

Registers: A, B, C?, D?, Result, Instruction, Memory Address, Button State, Video, Flags

RAM: 64K

Video: 640x480

Bus Width: 16 Bit

For EEPROM and RAM

Mem Address Register

Control logic counter increments on clk, not CLK

Instruction req
holds Operand
AND opcode?

Reset Circuit

Clock



Maybe a Stack?

RAM

ALU

VGA

PC

Control

Result

Buttons

IRQ

Programming
Switches

Address
to Data
Converter

Output?

Speaker

And their immediates

Opcodes: ADD, SUB, MUL, DIV, AND, OR, NAND, NOR, XOR, XNOR, NOT, CMP, JNE, JL, JE, JGT, JLT, JZ, LDA, LDB, LDC, LDD, RDB, VGA, STA, STB, STC, STD,

Maybe also
JGE / JLE
L>= UC=

And Immediates

MOV, HLT, NOP, JMP, SQRT, POW, SHL, SHR, INC, DEC, OUT, SPK, JSR, RTN, RTL, RTB

And Immediates

TSIO Init.s

RSI	RDI	RSA	RSA	RSI	RSI
-----	-----	-----	-----	-----	-----

Rotate

Parts List (Actual Numbers Required)

~~3x 555 Timer~~ X
~~1x 1M Pot~~ X
~~1 - S139~~ X
~~5x LS161~~ X
~~7x 28C256~~ X
~~4x LS181~~ X
~~16x LS245 (Get A Few More)~~ X
~~2x 71256SA15TPGI RAM chips~~ X
~~8x LS157~~ X
~~1x Character LCD~~ X
~~27x LS173~~ X
~~1x SN76489 AN~~ X
~~1x LM386N-4~~ X
~~1x 4MHz Oscillator~~ X
~~3x 100uF 25V Electrolytics~~ X
~~2x 100nF (ceramic)~~ X
~~1x Speaker~~ X
~~1x NC Pushbutton~~ X
~~1x NO Pushbutton~~ X
~~1~ 10K Pot~~ X
~~2x SPDT Toggles~~ X ** Maybe use a pushbutton for programming mode/clock switching*
~~3x 74LS00~~ X
~~3x LS32~~ X
~~13x LS04~~ X
~~3x LS02~~ X
~~3x LS86~~ X
~~4x LS4078~~ X
~~1x 5-6A PSU~~ X
~~46x Green LED~~ X
~~22x Yellow LED~~ X
~~70+ Blue LED~~ X
~~100+ Red LED~~ X
~~238x Current Limiting Resistor~~ X
~~LOTS OF WIRE~~ +
~~LOTS OF BREADBOARDS~~ +

~~1x 1M Pot~~ X 10k
~~1x LS181~~ X
~~3x 100uF 25V DIPIC~~ X
~~1x 100nF (ceramic)~~ X 0.1uF
~~3x LS32~~ X

Different Sound Chip?

Bypass Caps

~~PLA Plastic~~

~~Black Tape~~

~~*-Acto Knife~~

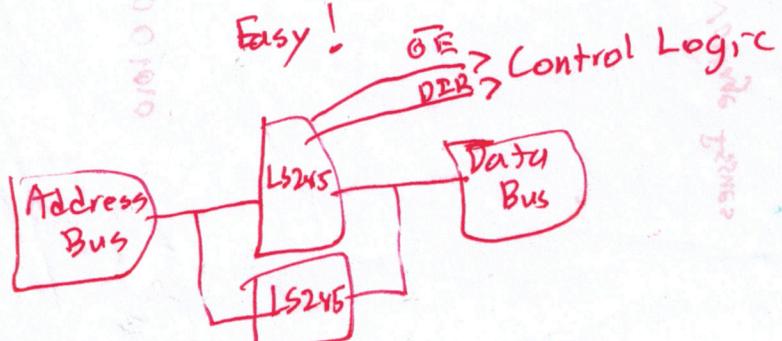
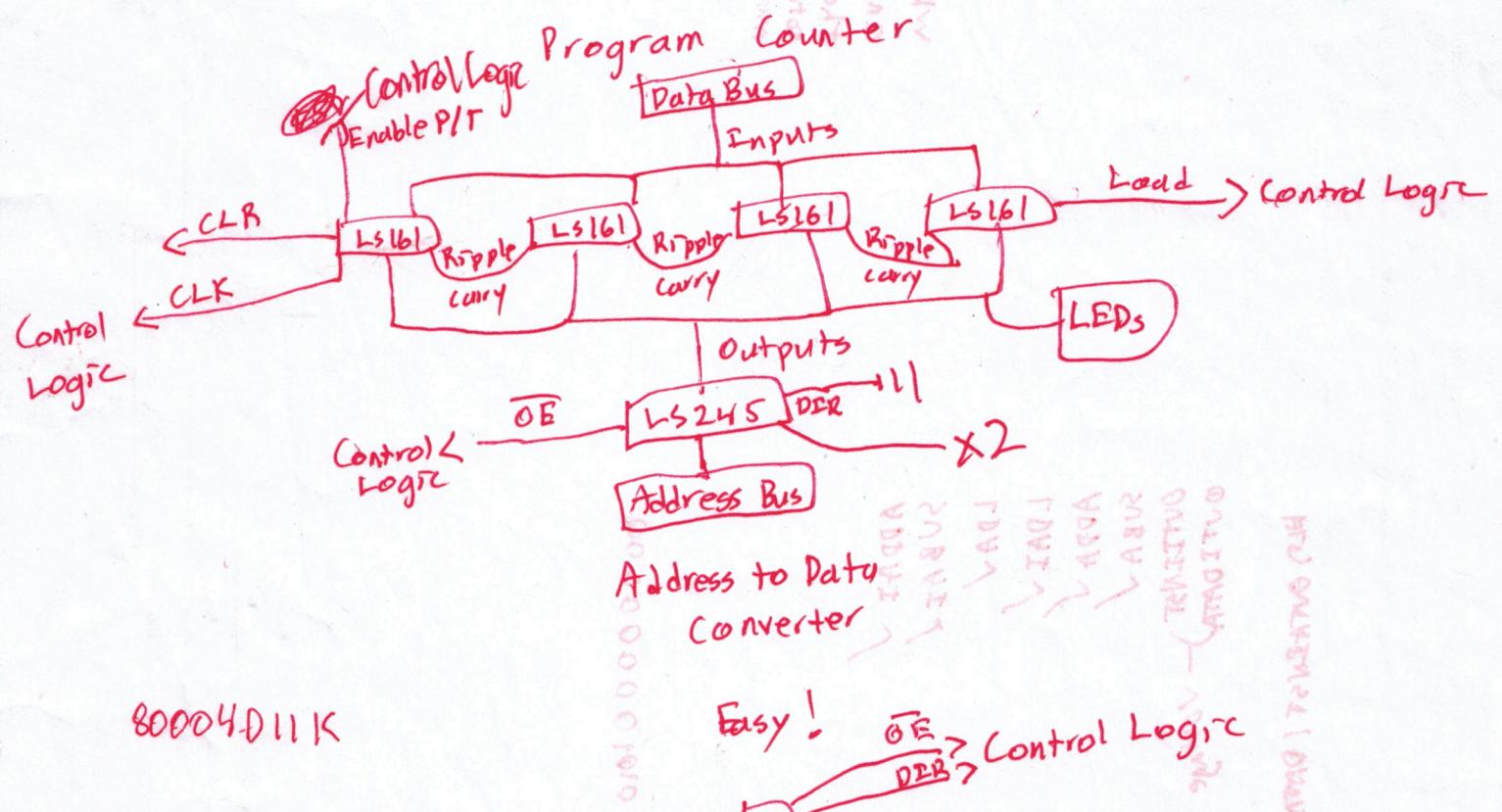
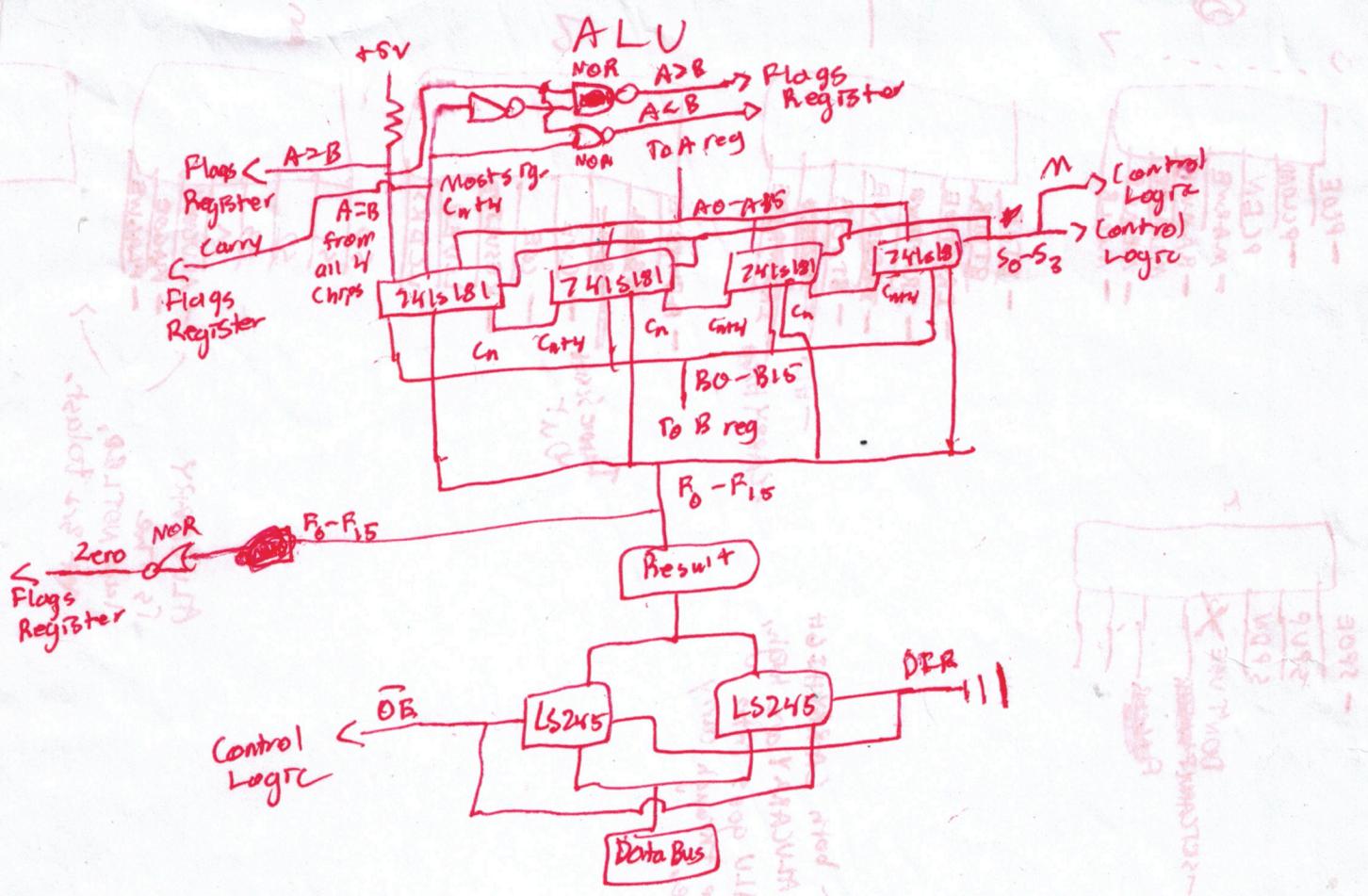
170 + 118.5 + 24 + 110 + 45 + 17 + 50

1uf or 0.1uf
1k 10k etc

10nf or 0.01uf

47uf 10k
Wire
Multi
Strippers
Power Adapter

Order when they're back in stock.



Hook Flags Reg to Control logic

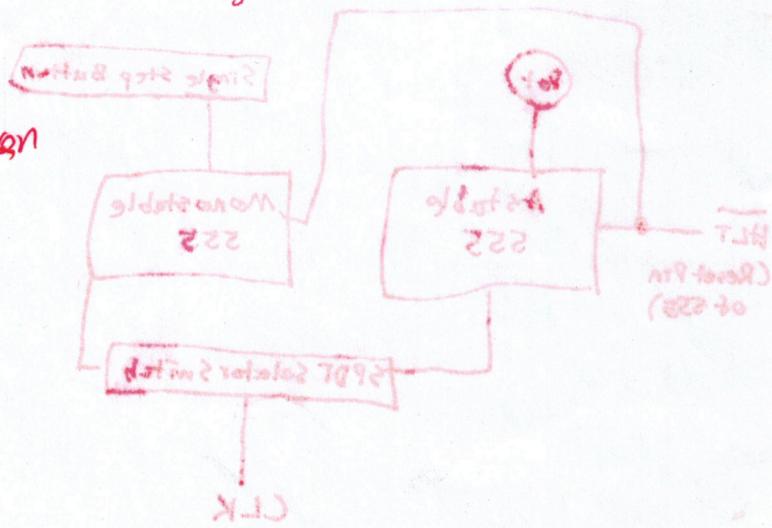
Clear FR with both reset AND control logic.

Hook Up Address Bus

Hook Up Buttons/IRQ

~~Invert~~ Control lines on

Resistors on Buses

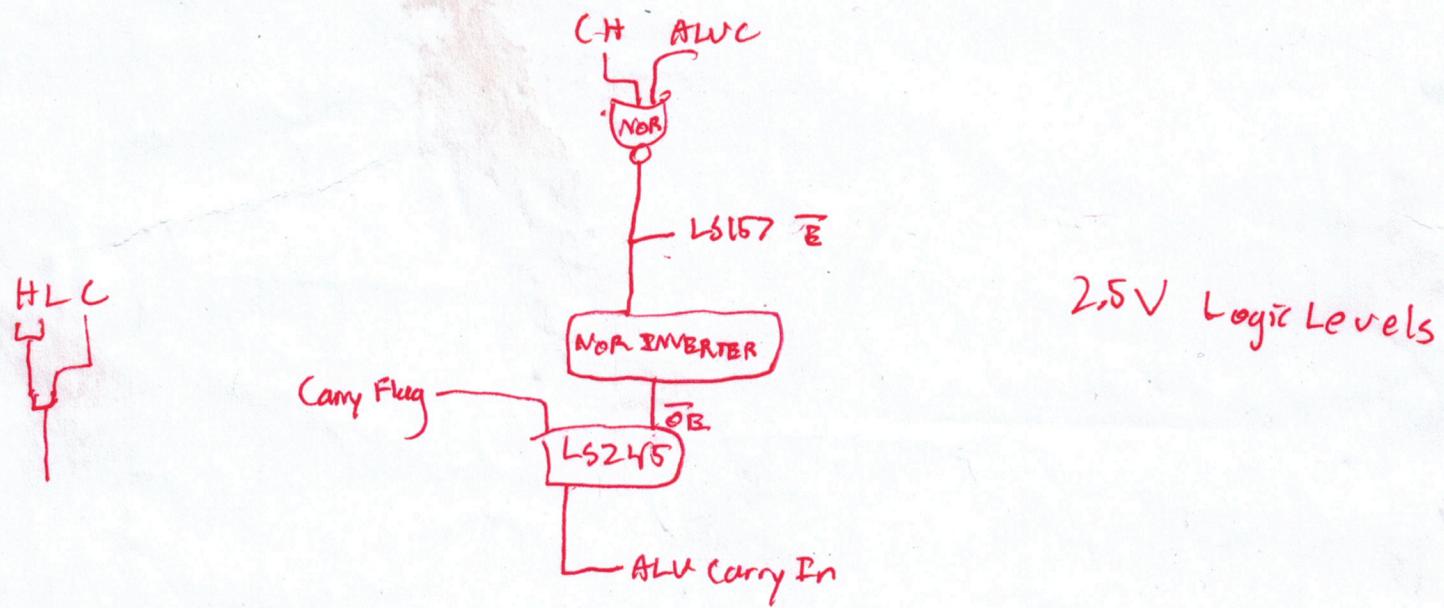


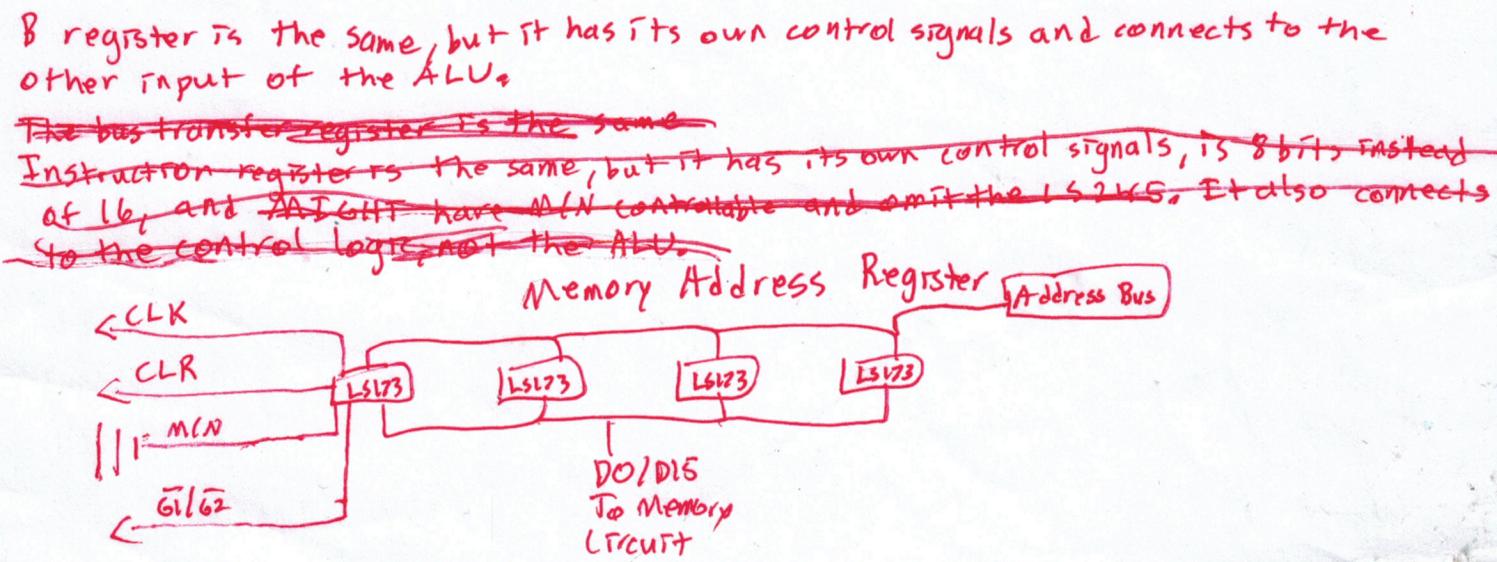
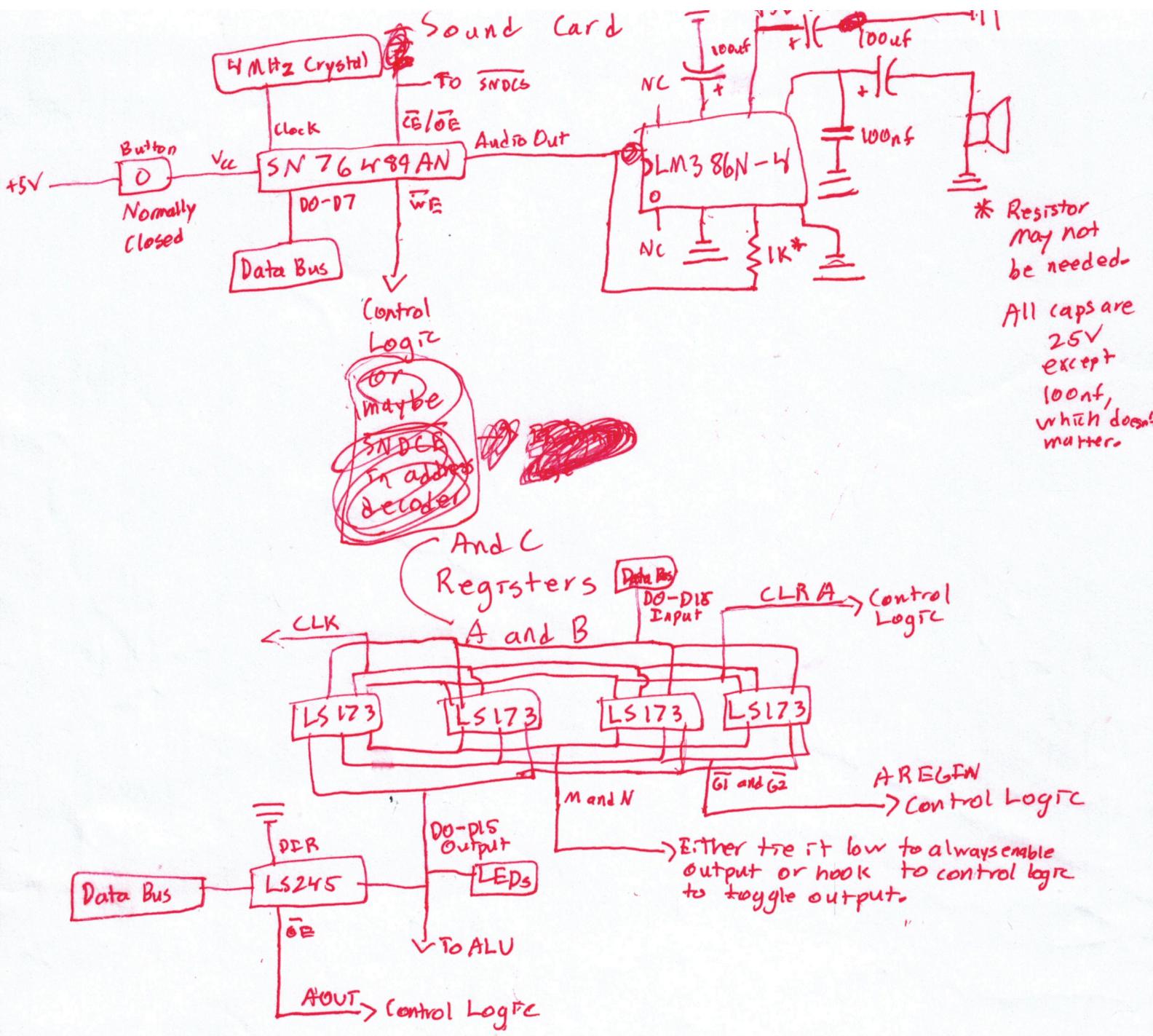
0	JMP	16384
16384	JMP	0

+IN17) UJA

H	H	L
H	L	L
L	H	L
L	L	H

NOR

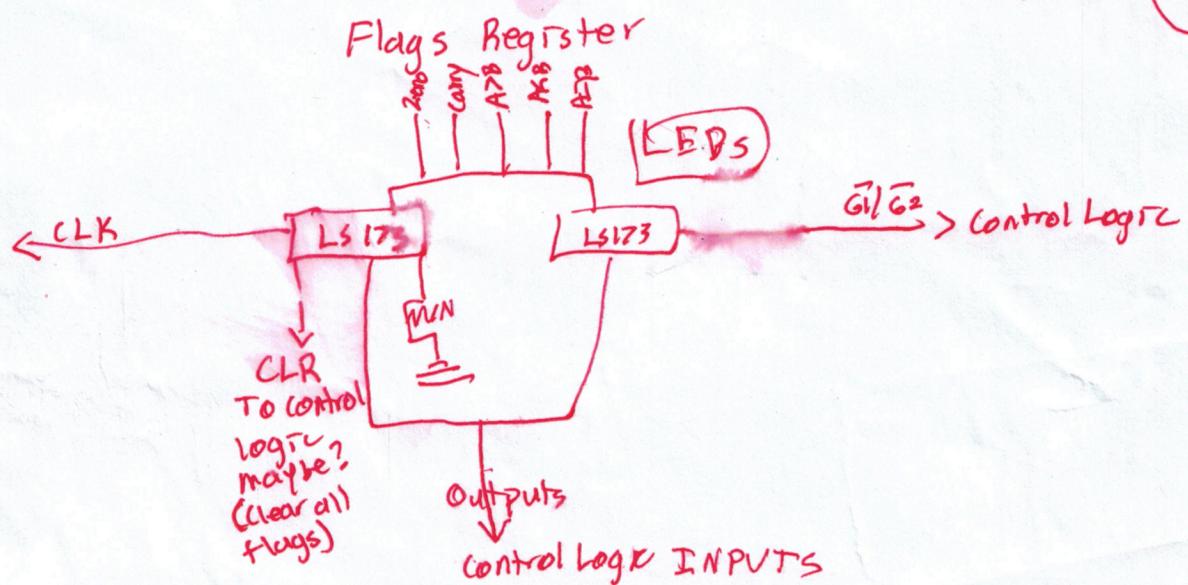
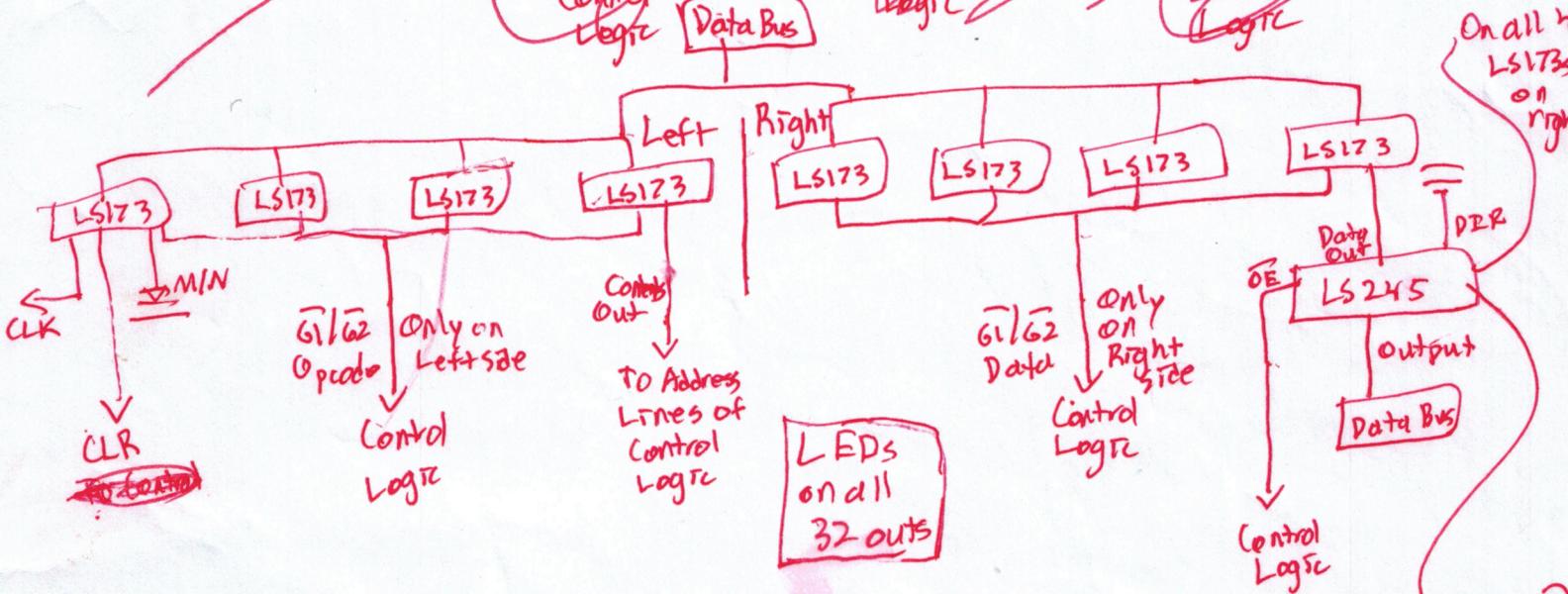
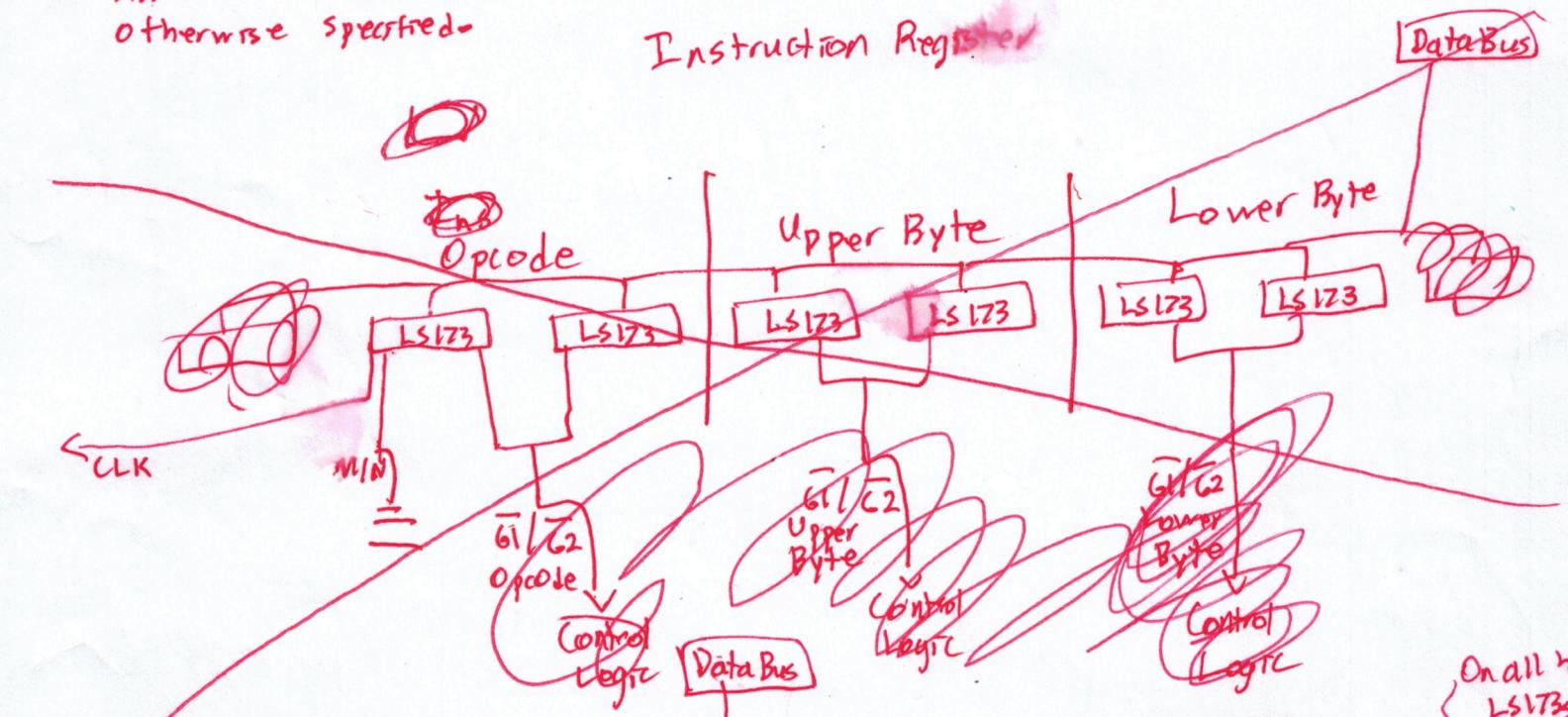




Registers Part 2

Assume that all control wires are repeated on all chips, unless otherwise specified.

Instruction Register

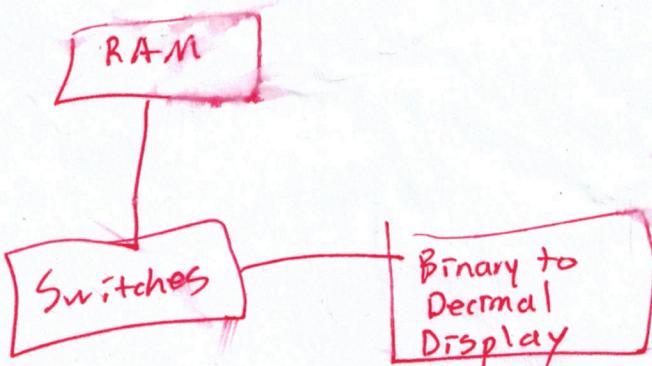


Bank 1



Bank switching based off (A15)
and (A14)

Bank 0

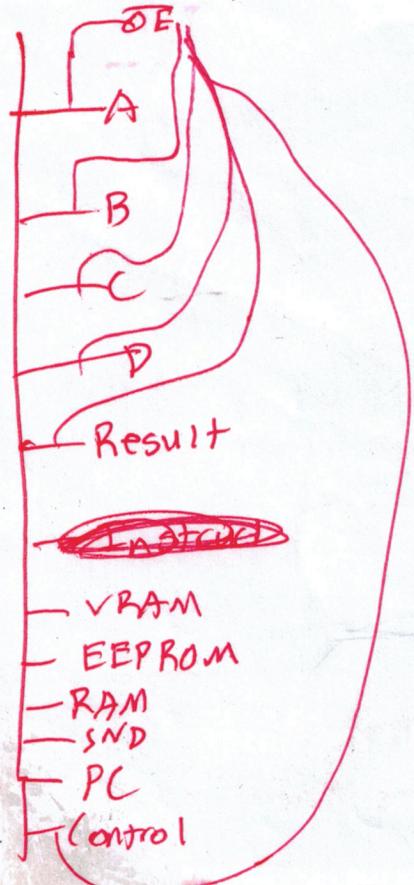


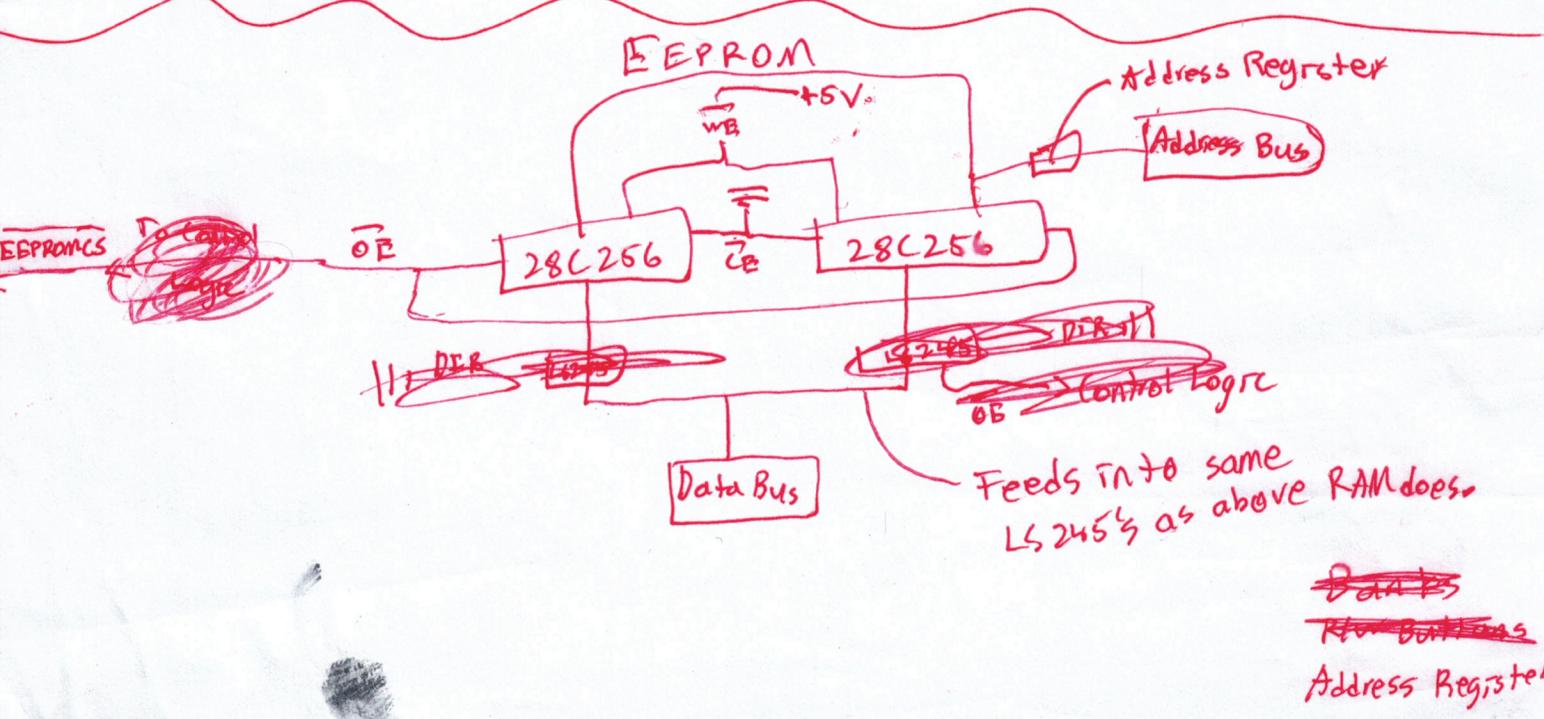
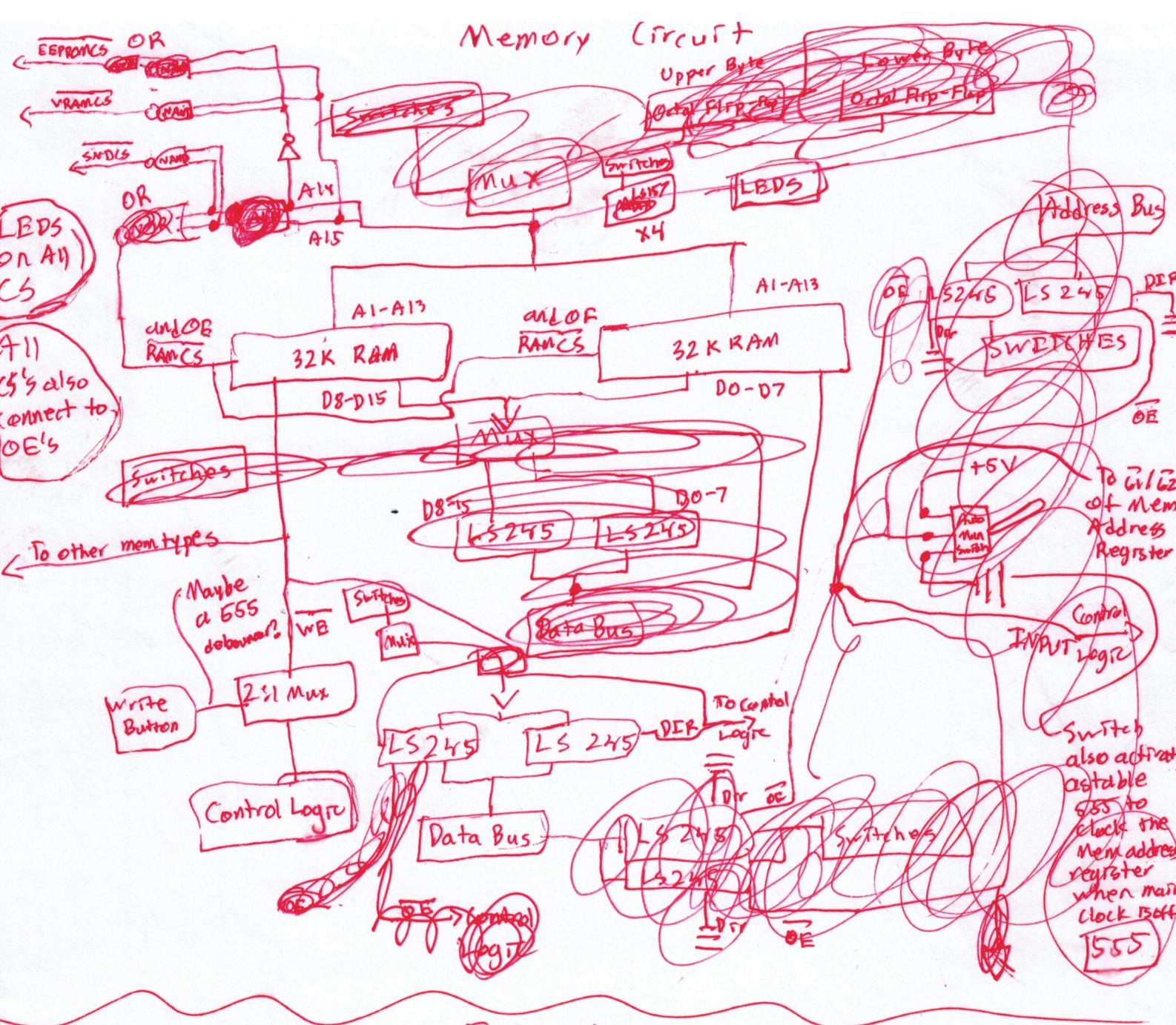
	RAM	EEPROM	VRAM	SND
0	0	1	2	3
A15	0	0	1	1
A14	0	1	0	1

\overline{OE} low
and
 $A-7B$
low

\overline{OE} to high
Stop everything from putting
out D/A. Set 'n'
mem A's to m and
mem D's to m.

Switches





Control Logic (Oh No!)

Control Lines

HLT - Halt Clock

ALUCTRL0)

ALUCTRL1) - ALU Mode Select

ALUCTRL2)

ALUCTRL3)

ALUCTRLM - ALU Arithmetic/Logic Select

ALUOE - ALU Output Enable

PCLOAD - Load Program Counter

PCOE - Program Counter Output Enable

→ ADCOE - Address to Data Output Enable

→ ADCDIR - Address to Data Direction

BUTOE - Button Output Enable

BUTIRQENABLE - Enable IRQ (1), Disable IRQ (0)

BUTIRQFF - Must Be Pulsed With BUTIRQENABLE.

LCDRS - LCD Register Select

LCDENABLE - Write Data Bus to LCD

MEMDIR - Memory In or Out (Direction); Setting to In Takes Mem Off Bus

MEMWE - Memory Write Enable (Including Sound Card WE)

MARLOAD - Load Memory Address Register

FLGCLR - Clear Flags Register

FLGIN - Load Flags Register with Current ALU Flags

OPCIN - Instruction Register Load Opcode

DATAIN - Instruction Register Load Operand/Data

INSTOE - Instruction Register Operand Output Enable

CLRA - Clear A Register

AIN - Load A Register

AOUT - Output Contents of A Register

ALUOUTA - Connect/Disconnect A Register From ALU.

CLRB - Clear B Register

BIN - Load B Register

BOUT - Output Contents of B Register

ALUOUTB - Connect/Disconnect B Register From ALU.

CLRC - Clear C Register

CIN - Load C Register

COUT - Output Contents of C Register

ALUOUTC - Connect/Disconnect C Register From ALU.

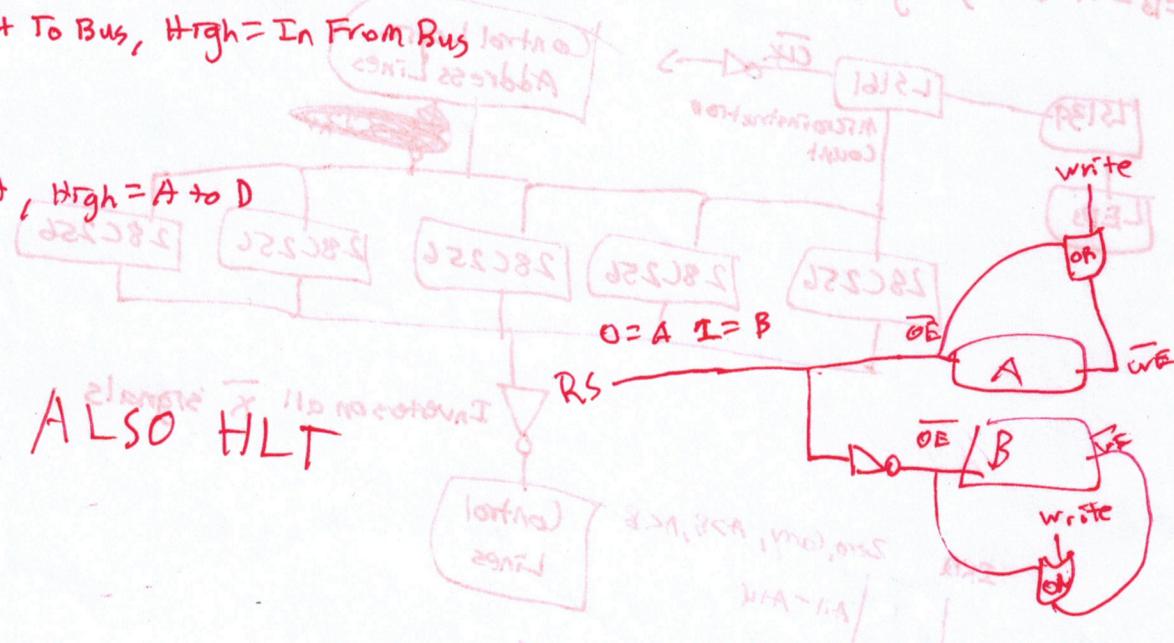
PCCLK - Program Counter Clock

PCINC - Increment Program Counter

Total: 34 Control Lines

PLOE
 PCLOAD
 PCENABLE
 MRSOURCE - Low = Bus, High = Panel
 MARWB
 RAMWB
 MEMDIR - Low = Out To Bus, High = In From Bus
 MEMOE
 BTDE
 BTDIR - Low = D to A, High = A to D

Button	CL	Result
0	0	0
0	1	0
1	0	0
1	1	1



ALSO HLT

AOE (To Bus)
 AWE
 AOE (To ALU)
 IRFLG
 FLAGOE
 FLAGWE
 S0
 S1
 S2
 S3
 M
 ALUCARRY
 RESULOE
 LCDEN
 LCDRS - Low = Inst, High = Data
 BDNOE
 IRQBN
 PRVPORT

Except AOB(Bus) and ADDR are shared by B
 A, B, LDIF are Low = Busto ALU, High = Reg to Bus

Maybe make these 1 line each
 Maybe remove?

ABWRITE
 ABSEL - Low = A, High = B
 28
 ABOE
 CWR
 COE

22 Lines

32

Control Logic #1

Address Lines

$OPCODE_8 \rightarrow OPCODE_{10}$ - Higher 8 Bits of Instruction Register Opcode

IRQ - Interrupt Request from Buttons

ZERO - Zero Flag

CARRY - Carry Flag

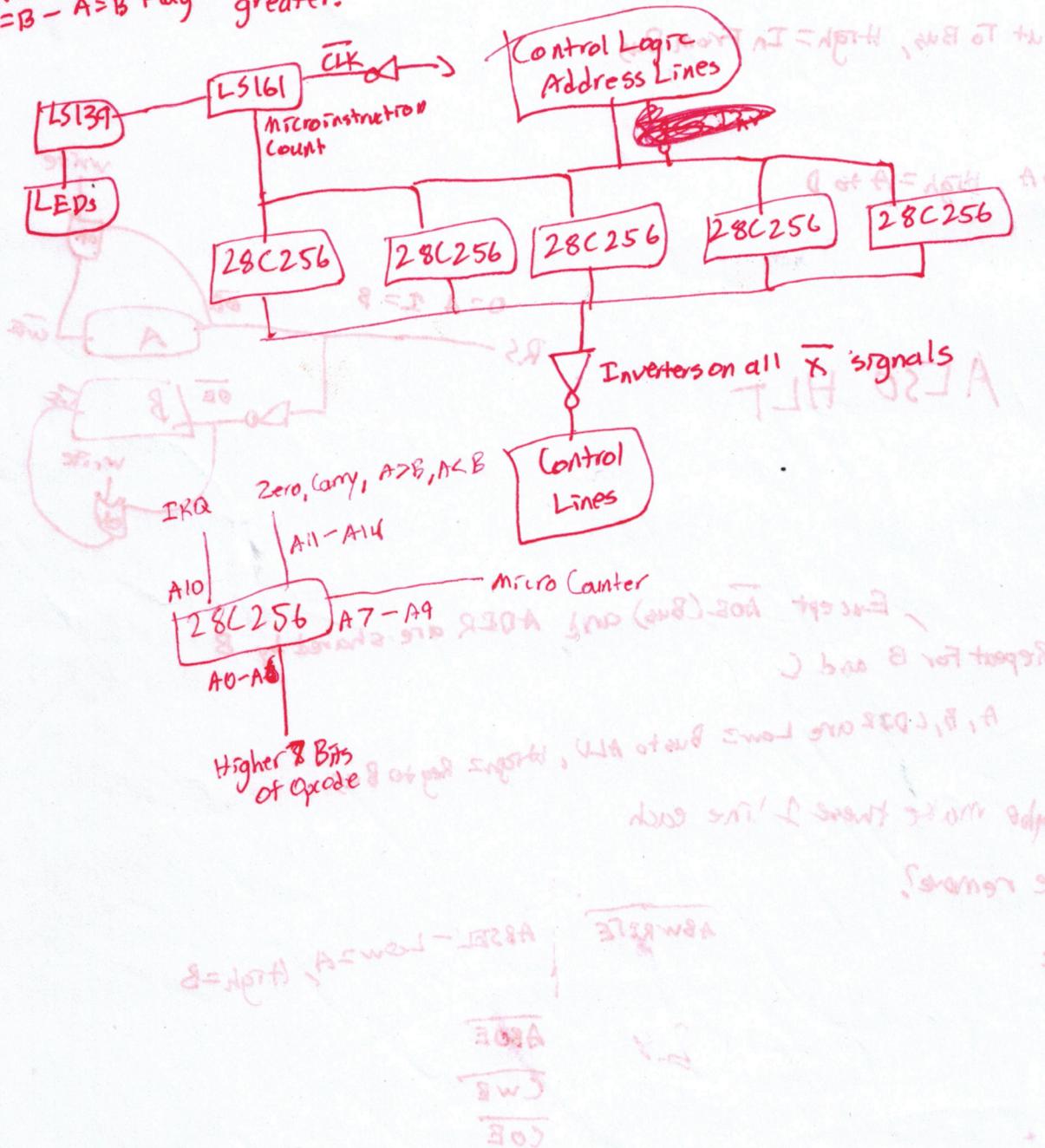
$A > B$ - $A > B$ Flag

$A < B$ - $A < B$ Flag

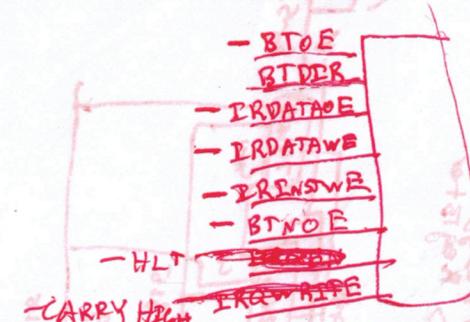
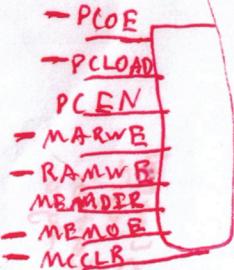
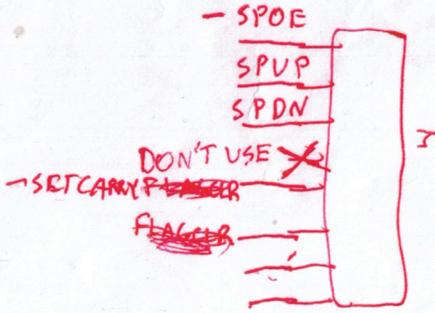
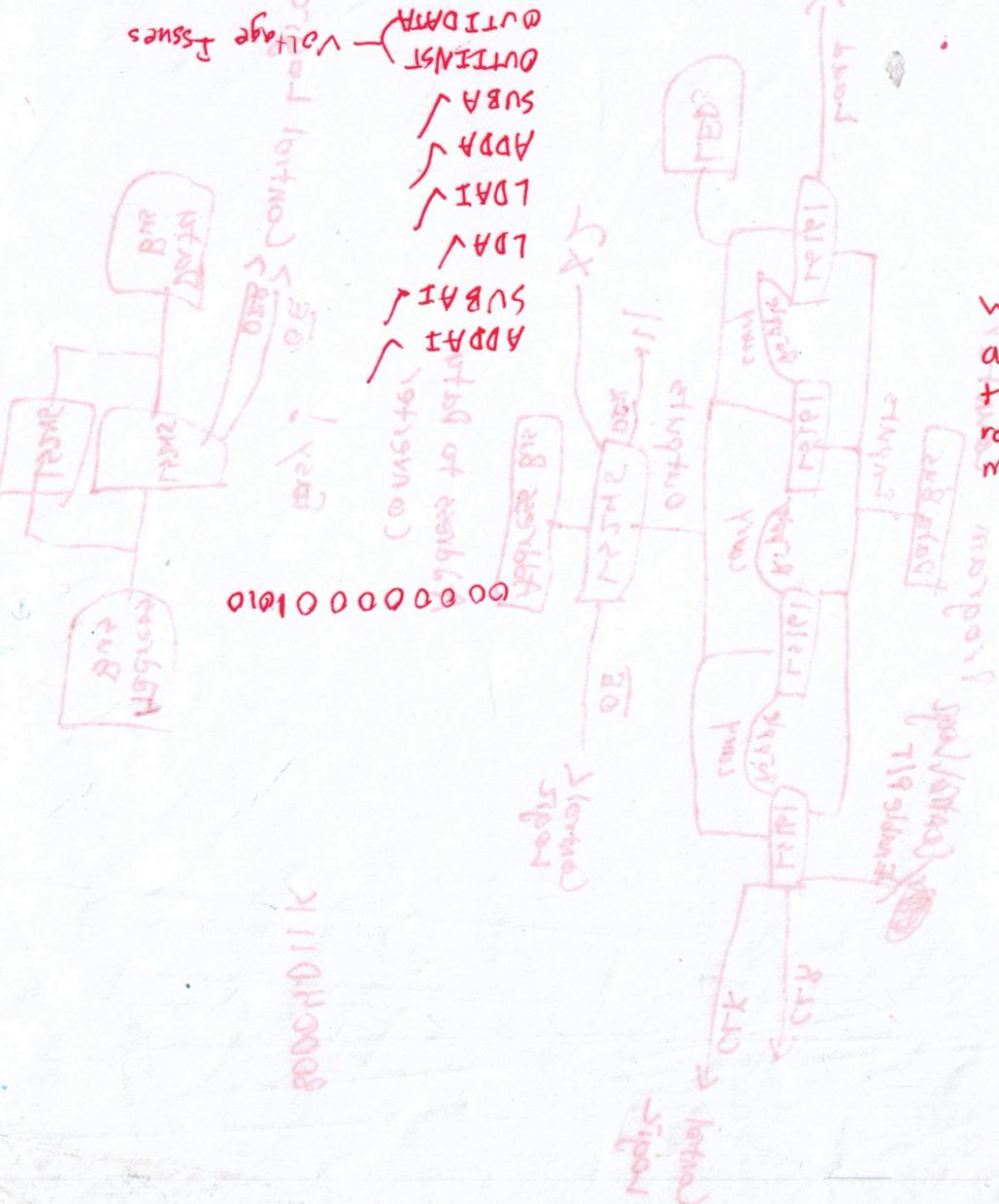
$A = B$ - $A = B$ Flag

If a isn't less than or equal to b , it has to be greater.

Total: 12 Address Lines

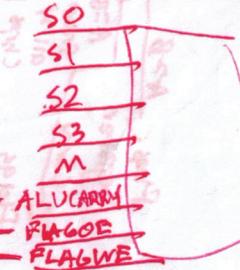


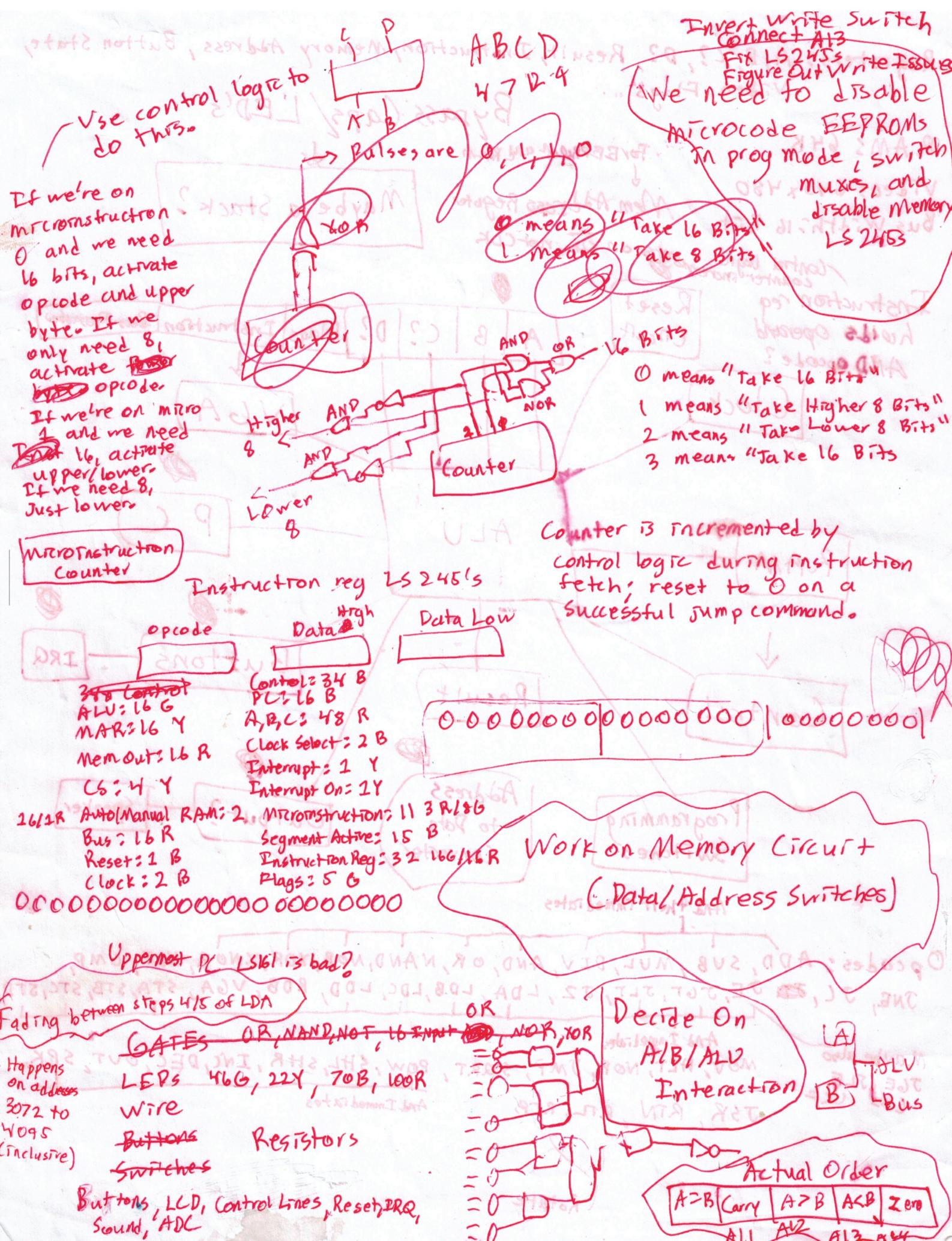
WORD OUTURNS 10 bits



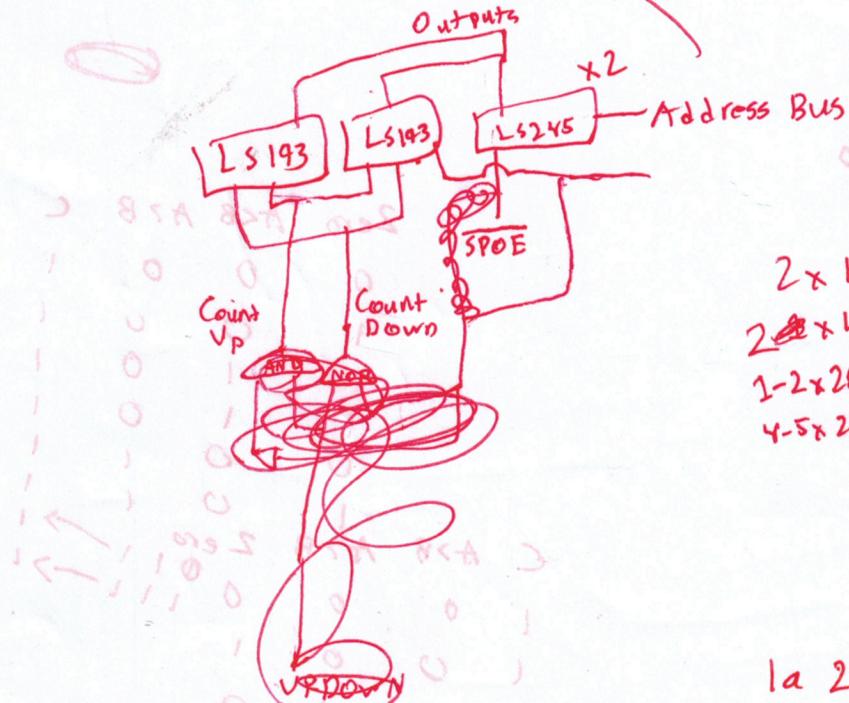
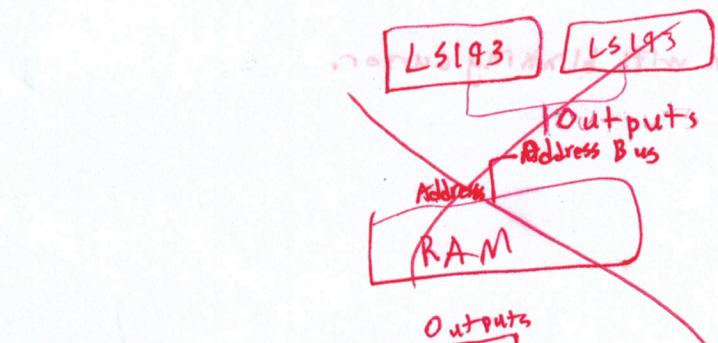
When both CARRYHIGH and ALUCARRY are high, the ALU goes into rotate through carry mode.

Ignore Scratch Out





Stack



Cats is at address 0 with a value of 5.

Address 5 contains a 6-

LDA #CATS \rightarrow LDA 0 \rightarrow LDA 5 \rightarrow V

LDA #Cats

A = 5

LDA Cats

A = 6

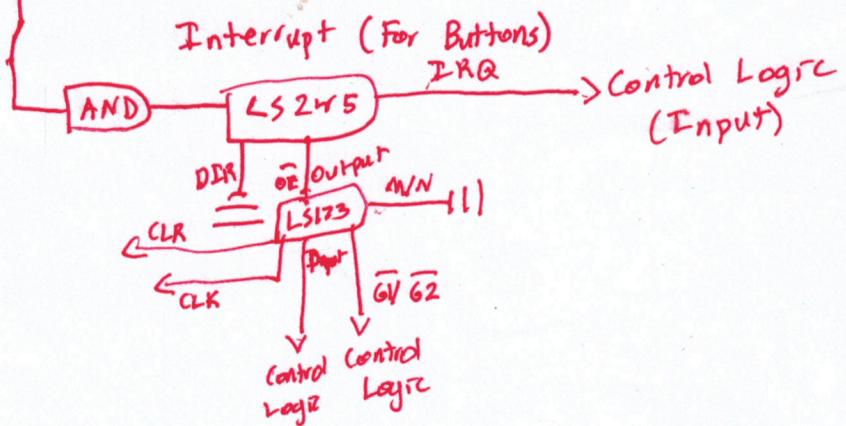
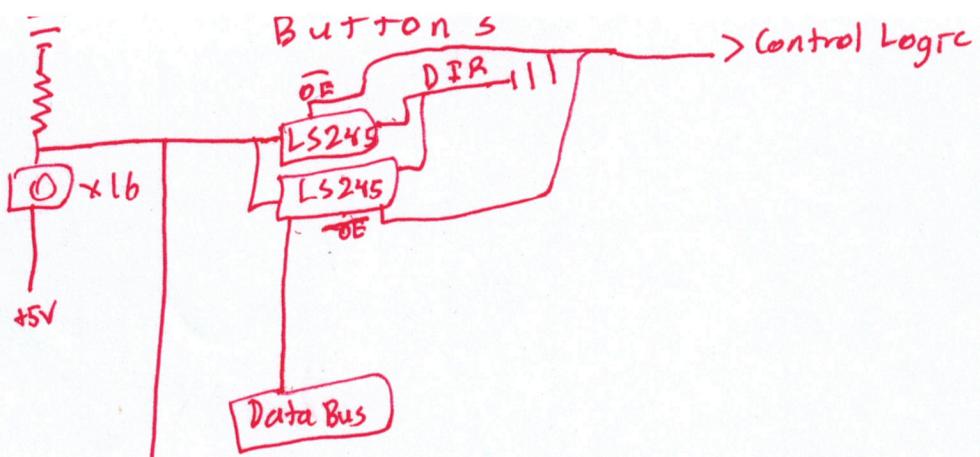
LDA 0

Cats w/o # loads A with a 5.

Cats with # loads A with the address of cats.

looks at data from

Make a new LDA that ~~looks~~ an add., goes to that add., and then ~~g~~ LDAs with the data from the final address.



Reset Circuit

