0b i2 i1 o2 o1 o0 i0 x x # 0b 17 16 15 14 13 12 11 10 # o0 = XOR(X-sig,x1) = a3 # i0 = XOR(X-sig,a2)reg_names_OUT = ["CLEAR INT STATUS","B","r0","r1","r2","r3","r4","r5"] reg_names_IN = ["U","_Uout","r0","r1","r2","r3","r4","r5"] V<u>C</u>C **VCC** V<u>C</u>C V<u>C</u>C V<u>C</u>C VCC Input demux U3 74LS299PC U1 74LS299PC U2 74LS299PC Output demux U8 SN74L<u>S2</u>38 SN74LS138N #RETI QH #Bo QH /QG QH /QE #r0_out H/QF H/QH E/QE H/QH D2 D5 F/QF F/QF F/QF #r1_out C/QC D3 D3 G#2B Y3 #r2_out /QA D/QD D/QD B/QB #r3_out B/QB B/QB !RESET !RESET !RESET CLK CLK CLK CLR# GND CLk CLK CLR# GND CLK #r4_out GND GND Ţ GND GND GND GND GND VCC VCC **VCC** U4 74LS299PC U5 74LS299PC **Duncatron** bus 74LS299PC 79 RESET 78 !RESET 77 D0 swap pass swap pass QH G/QG QH D4 D2 D4 On-On-Off MSK-12C03_DS MSK-12C03_DS On-On-Off H/QH E/QE H/QH E/QE D2 H/QH ___D3 F/QF F/QF F/QF /QA D/QD D/QD D/QD _D1 B/QB B/QB !RESET !RESET MC_RESET CLK CLK B/QB !RESET CLK Uout_BUS Uin_BUS #T_HL CLK 10 GND 70 **GND** 69 D8 호 GND 를 GND 66 D11 65 D12 _D13 A8/9/10 is not output onto bus in v1.0 of memory map Could use a wire on backplane from SEI pin to A8/9 or frontplane to A10 VCC V<u>C</u>C VCC **VCC NAND** NOR SN. 742574 74LS74 MICRO_CLR SN. 74LS74 74LS74 U18 XD74LS00 XD74LS02 !RESET !RESET !#RETI 4A 1B 54 I7 53 FLAG_C #MICRO_CLR #RETI SEI 4B 4B INT_D0 #MICRO_CLR 28 MICRO_CLR 52 FLAG_Z !INT_D0 MMREAD 29 2B 3A 3Y MMWRITE 30 !#RETI 3B RESET Uout_BUS 1PIN_CONNECTOR PIN1 0X,0Y GND 3Y GND In interupt INT ₹ GND ≢ GND — GND MEMMAP <u>₹</u> GND 44 A7 43 A8 42 A9 41 GND VCC V<u>C</u>C U11 10 LED BAR INT_REQUEST U14.1 10-Pin-Resistor-Network #RR is low when #RETI is low SN74LS377NG4 DM74LS245N or #RESET is low BREAKOUT SEI to a header for future memory mapping ("Control register") OE# Bit should be pulled down so interupts #RR2 is low when 2Q is set GND are disabled by default! Means can't interupt or RESET is high any important init code. OR USE FREE BUS CONN GND C10 C9 = 0.1uf C11 C14 C13 0.1uf 0.1uf 0.1uf C12 C11 = 0.1uf 0.1uf 0.1uf 0.1uf 0.1uf 0.1uf GND This board implements six general purpose hardware registers r0-r6 The instruction register (and interrupt logic)
Interrupts run PUSH_PC+1 and INT in sequence. Opcode can be hardwired with solder jumpers
Transfer register also provided for moving data between hi and low 8-bit buses. TITLE: **REG** schematic REV: 1.0 Can also perform a SHIFT RIGHT if X is enabled (HIGH) and LOAD is HIGH $(T_EN=1 \text{ and } \#T_IO=0)$ Possess LED bank instruction register (not output selection?) Company: Your Company Sheet: 1/1 **EasyEDA** Drawn By: Duncan Scott Date: 2024-01-03

ALU format:

0b xxxx a3 a2 a1 a0 # Reg select format: