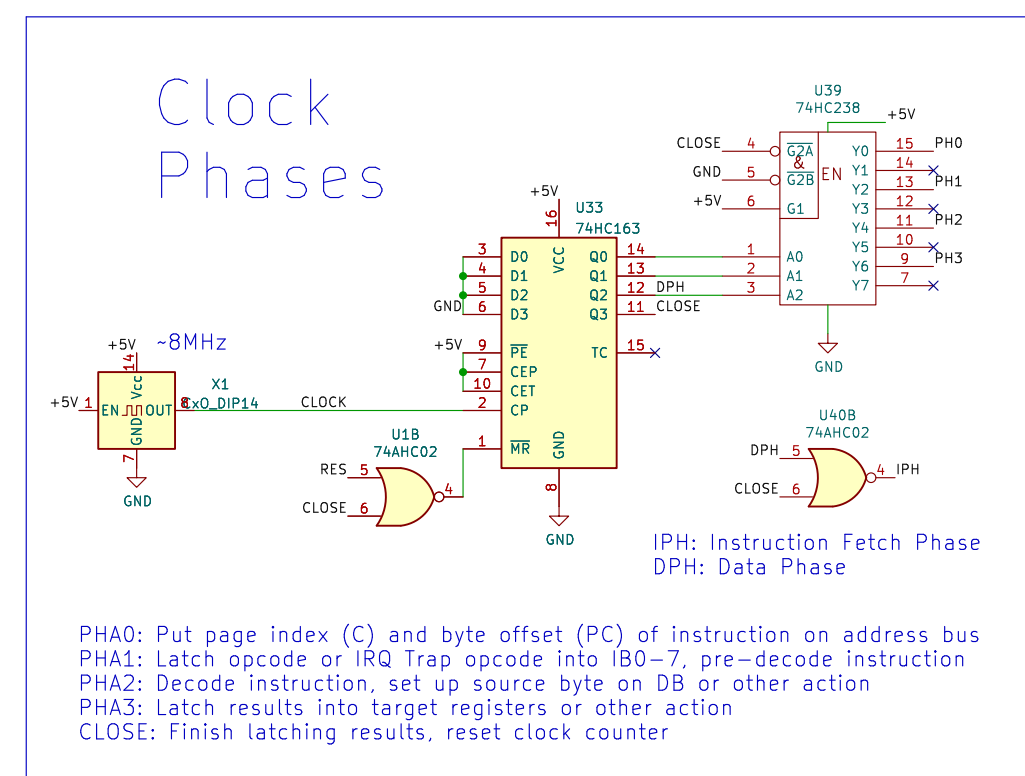
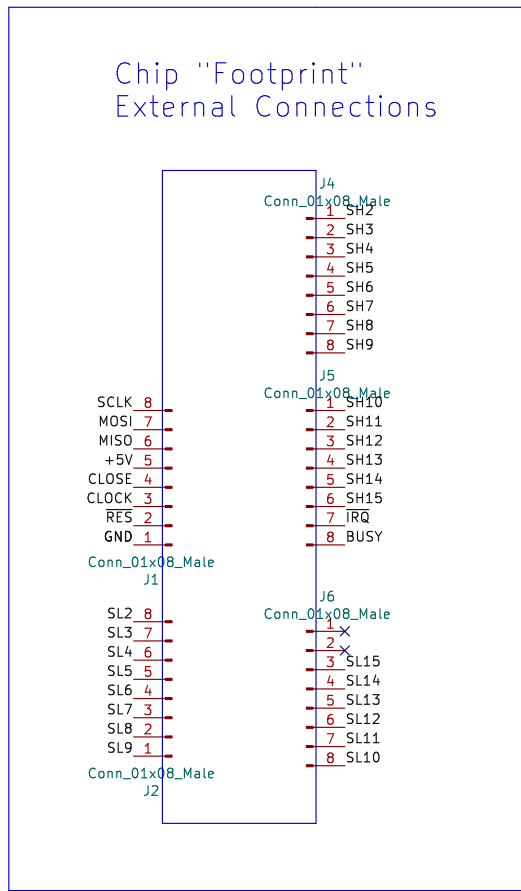


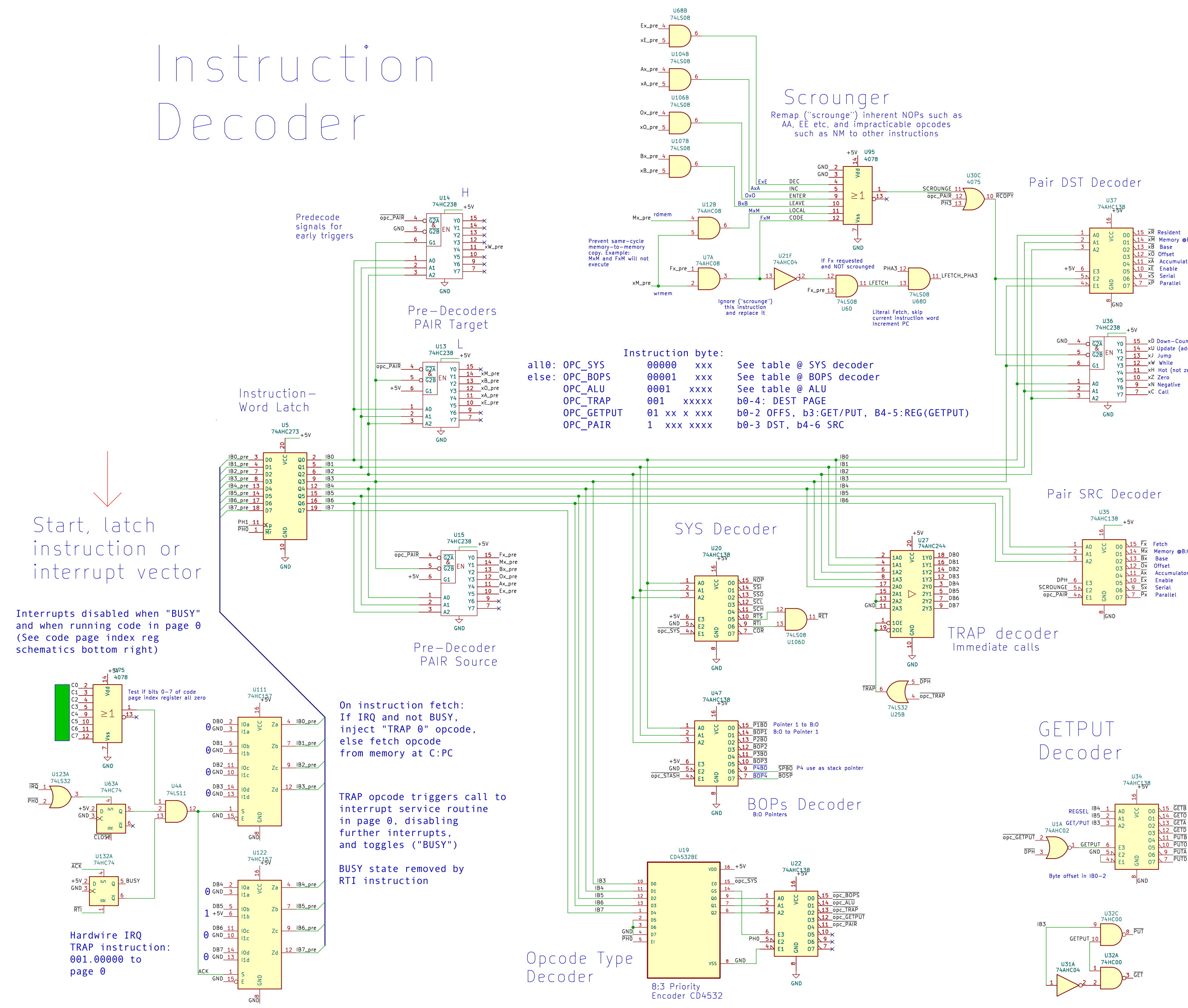
Sonne-8 Microcontroller

Reference Schematics

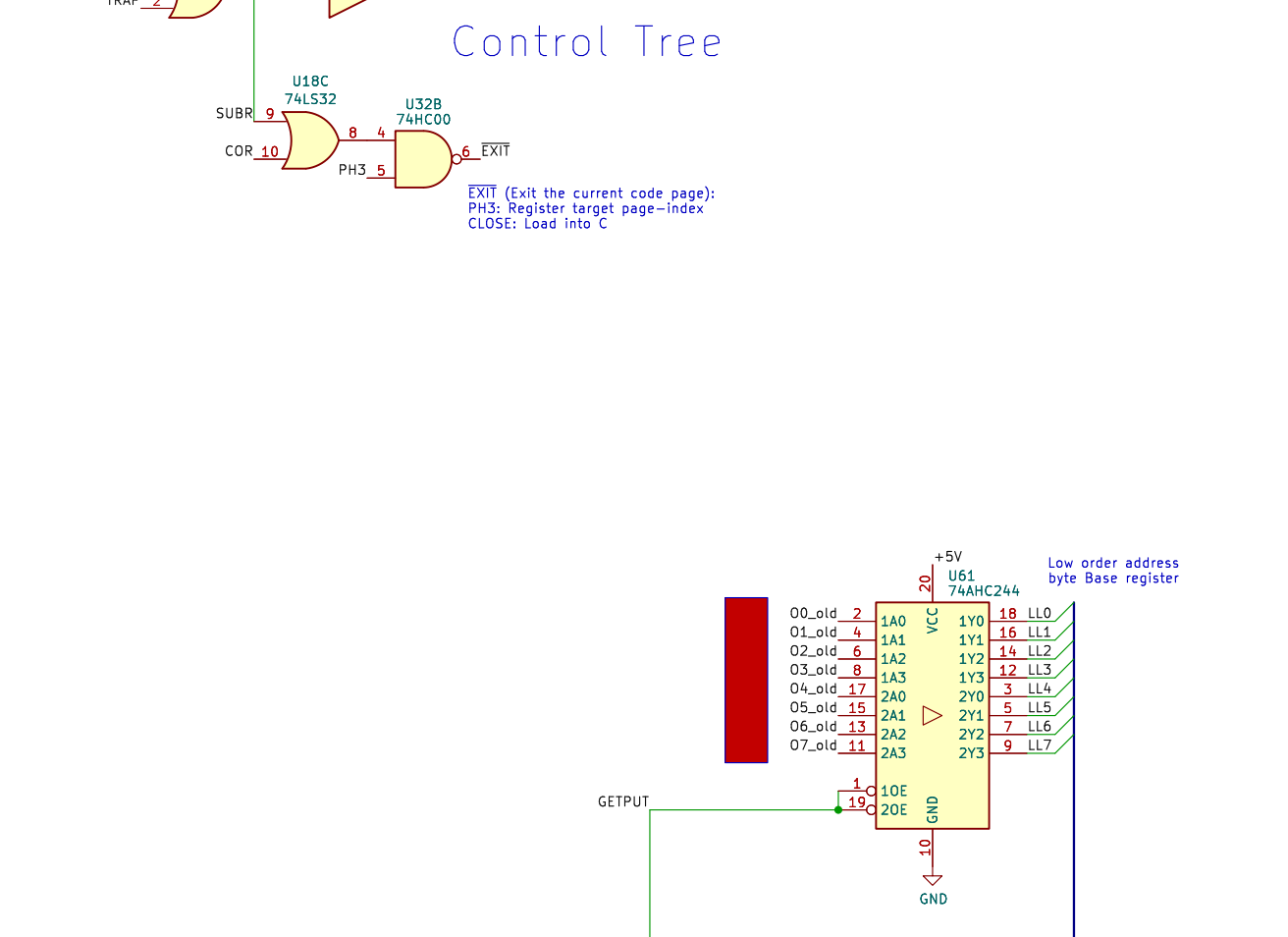
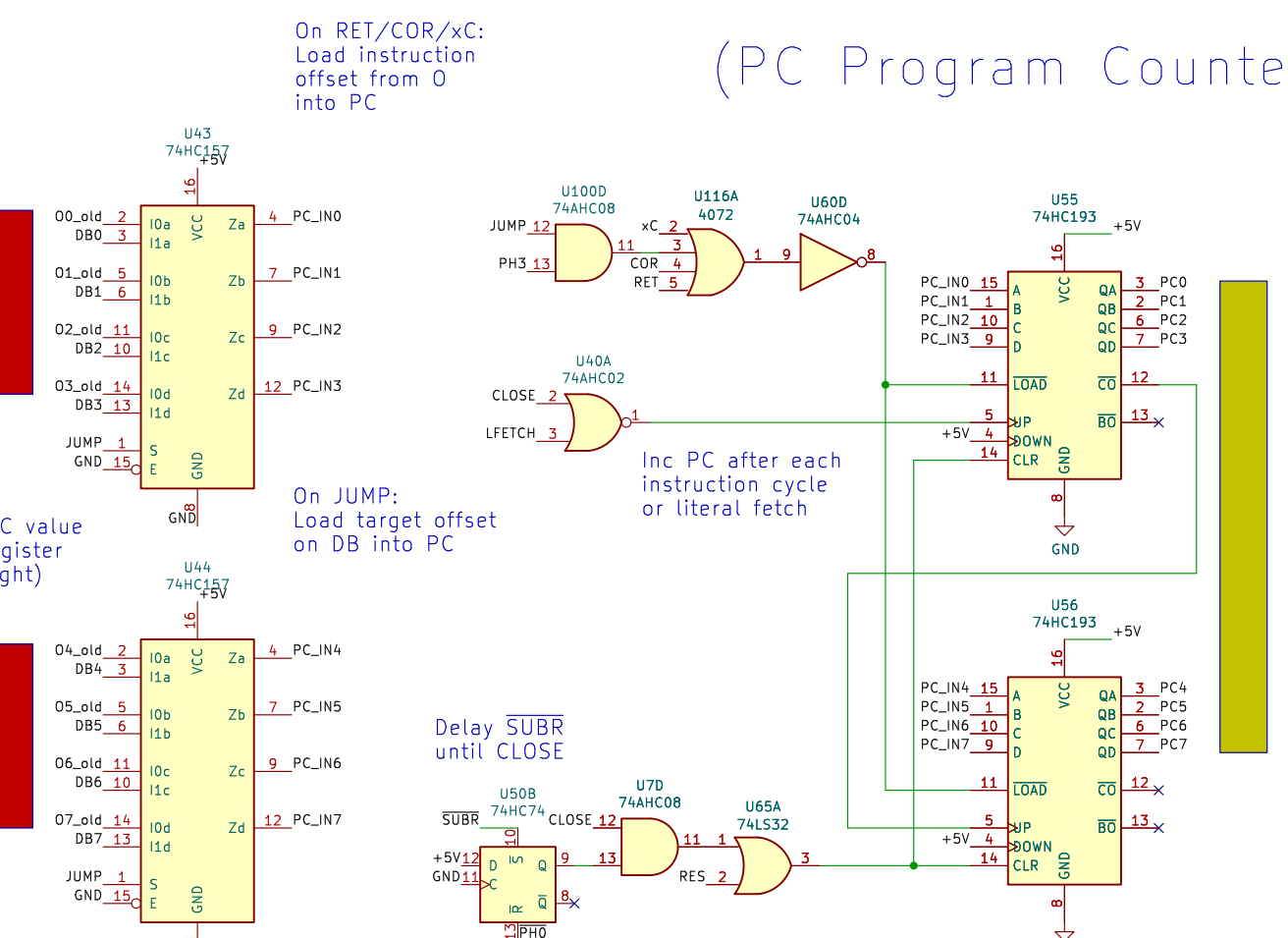
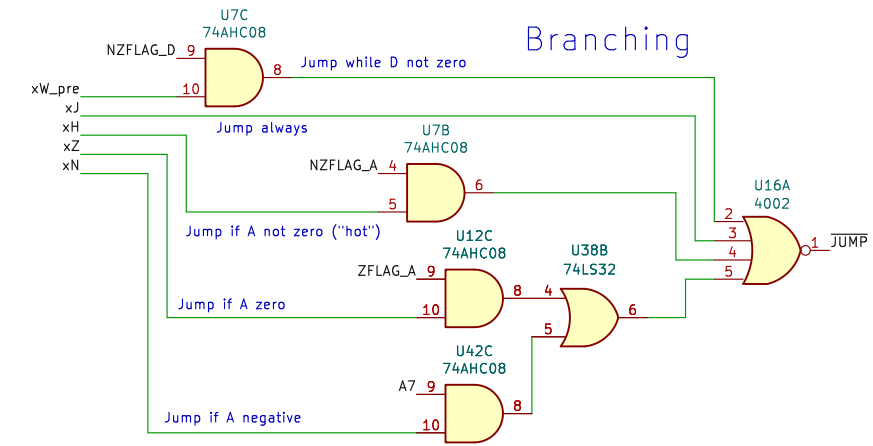
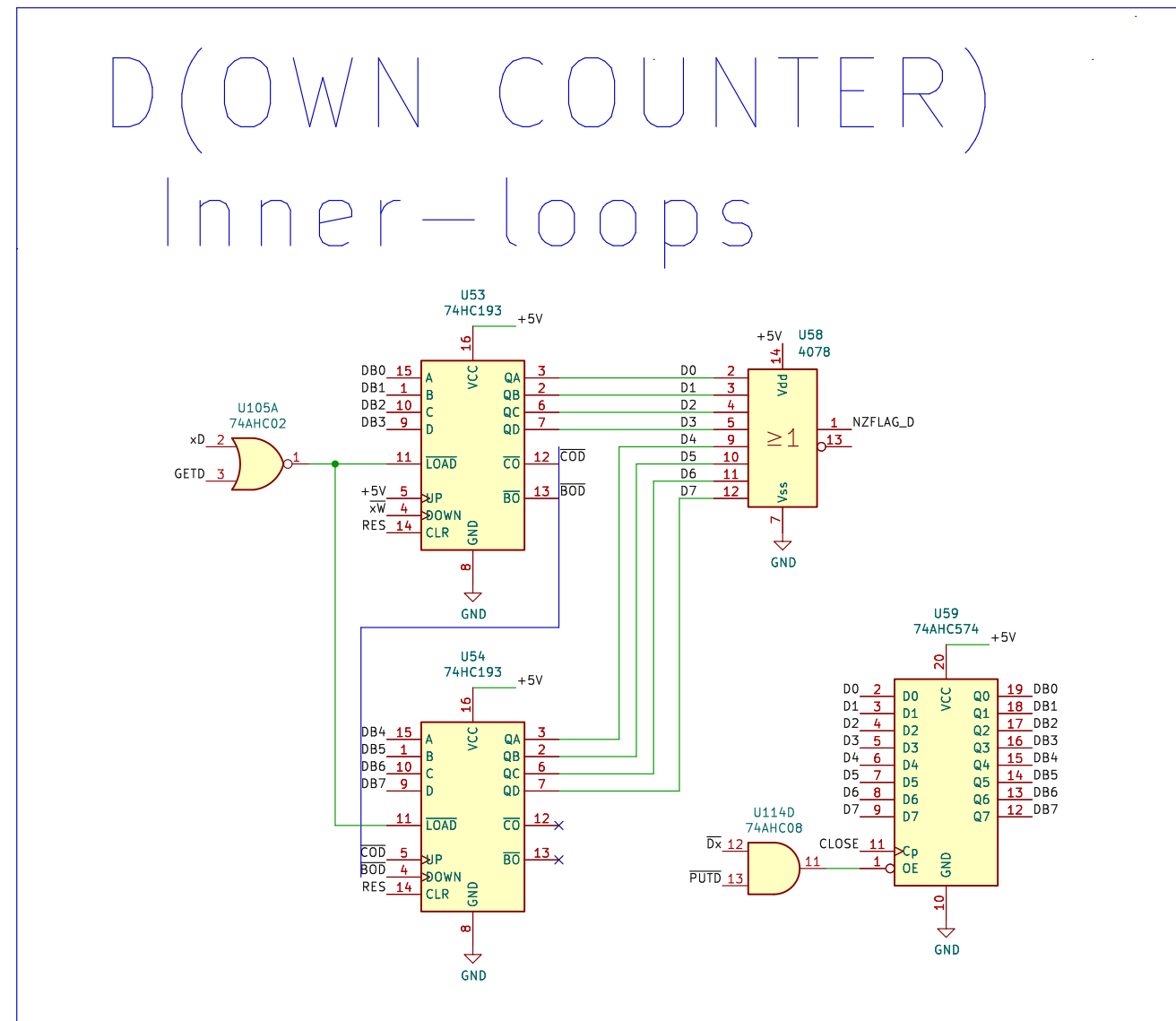
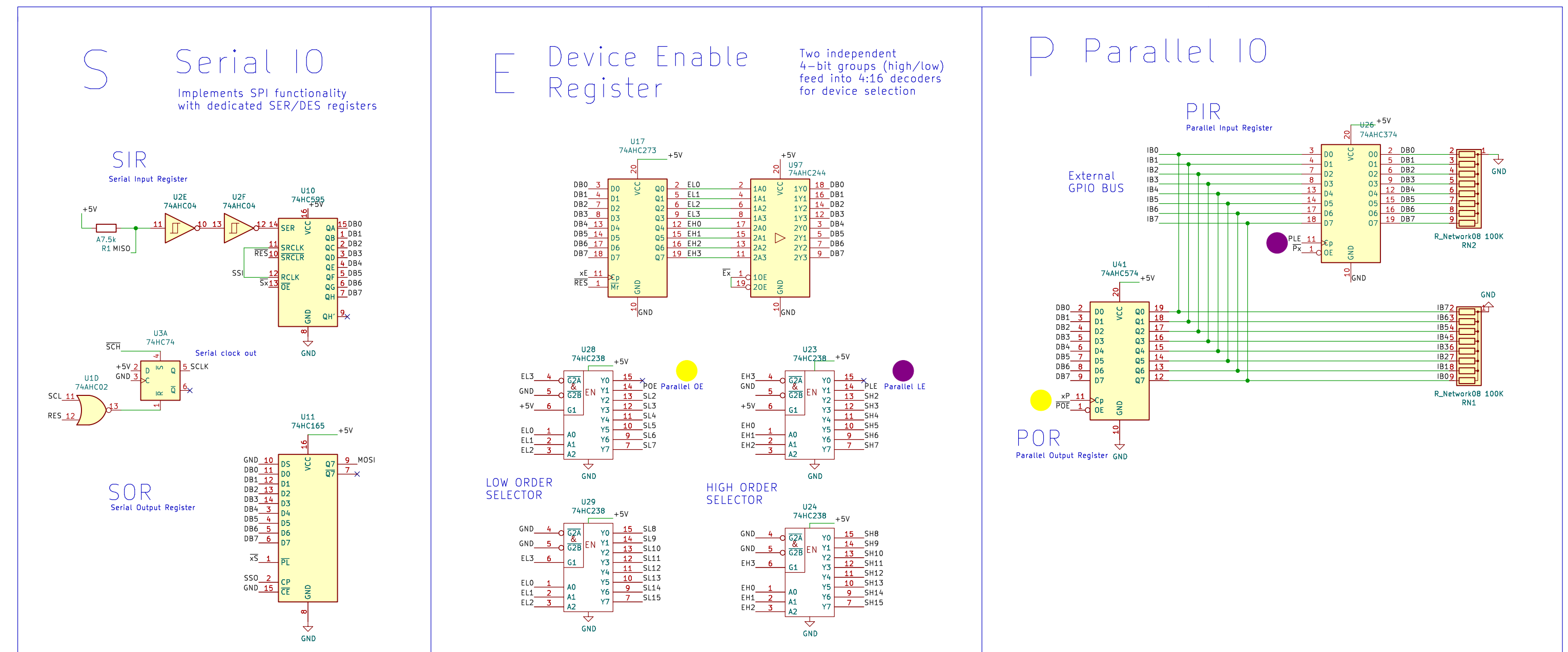
Rev. Myth



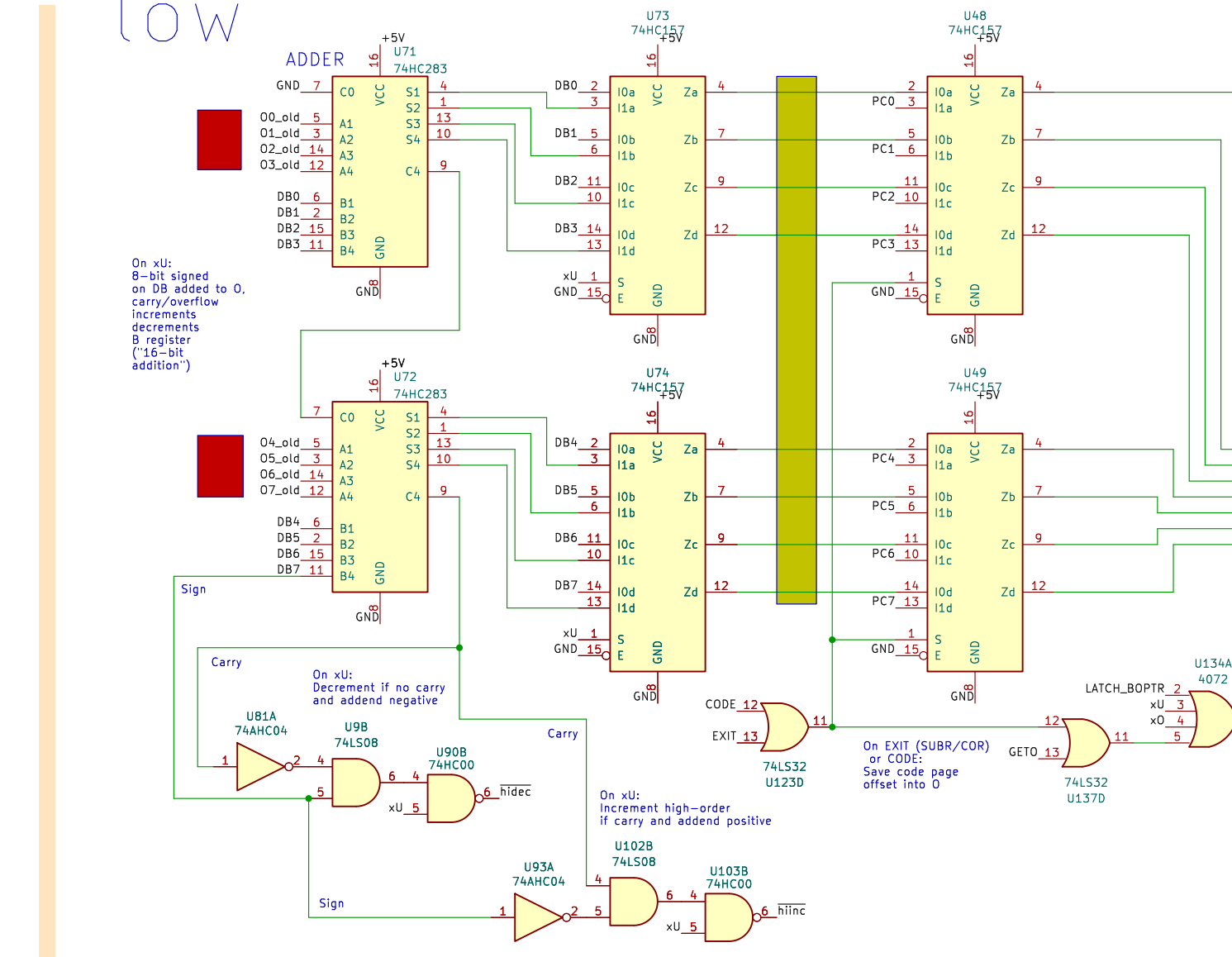
Instruction Decoder



Input/Output

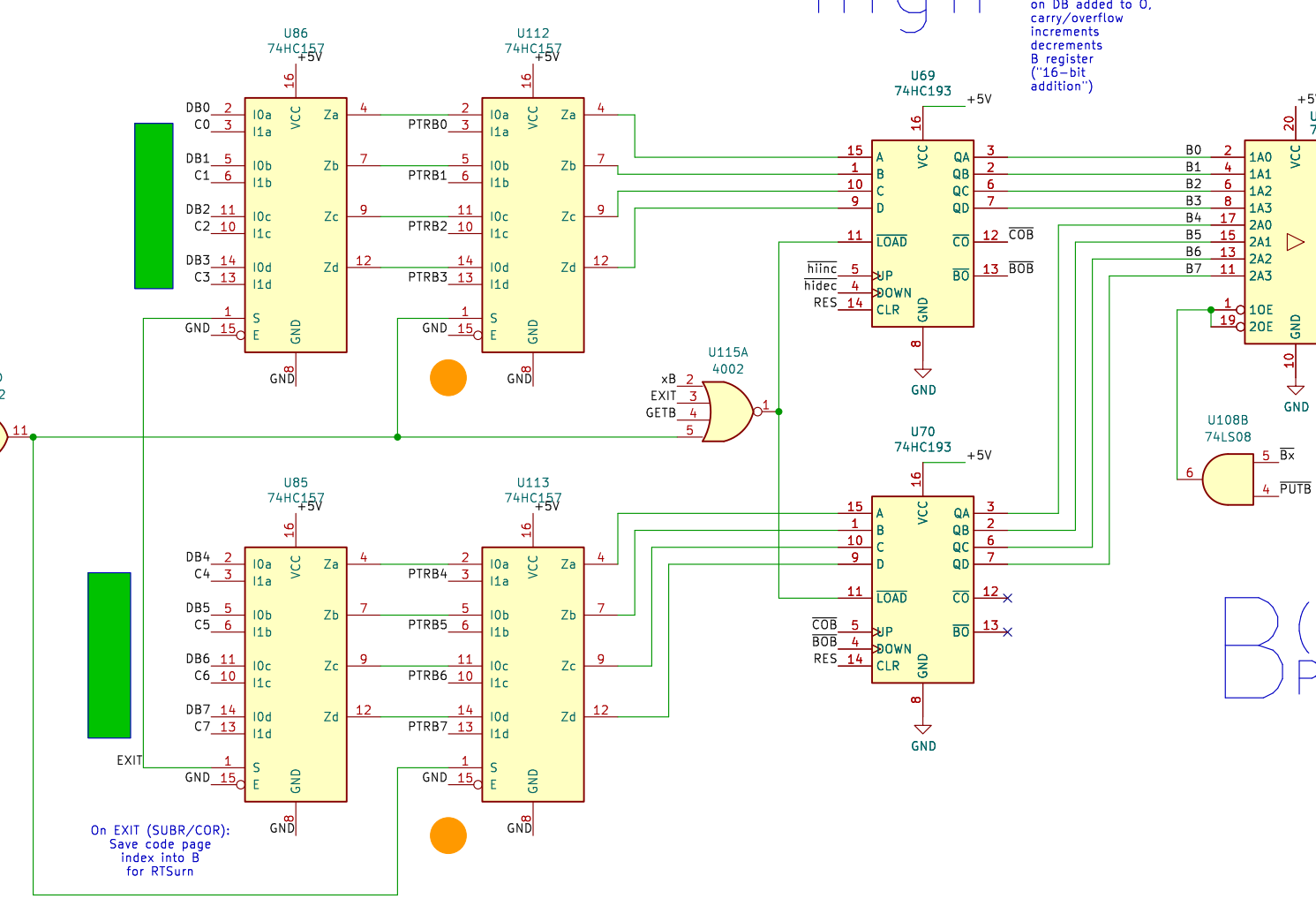


B:O Pointer low



- Saves offset of current instruction during call/trap/go
- Outputs return offset during RET
- Is the (only) implied address offset for memory read-write, with the corresponding page index stored in B (Base)
- Together with B forms 16-bit address pointers subject to the xU instruction
- The B:O pointer can be loaded from/saved into the B0-Pointers P1-4

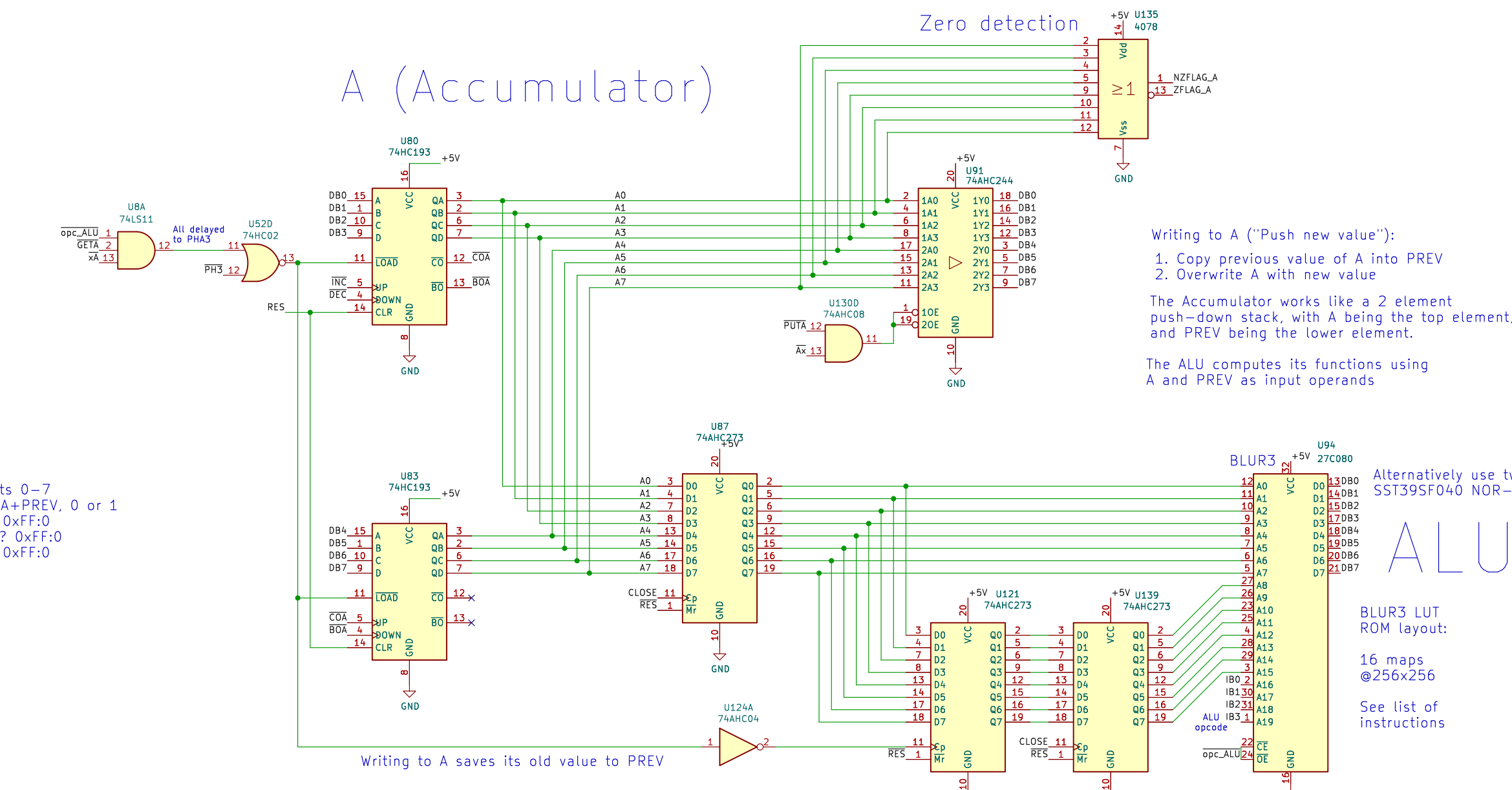
B:O Pointer high



B(Base) Page Index

- Saves page index of current instruction byte during call/trap/go
- Outputs return page index during RET
- Is the implied page index for memory read-write, with the corresponding byte offset stored in O (Offset)
- Together with O forms a 16-bit address pointers subject to the xU instruction
- The B:O pointer can be loaded from/saved into the B0-Pointers P1-4

A (Accumulator)



- Writing to A ("Push new value"):
1. Copy previous value of A into PREV
 2. Overwrite A with new value
- The Accumulator works like a 2 element push-down stack, with A being the top element, and PREV being the lower element.
- The ALU computes its functions using A and PREV as input operands

ALU

Alternatively use two 551595F040 NOR-Flash

BLUR3 LUT ROM layout:

16 maps
0:256-255

See list of instructions

PREV

This 2nd register is here, because when PREV copies the value of A this new value must not appear to the ALU until the instruction has finished "CLOSE". It would otherwise change the ongoing ALU operation.

O (Page Offset Register)

Register at CLOSE xU instruction address would feedback

B0-7 goes to PC module latched during RET

{PTR[7..0]}

{DB[7..0]}

B0 Pointer Registers

B1 Pointer Registers

B2 Pointer Registers

B3 Pointer Registers

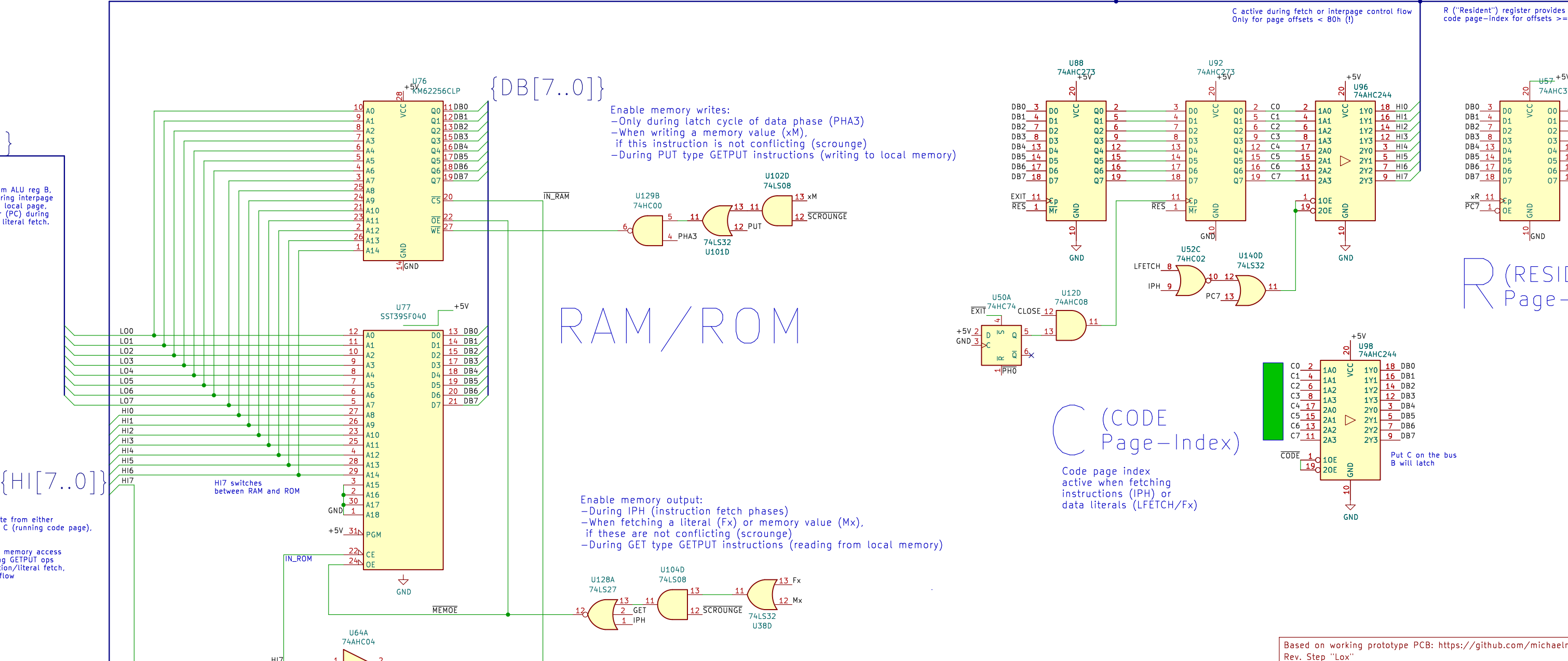
B4 Pointer Registers

B5 Pointer Registers

B6 Pointer Registers

B7 Pointer Registers

RAM/ROM



C (CODE) Page-Index

Code page index active when fetching instructions (PI) of data literals (LETCX/Fx)

Put C on the bus & wait