Init State:
shift register Q0–7 contain the address that the avr will listen on, set at boot
PortIORequest pulled up by mcu internal resistor
IO_add_comp is held low, flip flop passing ioreq out to wait
A0–7 + IORQ are non–matching

IO sequence:
1. cpu brings iorq rd/wt low with matching port address on bus
2. portIOrequest goes low after matching addr[2:7] with sr[2:7] and gets latched onto waitline with next clk pulse
2.5 avr falling edge intterrupt trigger on portIOrequest
3. avr checks RD and WR lines to determine what request is being made
if read:
4. avr sets databus to output and sets value
5. avr then asserts the busreq line to essentially stall the cpu in tristate after the databus has been read
6. avr tristate decoding logic by brining IO_add_comp high , wait goes high from pullup on cpu card, latch is set high on next clk pluse
7. cpu reads data lines
8. cpu sees asserted bus req at begining of last m cycle of read
9. cpu tristates its pins
10. avr Poll on busack, proceed after it goes low
11. avr tristate the databus
12. avr reenable address decoding by brining IO_add_comp low
13. avr release busreq
if write:
4. avr sets its data pins to input and collects value
5. avr tristate the data pins
6. avr tristate decoding logic by brining IO_add_comp low, wait goes high from pullup on cpu card, latch is set high on next clk pluse

io op complete

Atmega2560 + Cystal

U2A
ATMEGA2560–16AUR

Clock ctrl stuff here maybe?
mabe a seperate optional board?

ICSP Input Port

IO Wait state latch

RC2014 Bus

Atmega2560 Port Export

soft Addr

FP Addr

Sheet: /	
File: rc–duino.kicad_sch	
Title:	
Size: A3	Date:	Rev:
KiCad E.D.A. kicad 6.0.2+dfsg–1	Id: 1/1