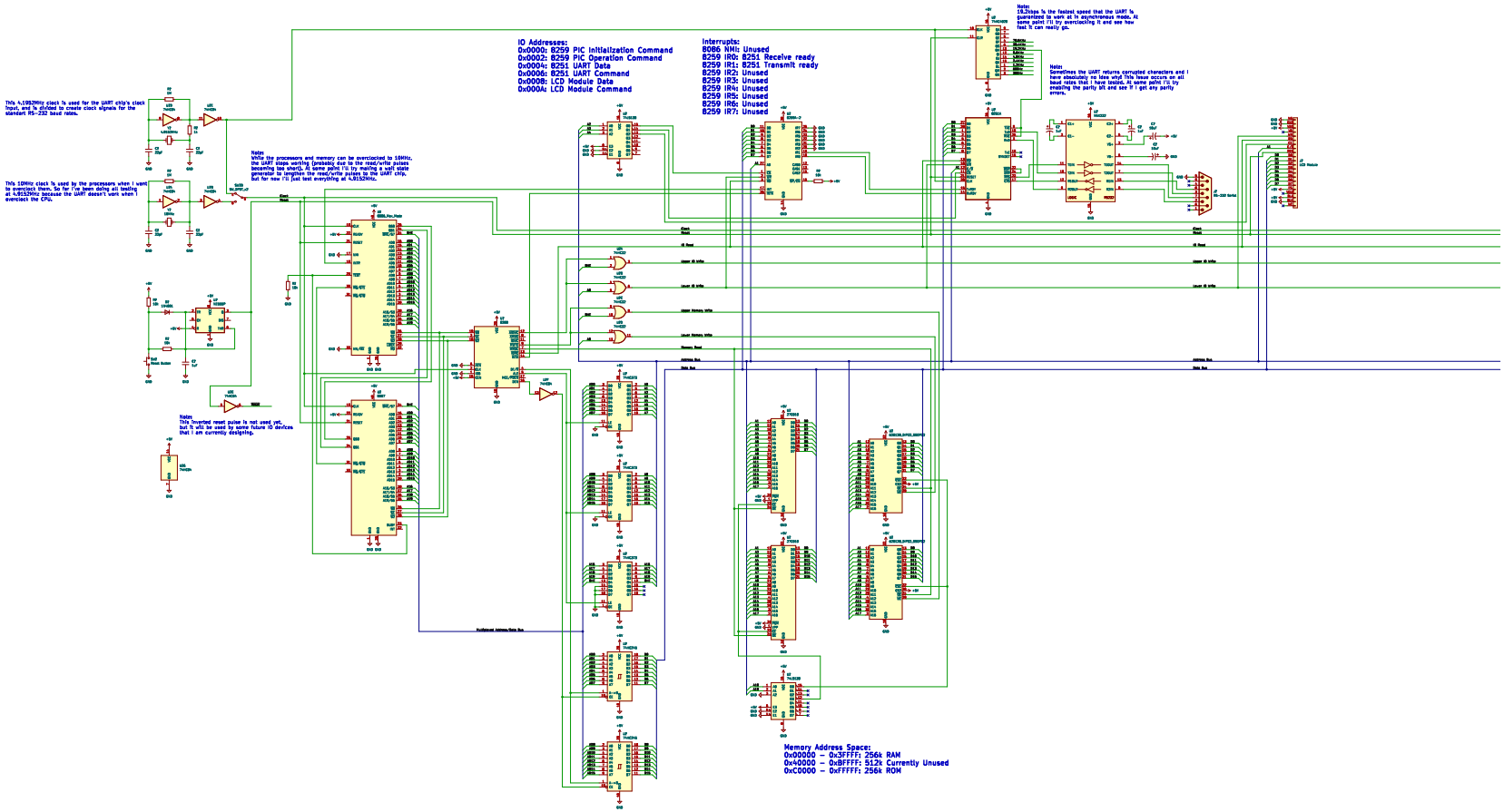
This 4.1952MHz clock is used for the UART chip's clock input, and is divided to create clock signals for the standard RS-232 baud rates.
This 10MHz clock is used by the processors when I want to overclock them. So far I've been doing all testing at 4.9152MHz because the UART doesn't work when I overclock the CPU.
Note:
While the processors and memory can be overclocked to 10MHz, the UART stops working (probably due to the read/write pulses becoming too short). At some point I'll try making a wait state generator to extend the pulses going to/from the UART chip, but for now I'll just test everything at 4.9152MHz.
Note:
This inverted reset pulse is not used yet, but it will be used by some future IO devices that I am currently designing.
IO Addresses:
0x0000: 8259 PIC Initialization Command
0x0002: 8259 PIC Operation Command
0x0004: 8251 UART Data
0x0006: 8251 UART Command
0x0008: LCD Module Data
0x000A: LCD Module Command
Interrupts:
8086 NMI: Unused
8259 IR0: 8251 Receive ready
8259 IR1: 8251 Transmit ready
8259 IR2: Unused
8259 IR3: Unused
8259 IR4: Unused
8259 IR5: Unused
8259 IR6: Unused
8259 IR7: Unused
Note:
19.2kbps is the fastest speed that the UART is guaranteed to work at in asynchronous mode. At some point I'll try overclocking it and see how fast it can really go.
Note:
Sometimes the UART returns corrupted characters and I have absolutely no idea what this issue occurs on all baud rates that I have tested. At some point I'll try enabling the parity bit and see if I get any parity errors.
Memory Address Space:
0x00000 - 0x3FFFF: 256k RAM
0x40000 - 0x8FFFF: 512k Currently Unused
0xC0000 - 0xFFFFF: 256k ROM