

Sheet: Memory and I/O
RAM, ROM and UART
File: MemoryIO.sch

Sheet: Addressing
PC, Address Register and Jump Logic
File: Addressing.sch

Sheet: ALU and Data Registers
ALU, A and B registers
File: ALU_DataRegs.sch

Sheet: Instruction Decode
Instruction register, Microsequencer, Decode logic and demultiplexers
File: IR_Decode.sch

Sheet: Analog
Clock generation, reset generation, bypass capacitors, misc. components
File: Analog.sch

These are the schematics for the CSCv0n8 CPU.
See <https://github.com/DoctorWkt/CSCv0n8/> for details.

Warren Toomey

Sheet: /
File: Schematic.sch

Title: CSCv0n8 CPU

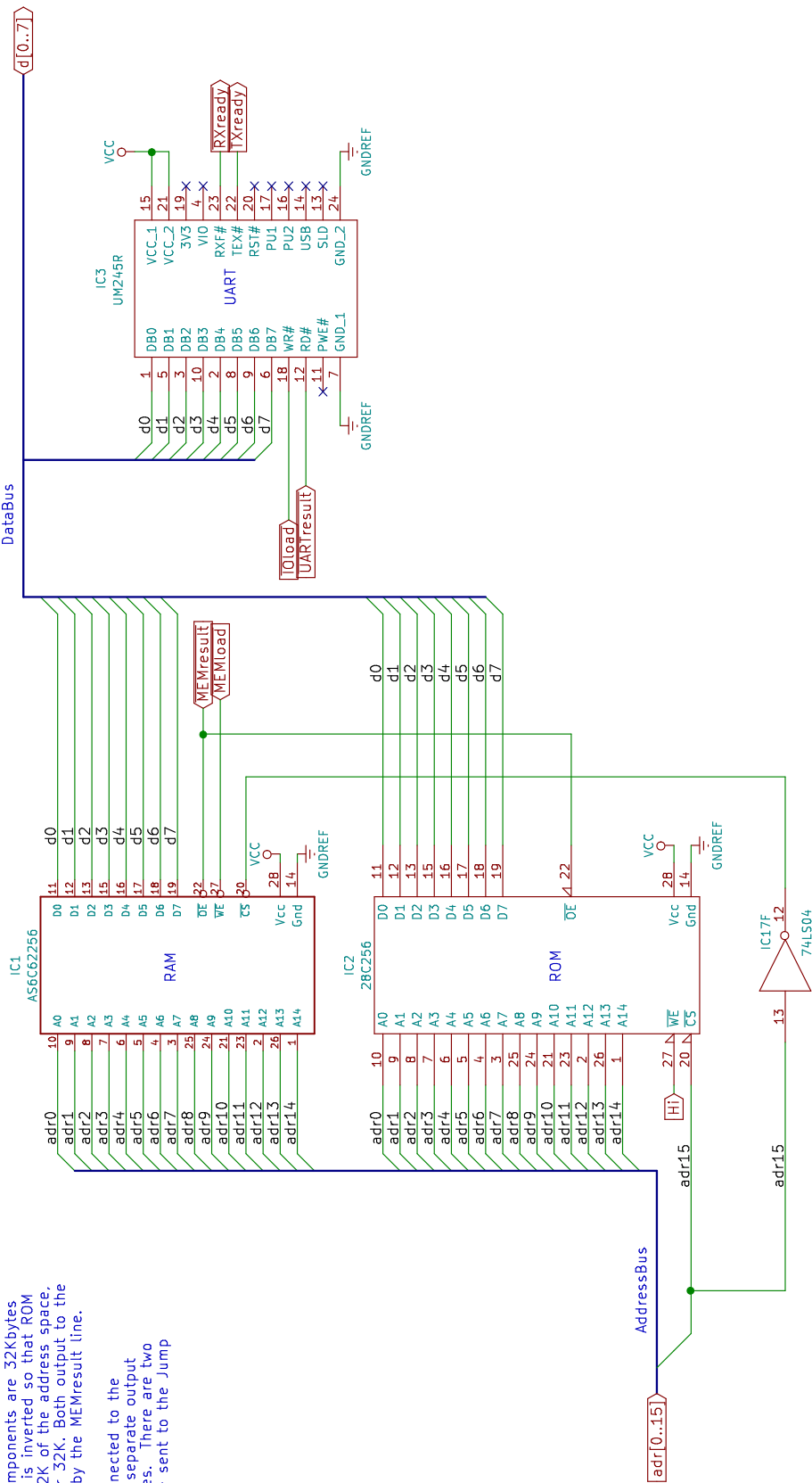
Size: A4 Date: 2019-04-02

KiCad E.D.A. kicad 4.0.7-e2-637661ubuntu18.04.1

Rev: 1.2
Id: 1/6

The RAM and ROM components are 32Kbytes each. The adr15 line is inverted so that ROM occupies the lower 32K of the address space, and RAM in the upper 32K. Both output to the data bus, controlled by the MEMresult line.

The UART is also connected to the data bus, and it has separate output and input control lines. There are two ready lines which are sent to the Jump logic.



Sheet: /Memory and I/O/
File: MemoryO.sch

Title:

Size: A4 Date:

KiCad E.D.A. kicad 4.0.7-e2-637661ubuntu18.04.1

Rev:

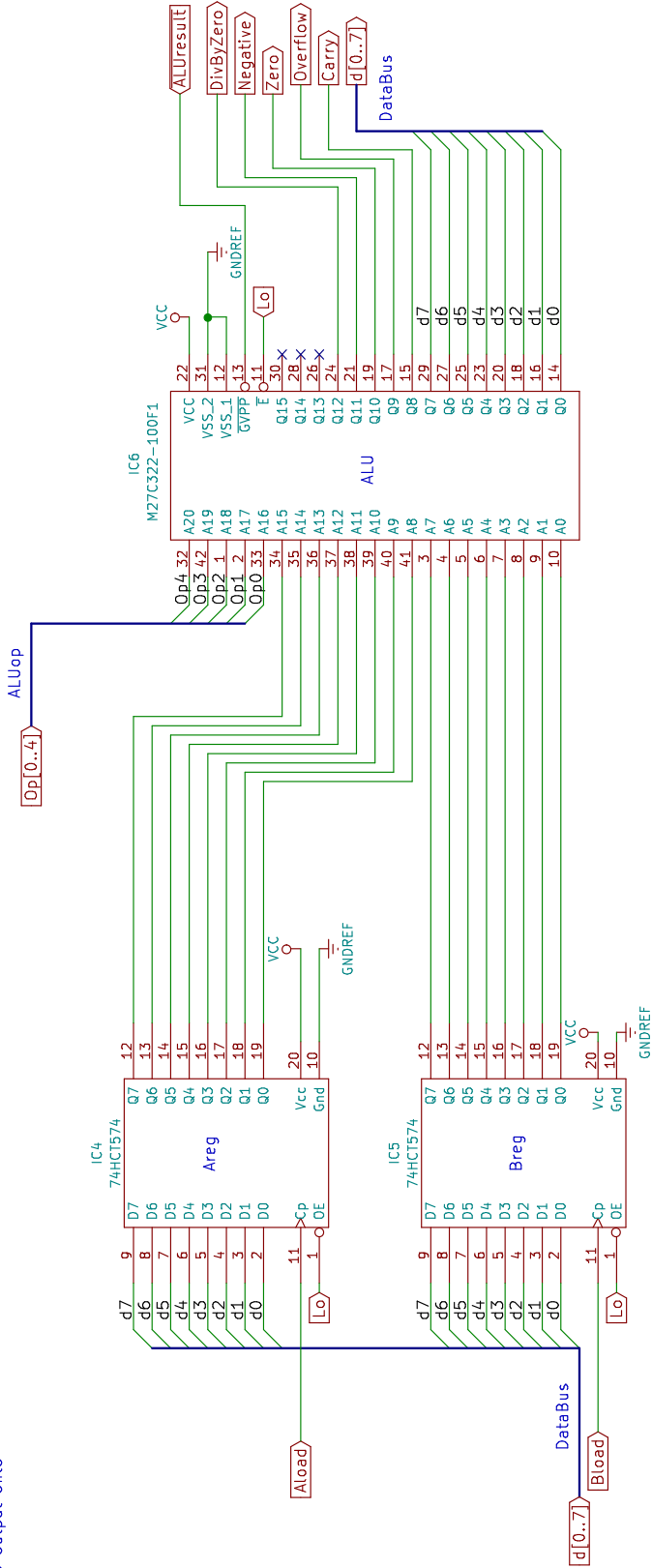
Id: 2/6

The A and B registers can load from the data bus. Their outputs are directly connected to the ALU.

The ALU also receives the 5-bit ALU operation from the instruction decoder logic.

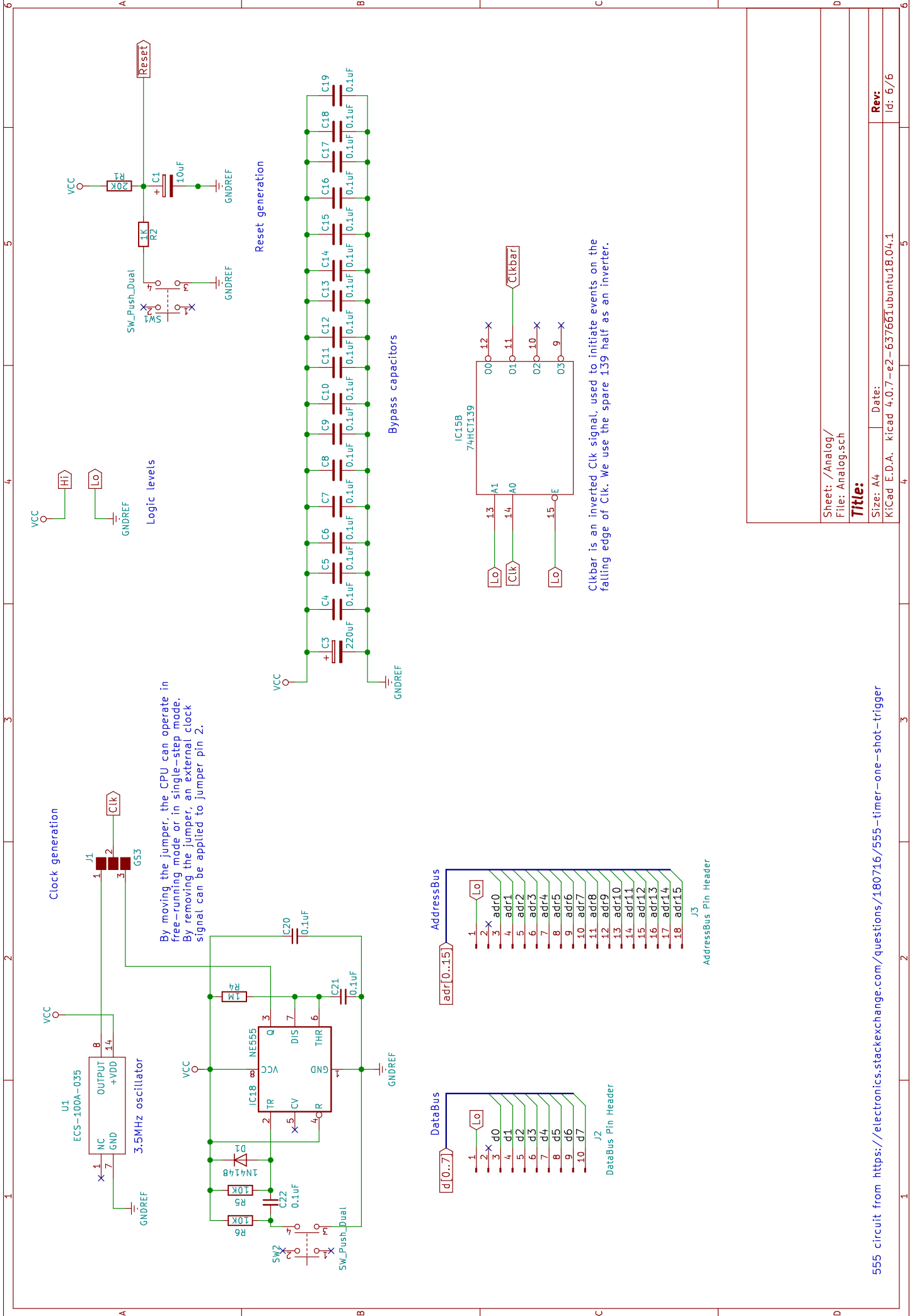
The ALU outputs the 8-bit result plus five flags that describe the type of output.

The ALUresult control line enables the ALU output onto the data bus.



Normally, the microsequence counter increments, but if `uSreset` goes low it will reset back to value zero. All microsequences have their "zero" microinstruction to load the instruction register from the data bus (ROM or RAM) and to increment the instruction register.





Sheet: /Analog/
File: Analog.sch

Title:

Size: A4 Date:

KiCad E.D.A. kicad 4.0.7-e2-637661ubuntu18.04.1