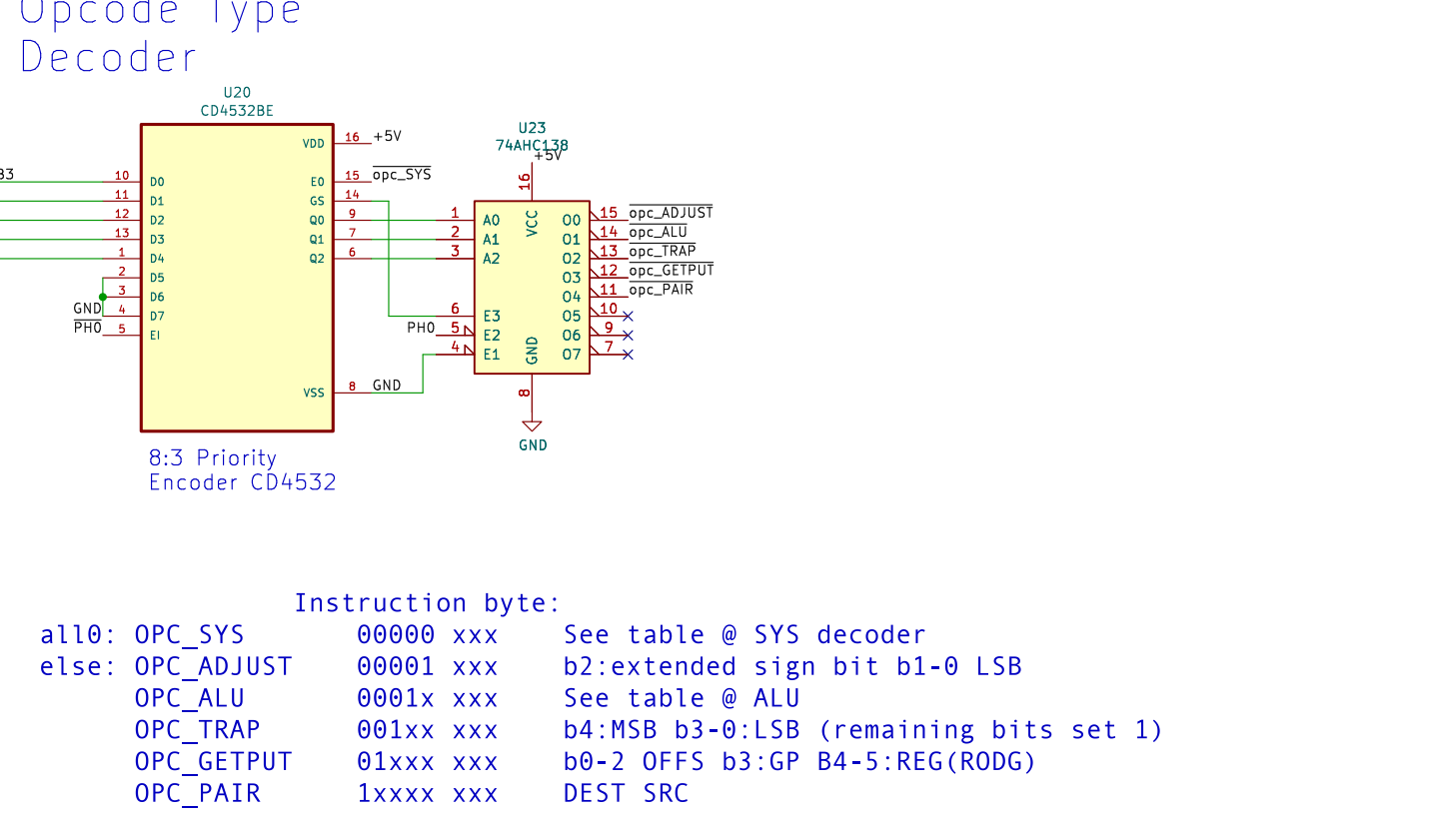
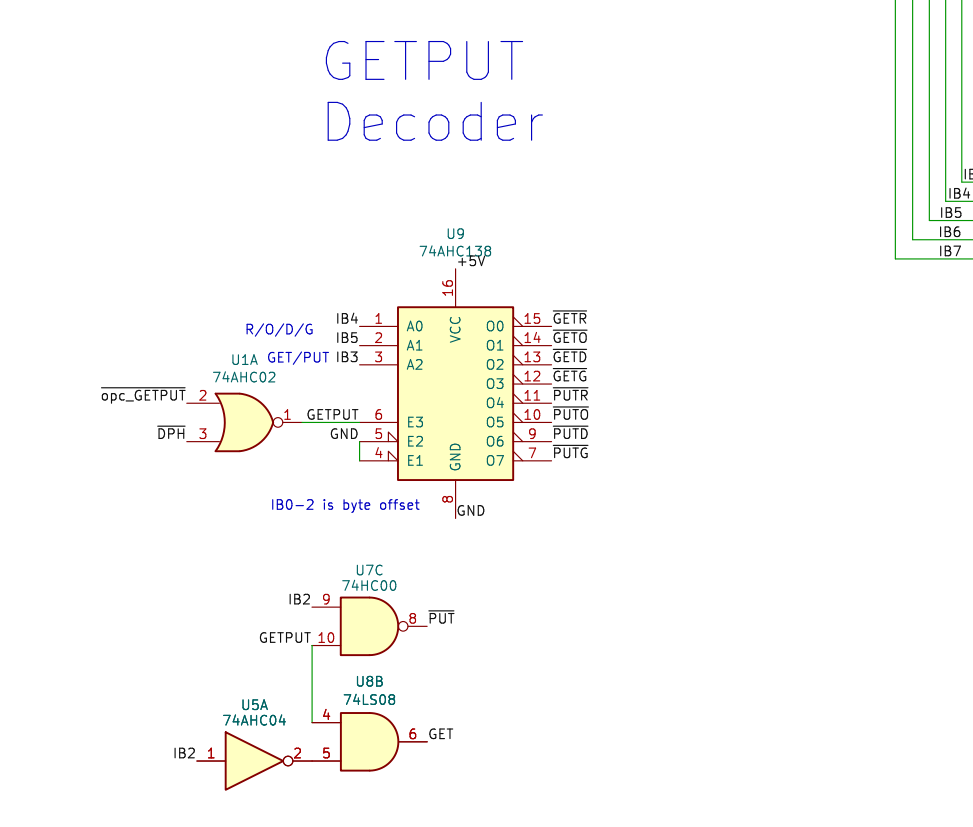
Scrounger

Remap 'scrounger' to 66, 68, 69, 70 etc. and impracticable opcodes such as NM



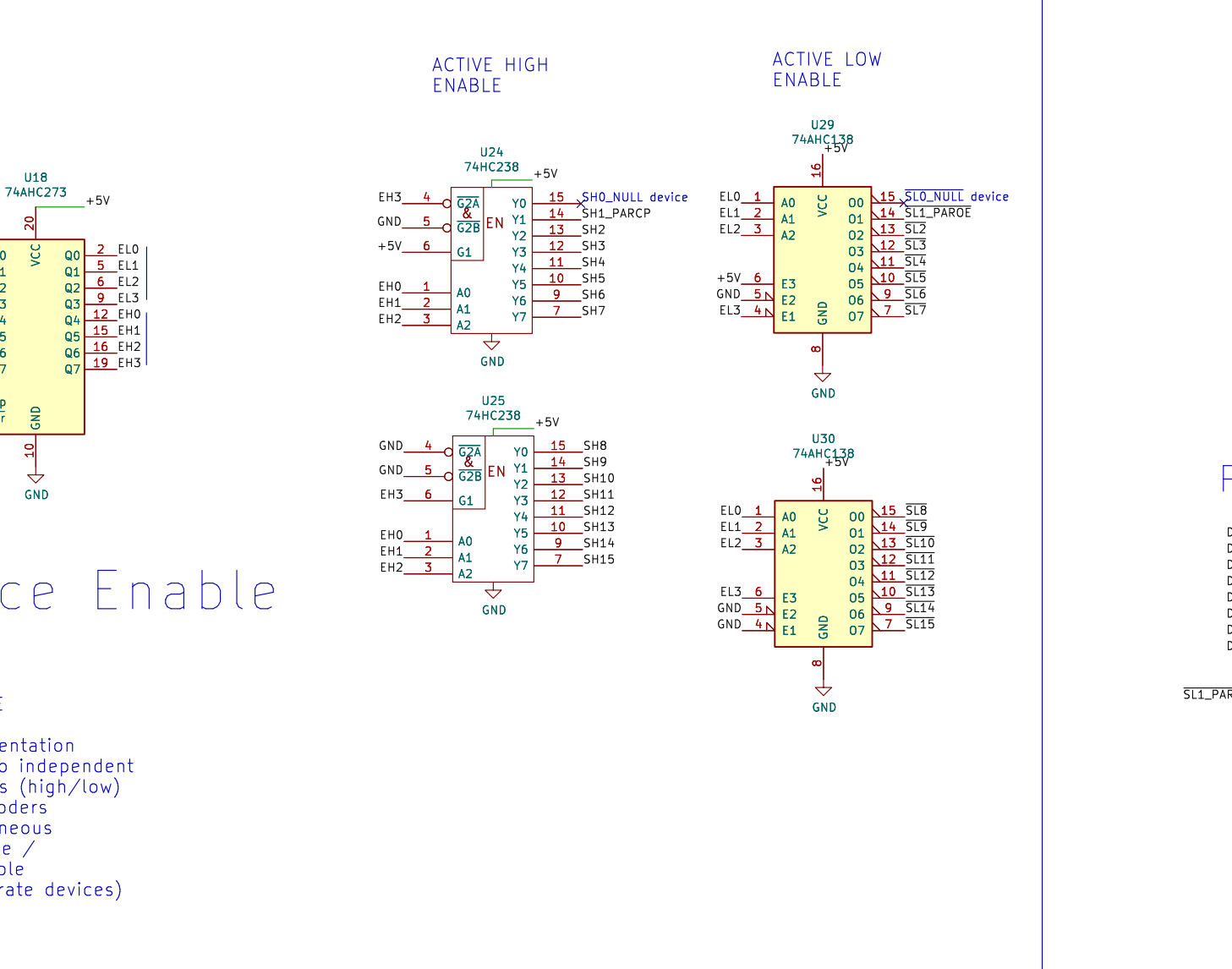
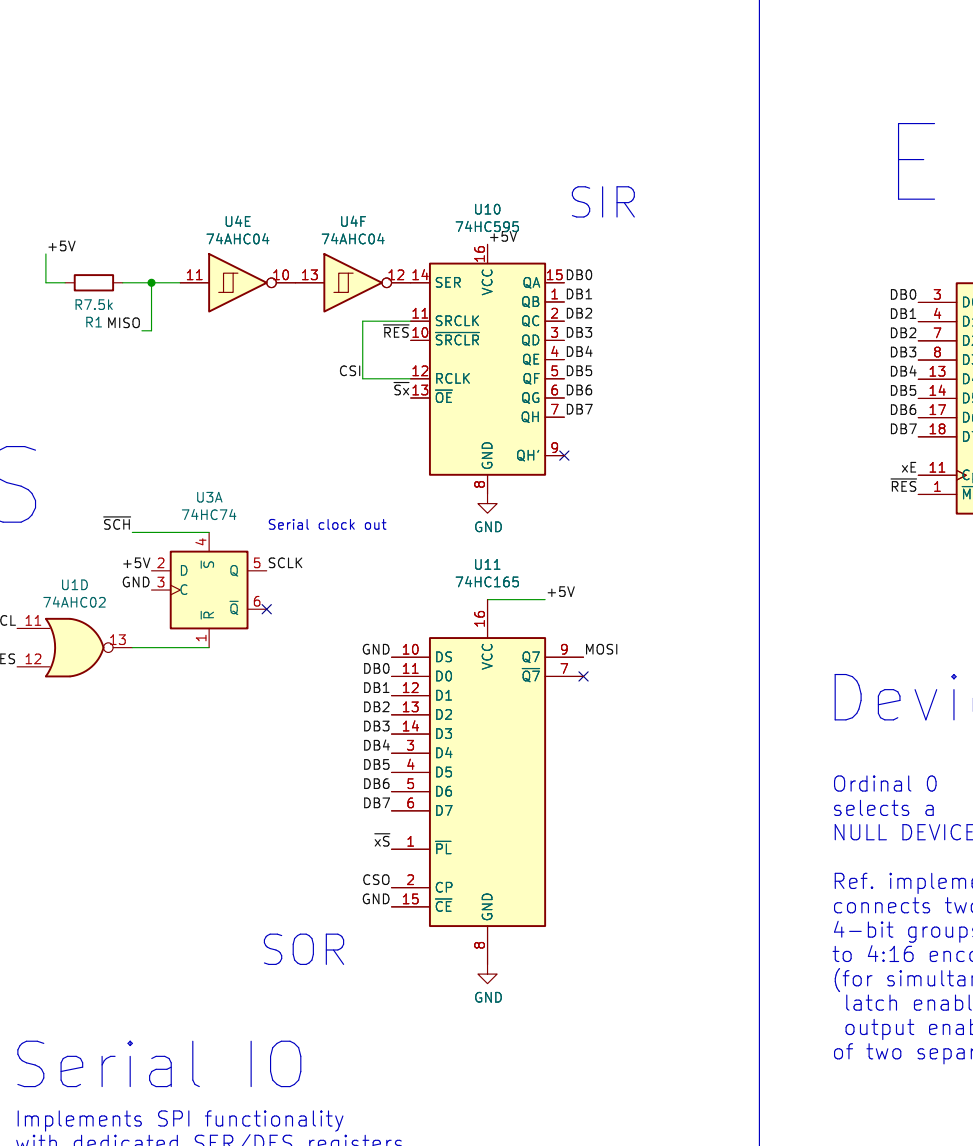
PHA0: Setup PC on address bus
PHA1: Latch opcode into IB0-7, predecoded
PHA2: Decode opcode source into IB or other action
PHA3: Latch source into target or other action
CLOSE: Cleanup

TRAP decoder
Immediate calls



all0: OPC_SYS
else: OPC_ADJUST
OPC_ALU
OPC_TRAP
OPC_GETPUT
OPC_PAIR

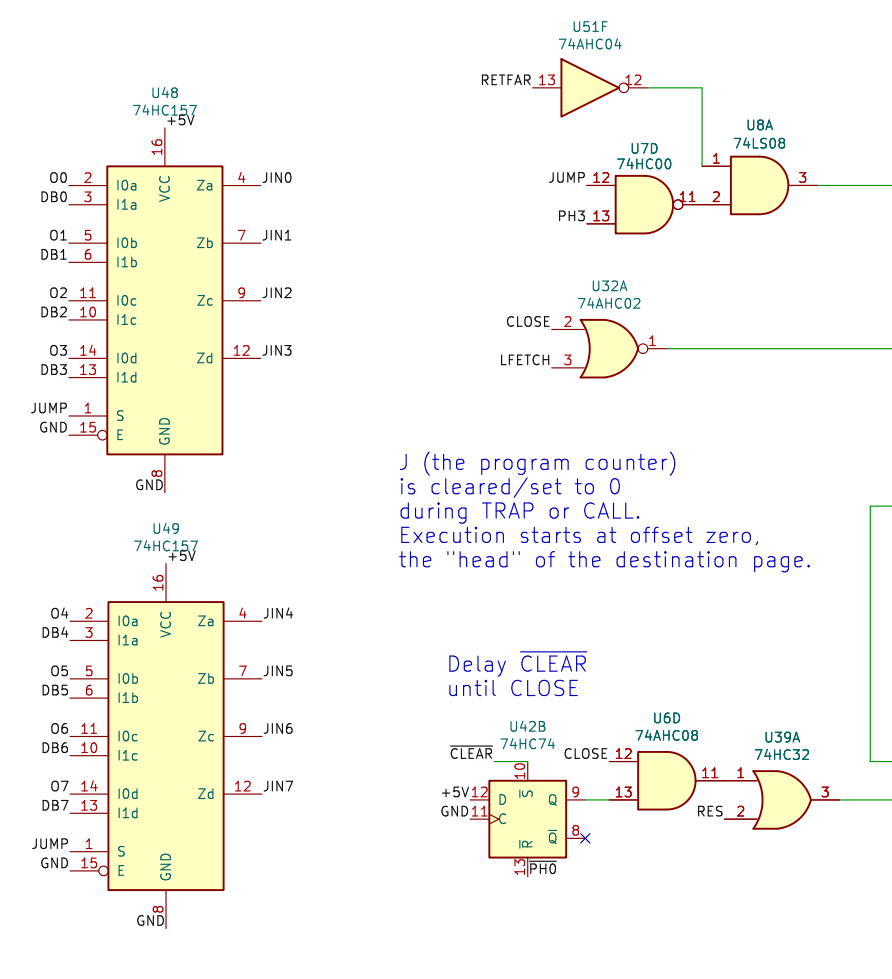
See table @ SYS decoder
b2: extended sign bit b1-0 LSB
See table @ ALU
b4: MSB b3-0: LSB (remaining bits set 1)
b0-2: OFFS b3: GP b4-5: REG (R006)
DEST SRC



Implements SPI functionality with dedicated SER/DES registers

Original 0 selects a NULL DEVICE
Ref. implementation connects two independent 4-bit groups (high/low) to 4-bit encoders (latch enable / output enable) of two separate devices

(PC Program Counter)



Delay CLEAR until CLOSE

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Delay EXIT until CLOSE

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Delay EXIT until CLOSE

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Delay EXIT until CLOSE

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Delay EXIT until CLOSE

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Delay EXIT until CLOSE

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Delay EXIT until CLOSE

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

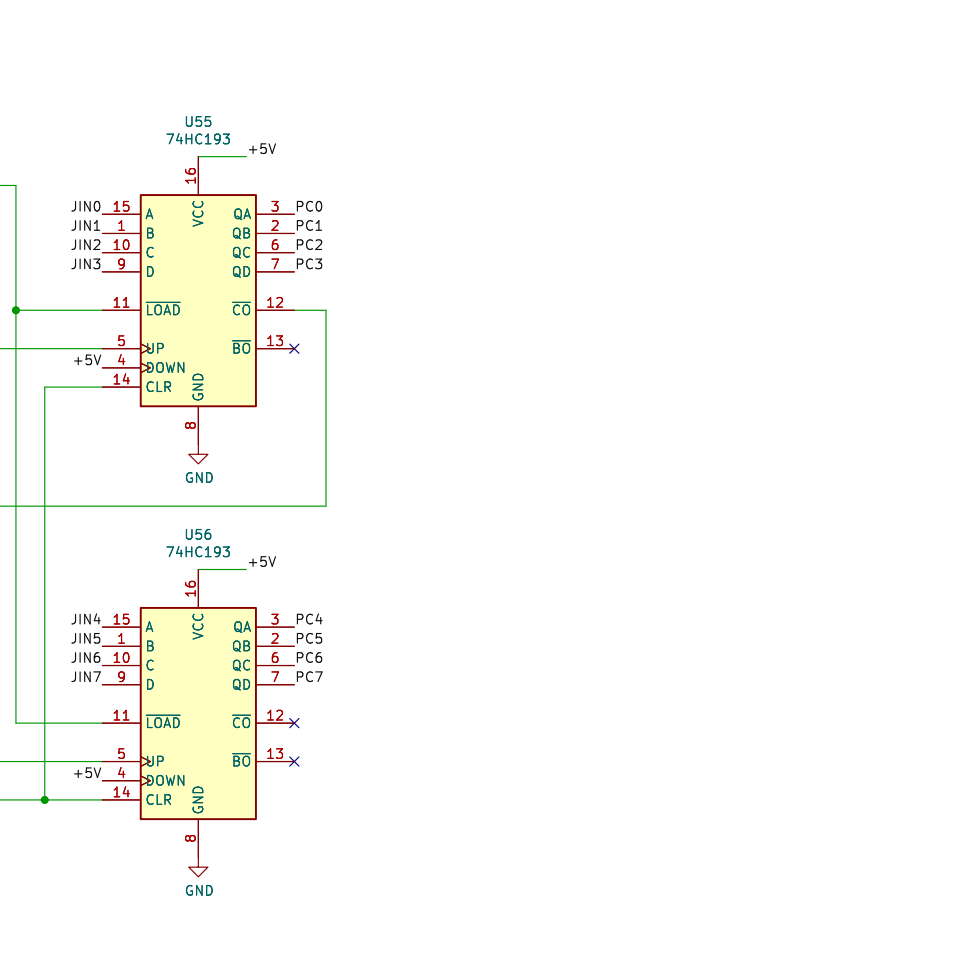
Delay EXIT until CLOSE

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Delay EXIT until CLOSE

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

(GLOBAL)



Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

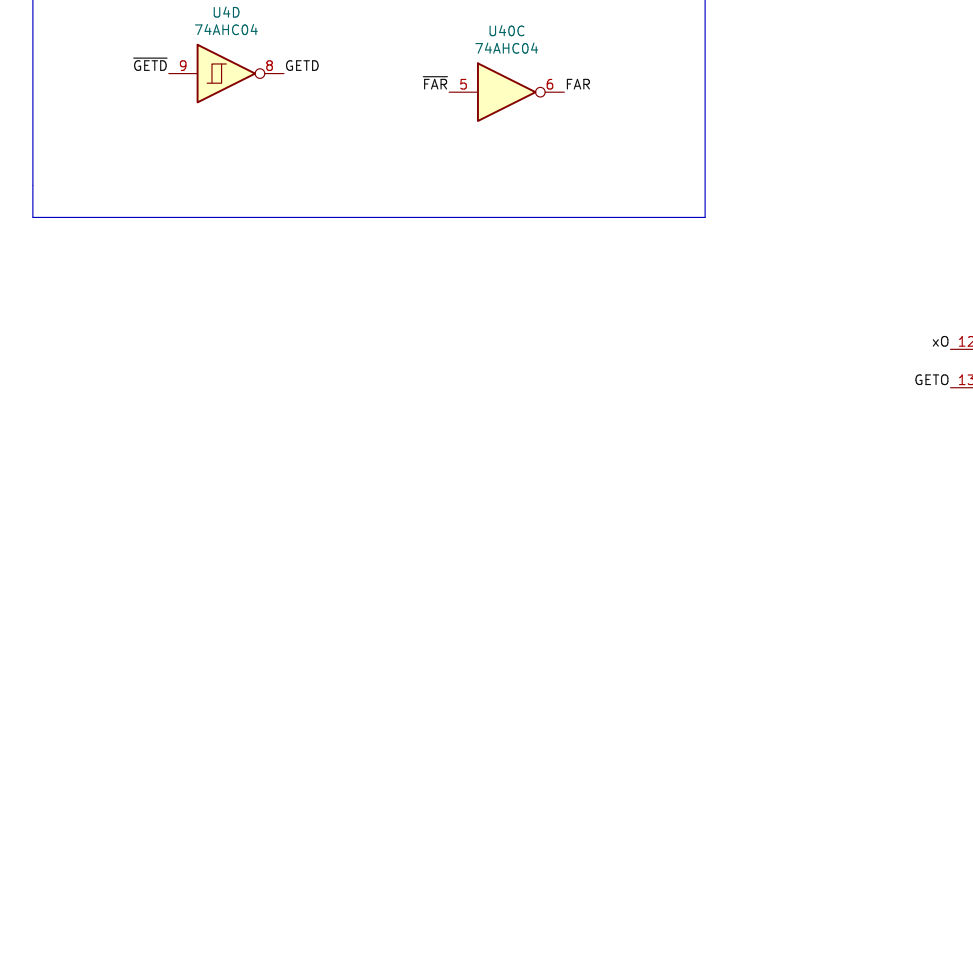
Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

Load DB value (xG) or DB + I (xA = "ADD")

(LOCAL)



Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

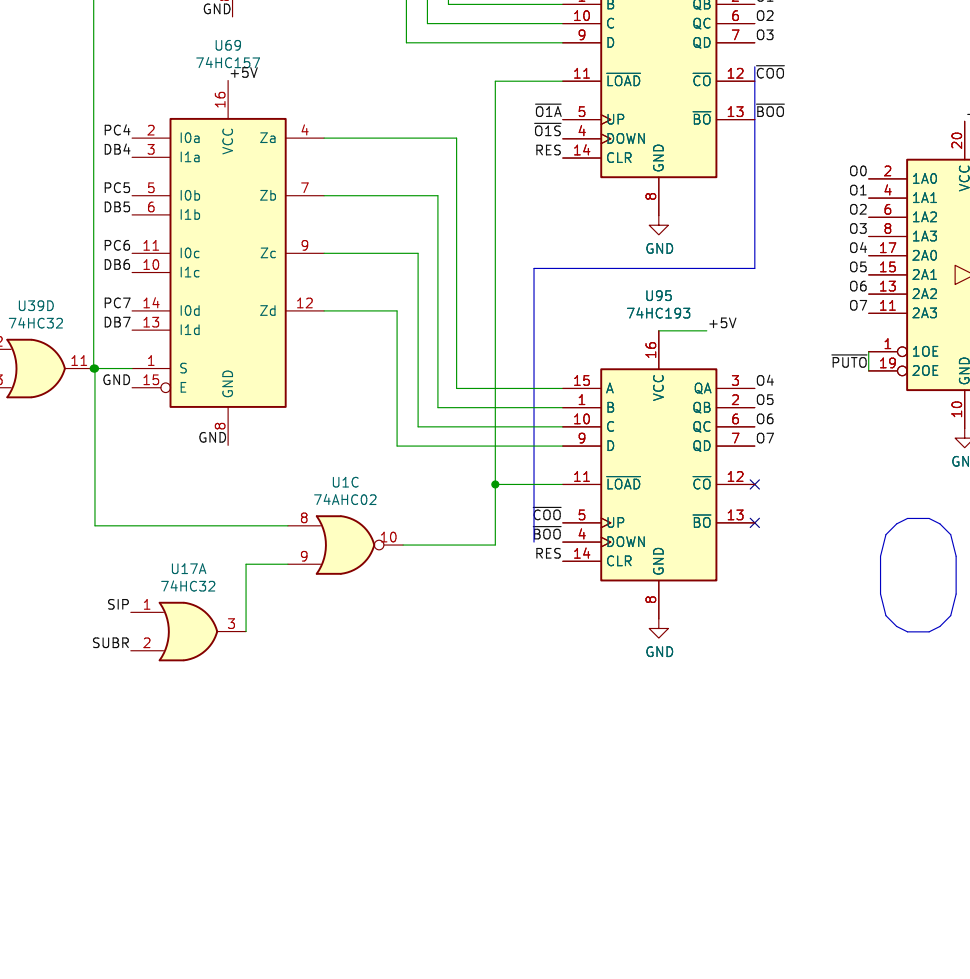
Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

(CODE)



Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

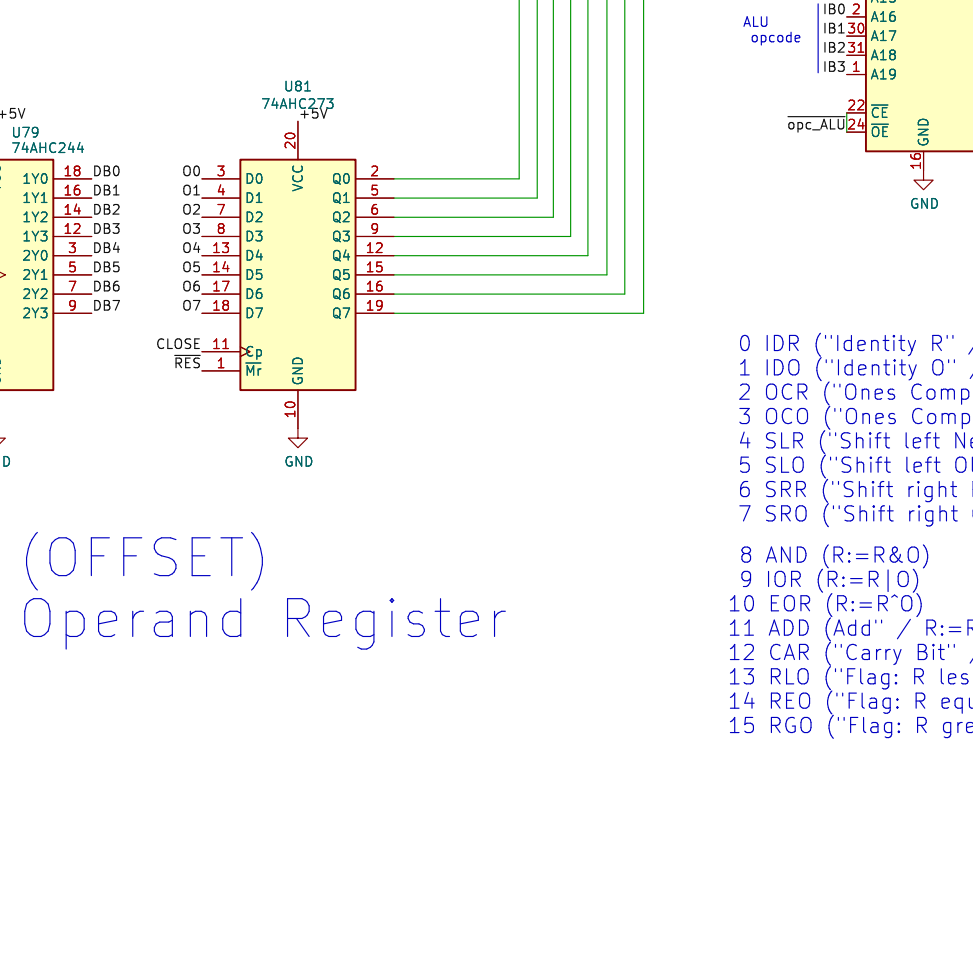
Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

(DATA)



Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

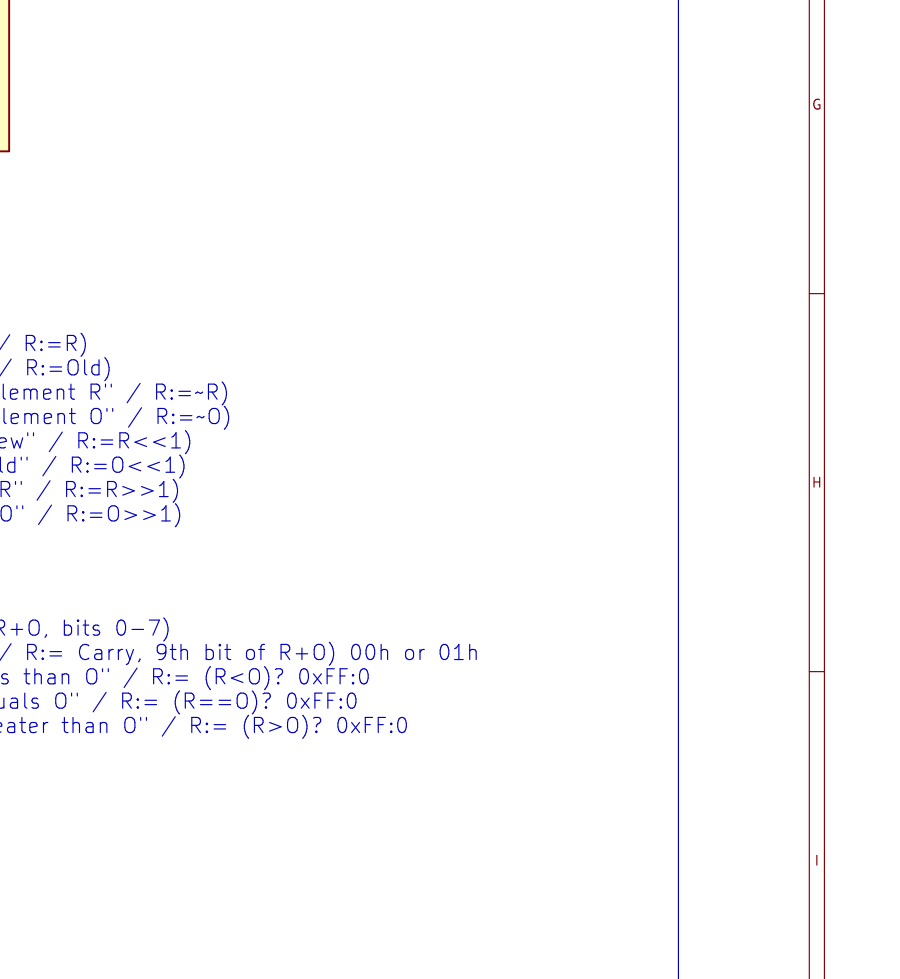
Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

(MEMORY)



Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame