Bit 0: Is it on? (InstructionReg)

Bit 1: Ram is ON/OFF (Ram Sel)

Bit 2: Ram is Being Read if 1. Write if 2 (RamLD)

ucodeData Bits (3-23)

Bit 3: Accepting input from RAM if 0, and from ALU if 1 (RlMux)

Bit 4: is regIn listening? inRegEnabled (RlEn)

Bit 5/6: Which Register to write to, (RISel)

Bit 7: is DMX, after regBank ON/OFF? (RO0En)

Bit 8/9: Select which Register to take data from (RO0Sel)

Bit 10/11: Which output of DMX receives data (RO0DMX)

Bit 12/13: Select which register to take data from (RO1Sel)

Bit 14: small DMX, after regBank which outpute receives data(RO1DMX)

Bit 15: is ALU listening? (ALUen)

Bit 16/17/18: what operation will ALU perform? (ALUop)

Bit 19: Accepting input from PC(1) or RO1DMX(0)(AddrMux)

Bit 20/21/22: which inputs of 2 big MUX's to forward (PCMuxMux)

Bit 23: is PC listening for updates? (PCEn)

Bit 24: if 1, next instruction is next ucodePtr++, if 0, reading from (instructionReg)

Bit 25: if 1, resets ucodePtr register to 000 for next fetch cycle

st(y), x // at location that Y is pointing to store value of regX

ld x, (y) // load regX with value at location that Y is pointing to