theWickedWebDev/8-bit-computer

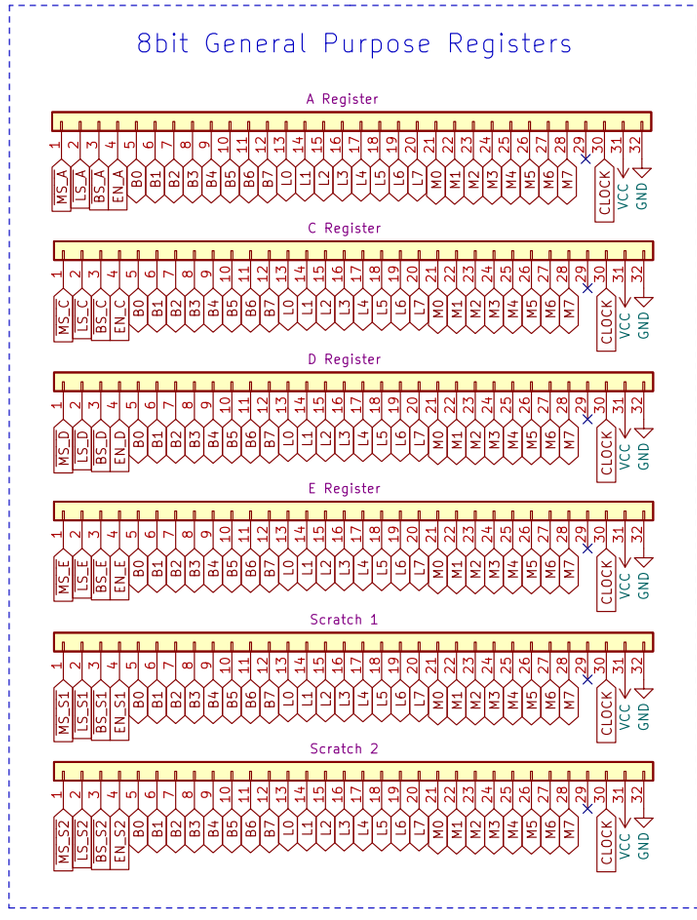
Archimedes8 CPU

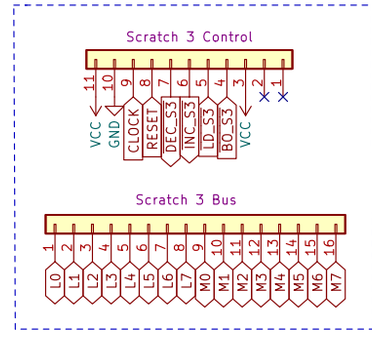
USER MANUAL

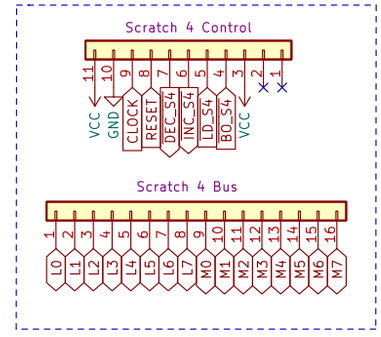
V0.1

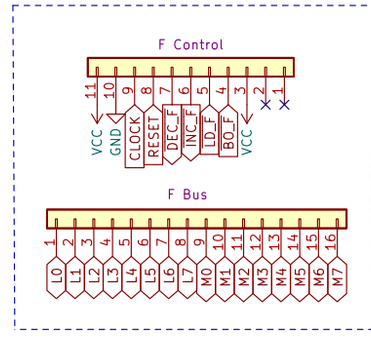
General Purpose Registers & Transfer Register

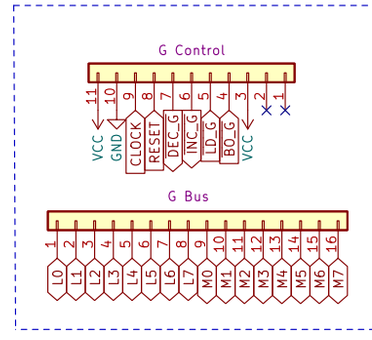
GPR Backplane

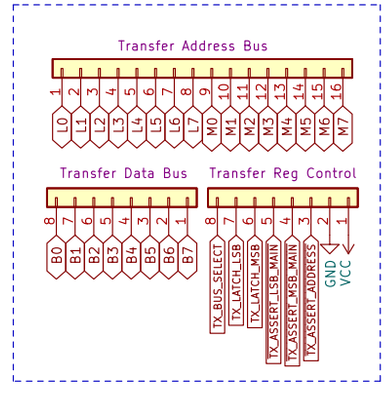


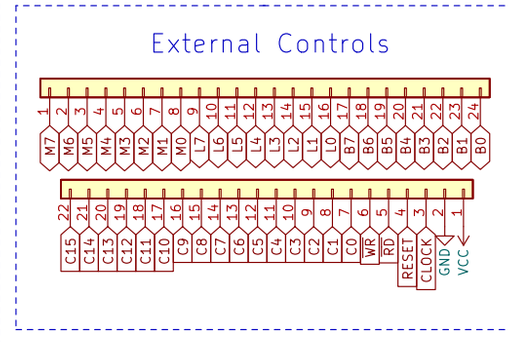


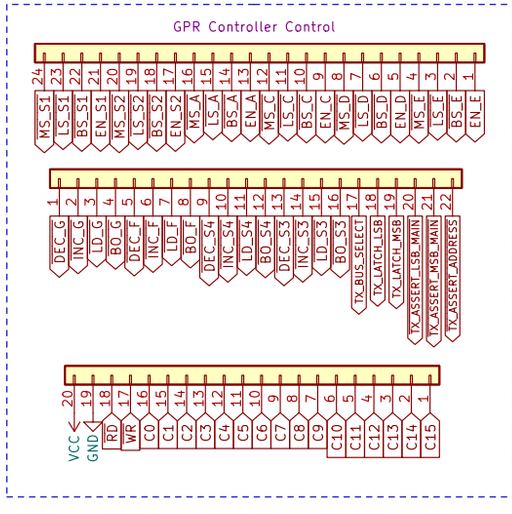


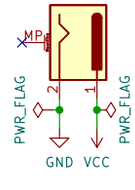












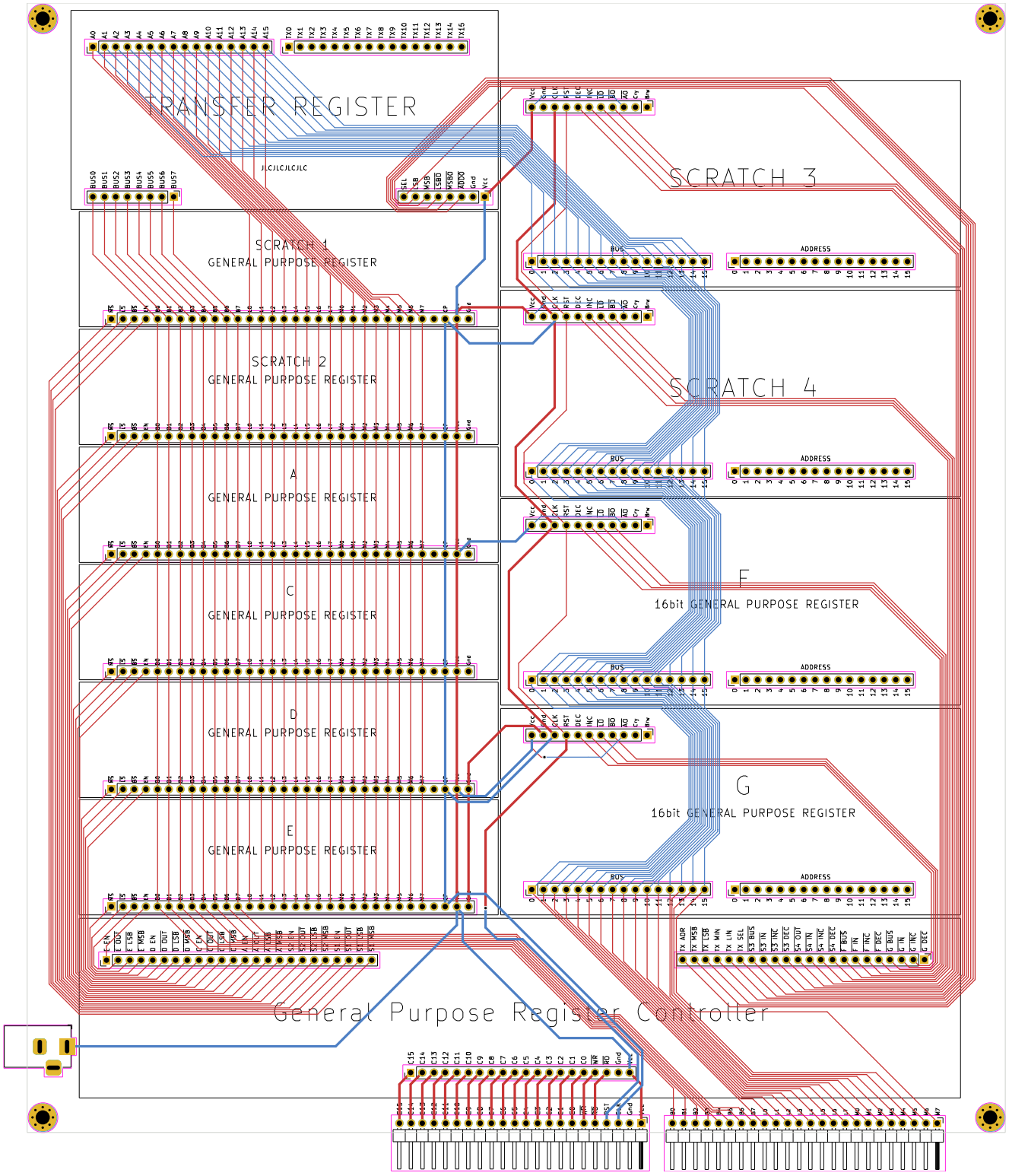
theWickedWebDev/8-bit-computer

Sheet: /

File: new-register-backplane.kicad_sch

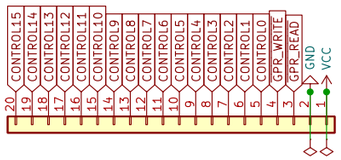
Title: GPR Backplane

Size: User	Date: 2022-01-18	**Rev: 4**
KiCad E.D.A. eeschema (6.0.0-0)		Id: 1/1

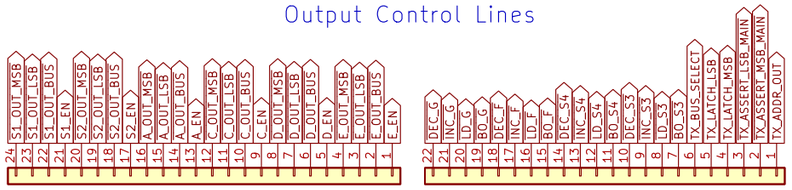

TRANSFER REGISTER
SCRATCH 1
GENERAL PURPOSE REGISTER
SCRATCH 2
GENERAL PURPOSE REGISTER
A
GENERAL PURPOSE REGISTER
C
GENERAL PURPOSE REGISTER
D
GENERAL PURPOSE REGISTER
E
GENERAL PURPOSE REGISTER
SCRATCH 3
SCRATCH 4
F
16bit GENERAL PURPOSE REGISTER
G
16bit GENERAL PURPOSE REGISTER
BUS
ADDRESS
General Purpose Register Controller

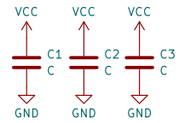
GENERAL PURPOSE REGISTER ARRAY CONTROLLER

Control Input Connections

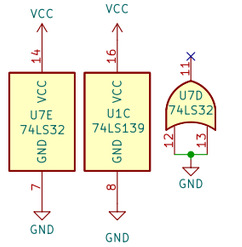


Output Control Lines

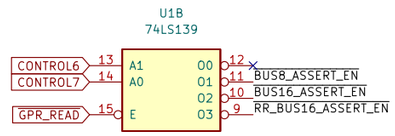




Logic Gate Power

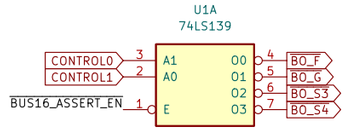


Assertions MUX



aaXXXXXX

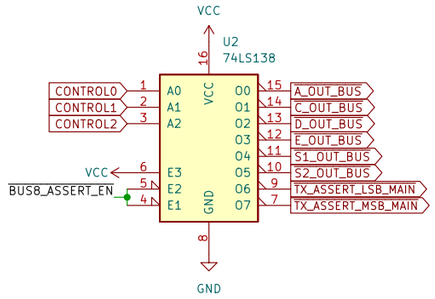
Assert R16 => Bus16



10XXXXaa

00 – F
01 – D
10 – S3
11 – S4

Assert R8 => Bus8

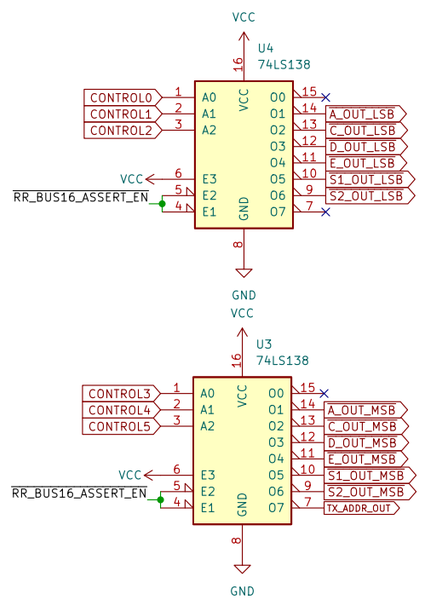


01aaabbb

01xxxaaa

000 – A
001 – C
010 – D
011 – E
100 – S1
101 – S2
110 – TX LSB
111 – TX MSB

Assert R8R8 => Bus16

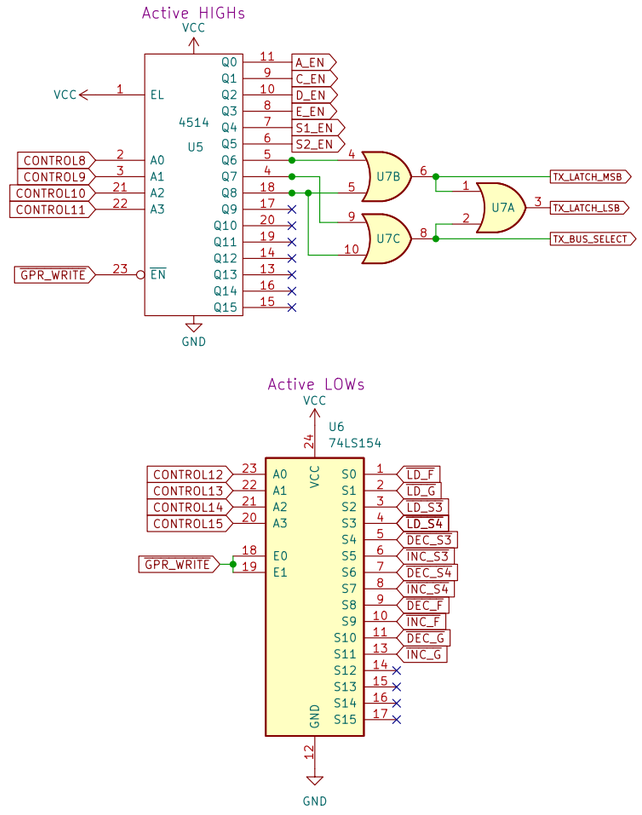


11aaabbb

aaa & bbb are register addresses for LSB and MSB respectively

000 – NOP
001 – A
010 – C
011 – D
100 – E
101 – S1
110 – S2
111 – NOP

Loads



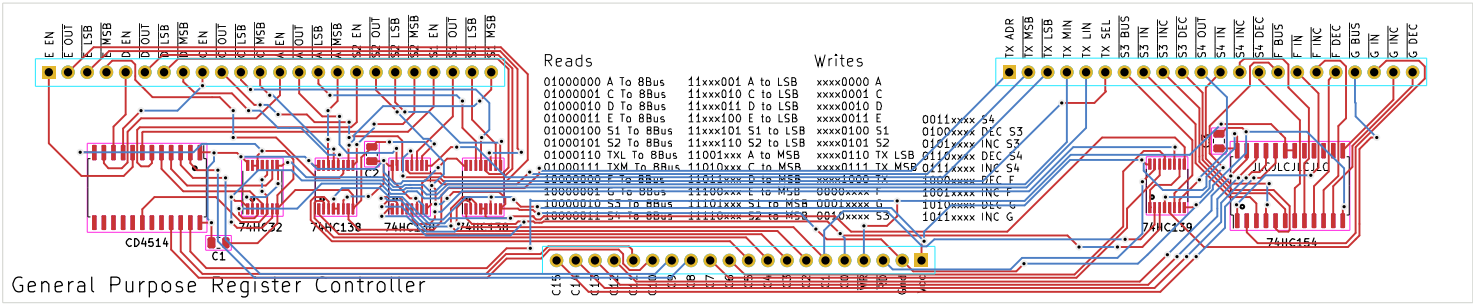
Controls Load, Assert, Inc/Dec's for all GPRs

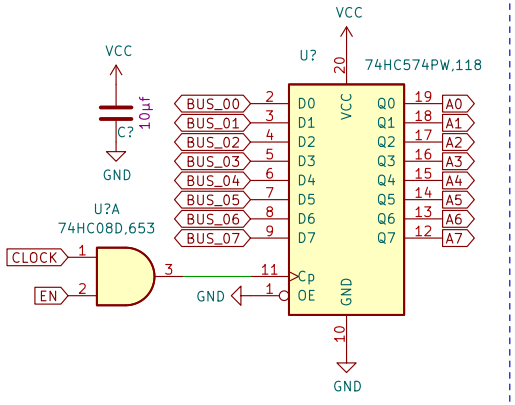
theWickedWebDev/8-bit-computer

Sheet: /
File: gpr-controller-combined.kicad_sch

Title: General Purpose Register Controller

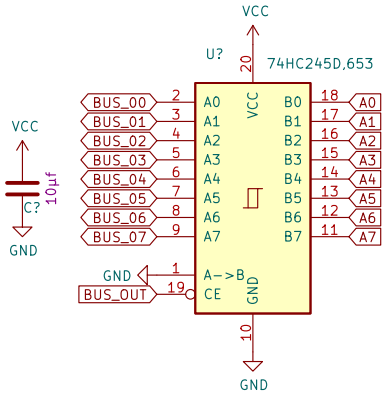
Size: User	Date: 2022-01-11	**Rev: 3**
KiCad E.D.A. kicad (6.0.0-0)		Id: 1/1


E EN
E OUT
E LSB
E MSB
D EN
D OUT
D LSB
D MSB
C EN
C OUT
C LSB
C MSB
A EN
A OUT
A LSB
A MSB
S2 EN
S2 OUT
S2 LSB
S2 MSB
S1 EN
S1 OUT
S1 LSB
S1 MSB
Reads
01000000 A To BBus
01000001 C To BBus
01000010 D To BBus
01000011 E To BBus
01000100 S1 To BBus
01000101 S2 To BBus
01000110 TXL To BBus
01000111 TXM To BBus
10000000 F To BBus
10000001 G To BBus
10000010 S3 To BBus
10000011 S4 To BBus
11xxx001 A to LSB
11xxx010 C to LSB
11xxx011 D to LSB
11xxx100 E to LSB
11xxx101 S1 to LSB
11xxx110 S2 to LSB
11001xxx A to MSB
11010xxx C to MSB
11011xxx D to MSB
11100xxx E to MSB
11101xxx S1 to MSB
11110xxx S2 to MSB
Writes
xxxx0000 A
xxxx0001 C
xxxx0010 D
xxxx0011 E
xxxx0100 S1
xxxx0101 S2
xxxx0110 TX LSB
xxxx0111 TX MSB
1000xxxx TX
1001xxxx F
1010xxxx S3
1011xxxx S4
0011xxxx S4
0100xxxx DEC S3
0101xxxx INC S3
0110xxxx DEC S4
0111xxxx INC S4
1000xxxx DEC F
1001xxxx INC F
1010xxxx DEC G
1011xxxx INC G
TX ADR
TX MSB
TX LSB
TX MIN
TX LIN
TX SEL
S3 BUS
S3 TN
S3 INC
S3 DEC
S4 OUT
S4 IN
S4 INC
S4 DEC
F BUS
F TN
F INC
F DEC
G BUS
G TN
G INC
G DEC
CD4514
C1
74HC32
74HC138
74HC139
74HC154
C15
C14
C13
C12
C11
C10
C9
C8
C7
C6
C5
C4
C3
C2
C1
C0
GND
VCC
General Purpose Register Controller



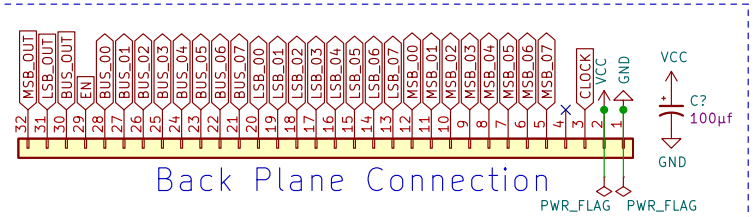
8bit Register

This will latch in data from the Data BUS with the rising edge of the clock and a HIGH signal on EN line

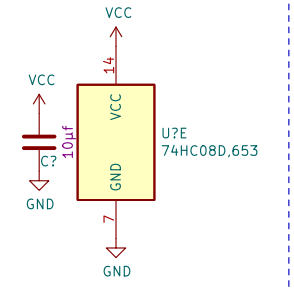


Non Inverting Buffer

When BUS_OUT is set LOW, this register will assert its value out onto the DATA BUS

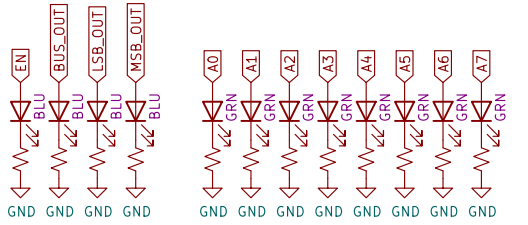


Back Plane Connection

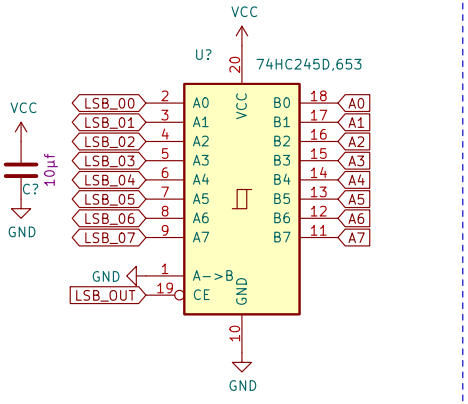


AND Gate

Used to Enable the Register along with the Clock Pulse

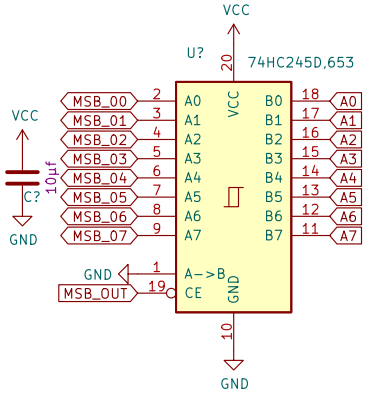


LED Indicators



Non Inverting Buffer

When LSB_OUT is set LOW, this register will assert its value out onto the LSB BUS



Non Inverting Buffer

When MSB_OUT is set LOW, this register will assert its value out onto the MSB BUS

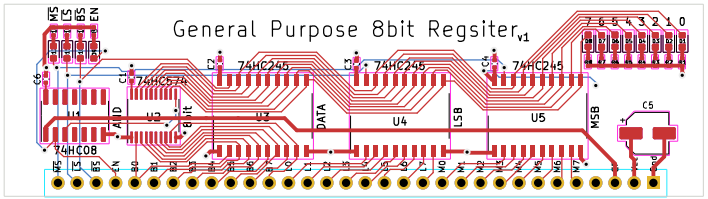
theWickedWebDev/8-bit-computer

Sheet: /

File: smd-register-array.kicad_sch

Title: 8-bit General Purpose Register

Size: A4	Date: 2022-01-18	**Rev: 2**
KiCad E.D.A. kicad (6.0.0-0)		Id: 1/1

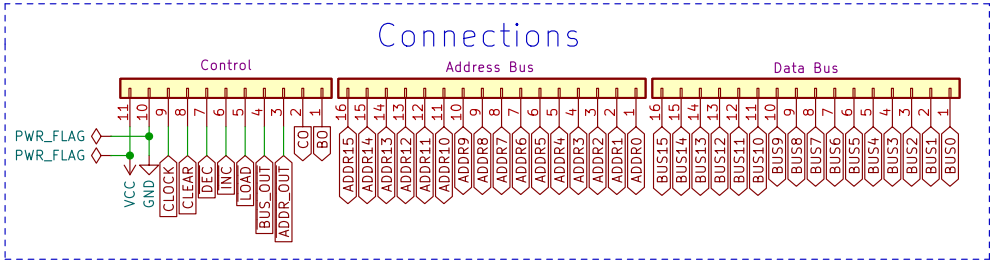

General Purpose 8bit Regsiter v1
MS
CS
BS
EN
74HC574
74HC245
74HC245
74HC245
74HC08
AND
8bit
DATA
LSB
MSB
U1
U2
U3
U4
U5
C1
C2
C3
C4
C5
C6

16bit Address Register

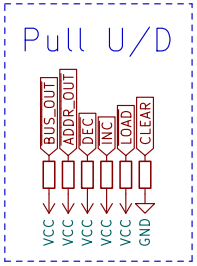
These registers provide functionality to different components
such as the Stack Pointer, Program Counter, or any 16bit
GPR. They have the ability to INC or DEC without using the ALU.

CLK	$\overline{DEC}$	$\overline{INC}$	$\overline{LOAD}$	$\overline{BUS}$	$\overline{ADDR}$		
x	H	H	H	H	H	–	NOOP
/	H	H	L	x	x	–	Load
/	L	H	x	x	x	–	Decrement
/	H	L	x	x	x	–	Increment

Connections

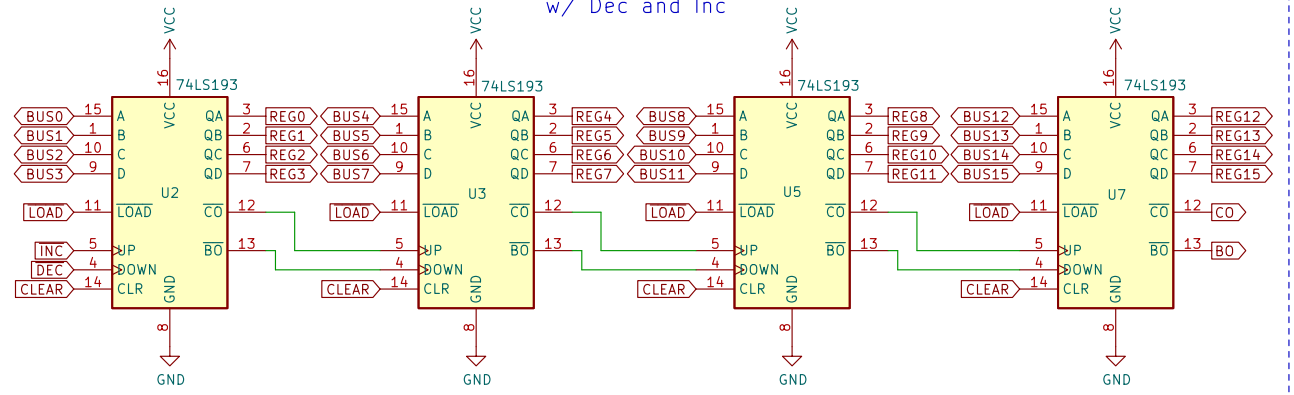


Pull U/D

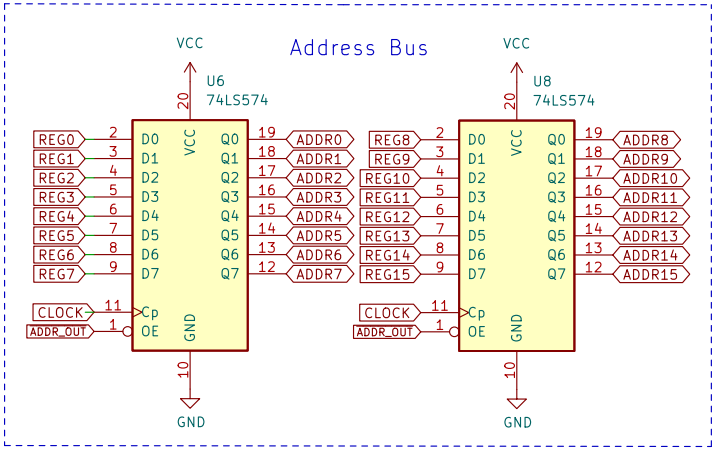


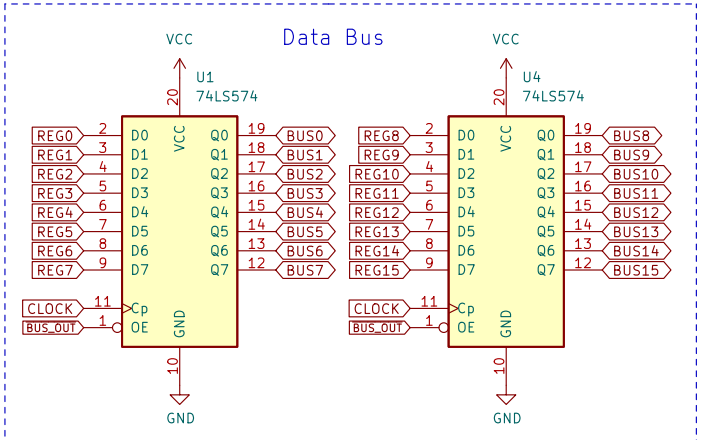
16bit Register

w/ Dec and Inc

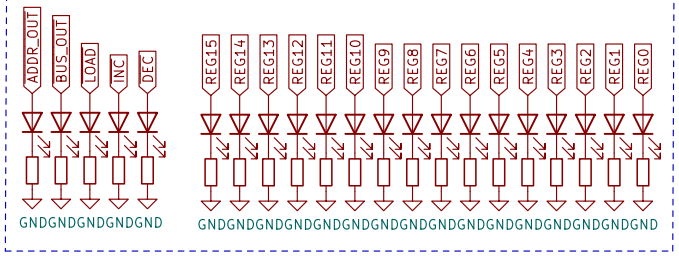


Output Buffers

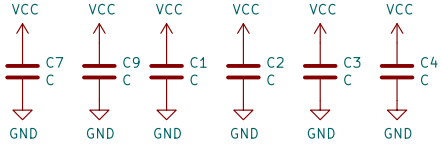




LED Indicator's



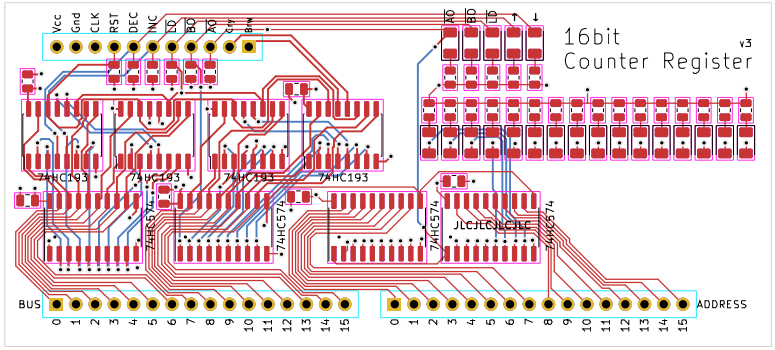
Smoothing Caps



Sheet: /
File: address-register-smd.kicad_sch

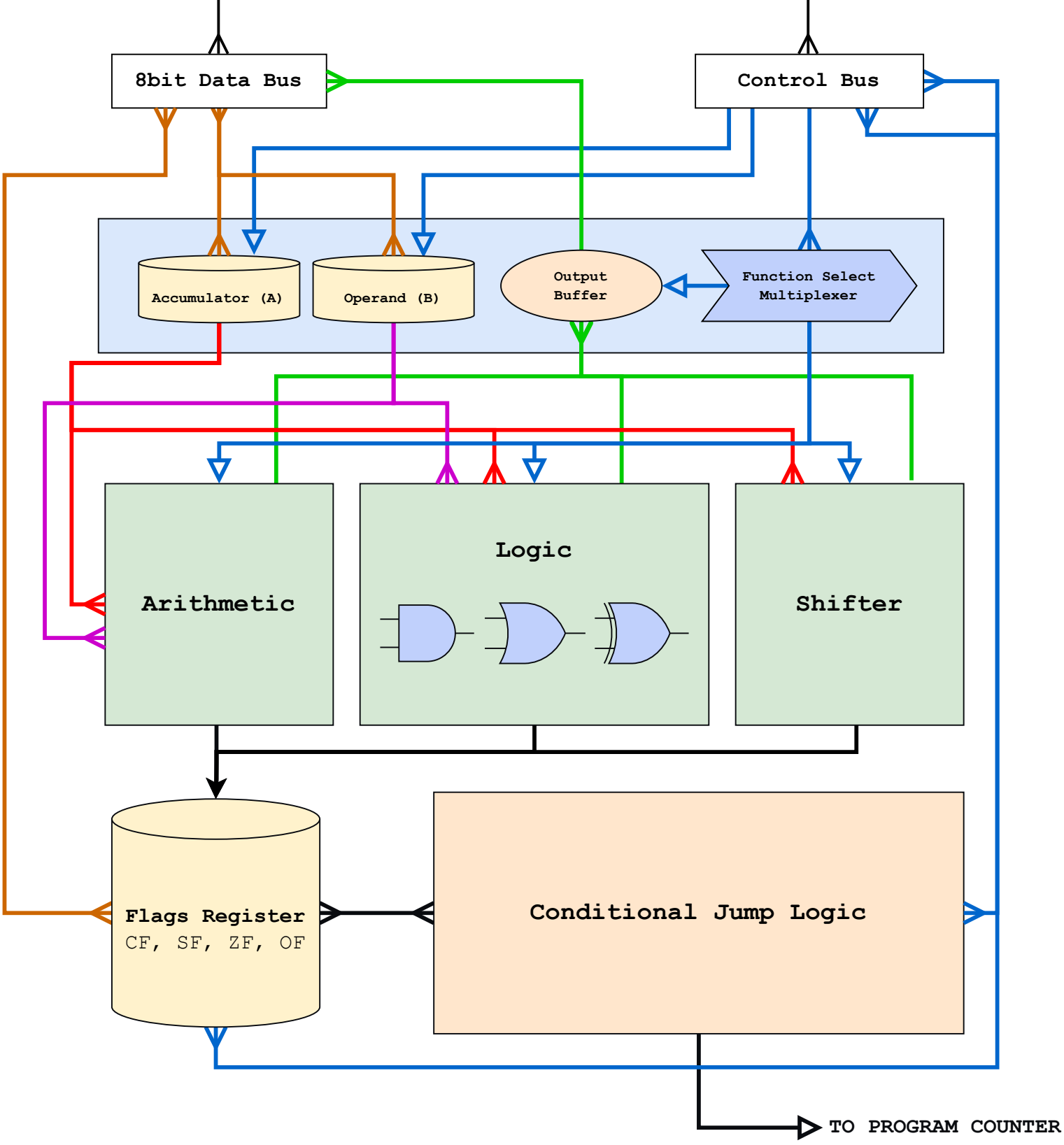
Title: 16 Bit Address Register w/ INC and DEC

Size: A4	Date:	**Rev: 3**
KiCad E.D.A. kicad (6.0.0-0)		Id: 1/1


16bit
Counter Register
v3
Vcc
Gnd
CLK
RST
DEC
INC
~LD
~BD
~AD
~CW
~BW
~AD
~BD
~LD
74HC193
74HC193
74HC193
74HC193
74HC574
74HC574
74HC574
74HC574
JLCJLCJLCJLC
BUS
ADDRESS
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Arithmetic Logic Unit (ALU)

& Flags Register


8bit Data Bus
Control Bus
Accumulator (A)
Operand (B)
Output Buffer
Function Select Multiplexer
Arithmetic
Logic
Shifter
Flags Register
CF, SF, ZF, OF
Conditional Jump Logic
TO PROGRAM COUNTER

Arithmetic Logic Unit and Conditional Jump

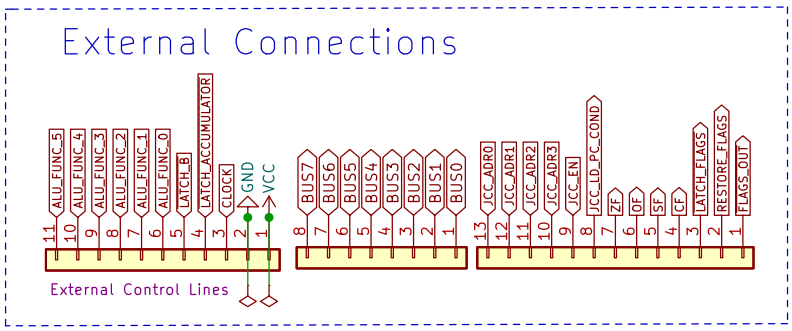
TRUTH TABLE

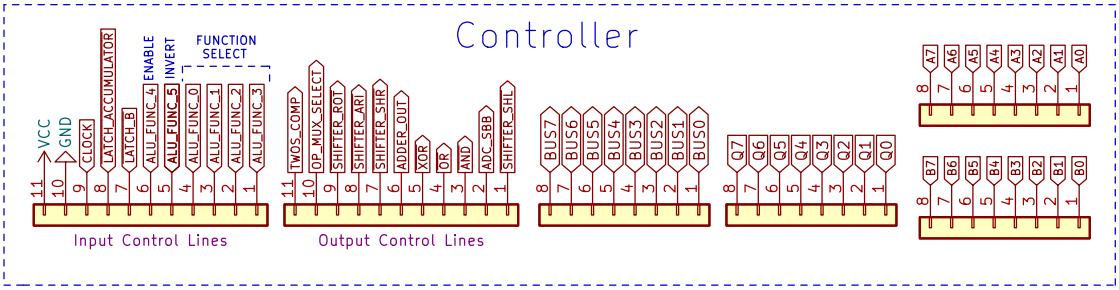
		A*	ADD	ADC	SUB	SBB	INC	DEC	AND	OR	XOR	SHL	SHR	ASL	ASR	ROR	ROL	CMP	TEST	NAND	NOR	XNOR	NOT A
AI	L	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/
BI	L	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/	_/
FLAG_IN	L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
ALU F0		0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	0	1	0
ALU F1		0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	1	1	0	0	0
ALU F2		0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	1	1	0	0	0
ALU F3		0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	1	1	0
(Invert Output) F4		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
(Enable ALU) F5		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

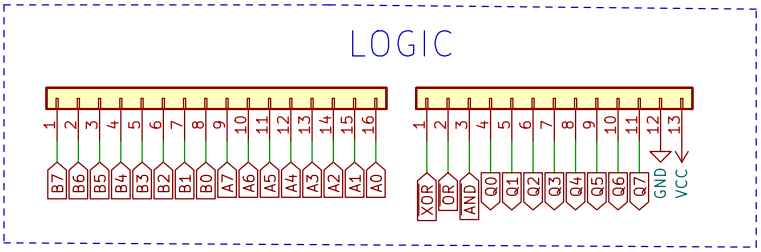
		NOP	JP	JLE JNG	JG JNLE	JGE JNL	JL JNGE	JA JNBE	JBE JNA	JNB JAE JNC	JB JNAE JC	JNE JNZ	JE JZ	JNS	JS	JNO	JO
JCC_EN	L	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
JCC_ADD3		x	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
JCC_ADD2		x	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
JCC_ADD1		x	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
JCC_ADD0		x	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

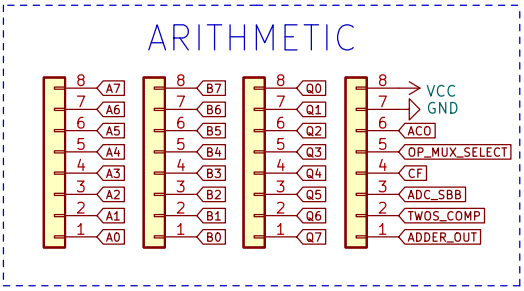
		PUSH FLG*	PUSH FLG*
FLAG_OUT	L	L	L
RESTORE_FLAGS	H	L	H

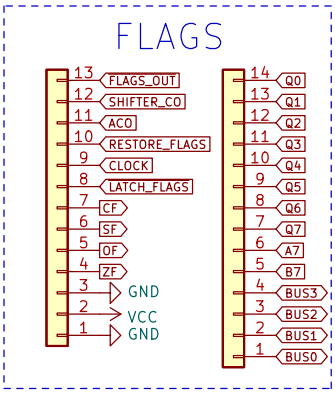
ALU Backplane

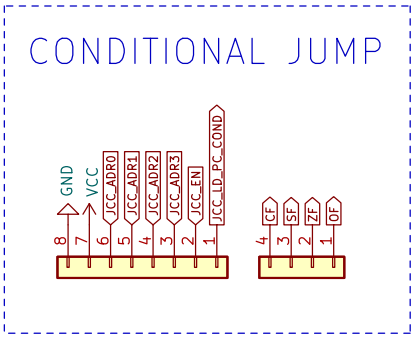


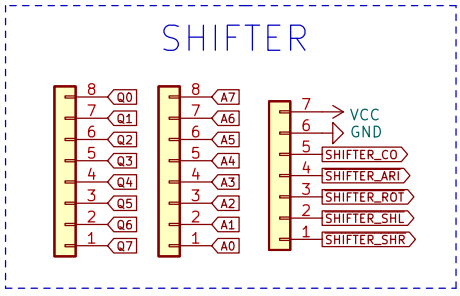










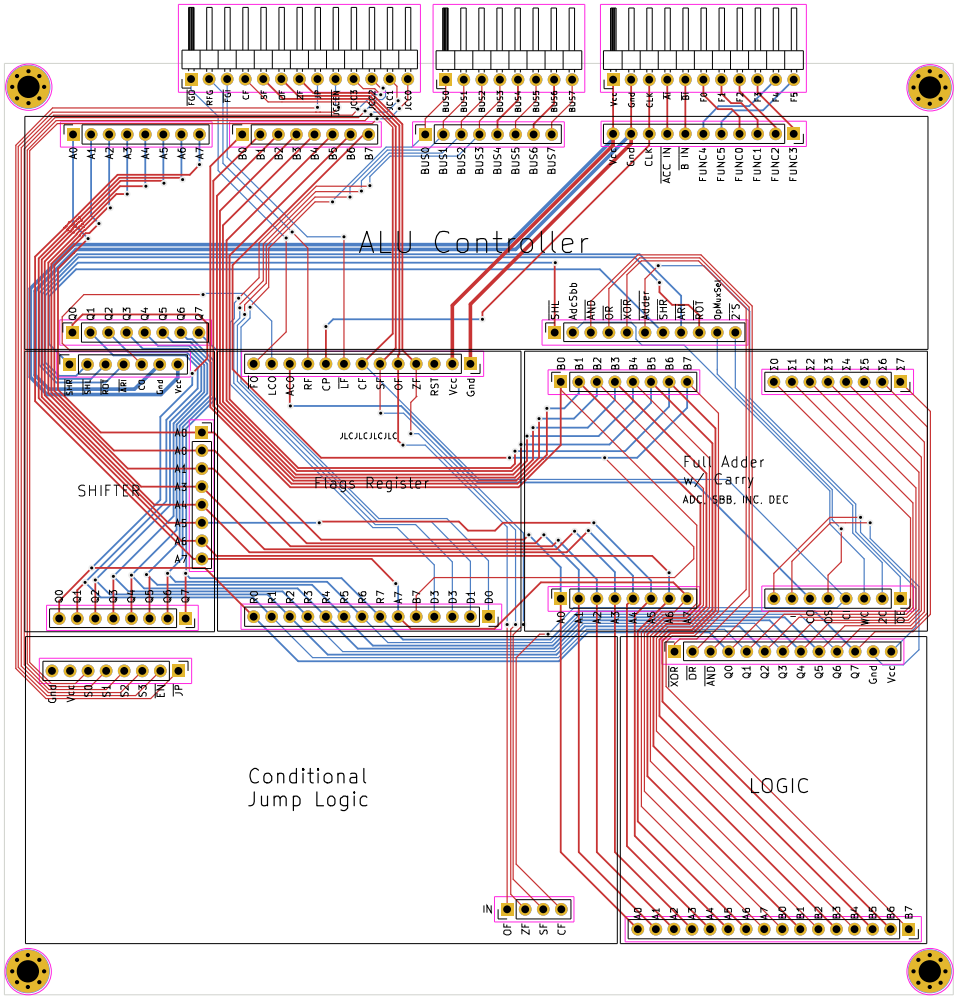


theWickedWebDev/8-bit-computer

Sheet: /
File: ALU-Backplane.kicad_sch

Title: ALU & Conditional Jump Backplane

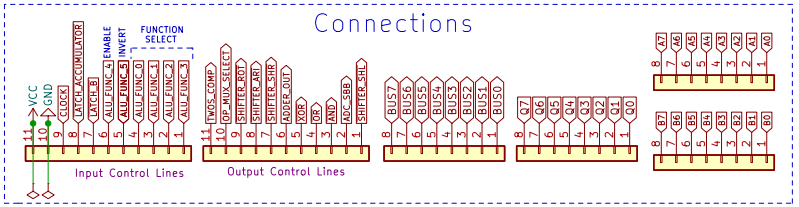
Size: User	Date: 2022-01-17	**Rev: 4**
KiCad E.D.A. kicad (6.0.0-0)		Id: 1/1


ALU Controller
SHIFTER
Flags Register
Full Adder
w/ Carry
ADC, SBB, INC, DEC
Conditional
Jump Logic
LOGIC

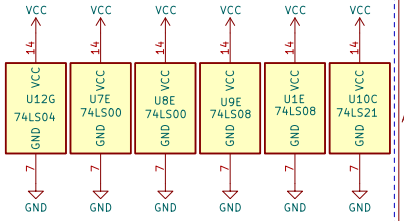
ALU Control Module

A + 1	AND	SHL
A − 1	NAND	SHR
A + B	OR	ASL
A − B	NOR	ASR
A	XOR	ROR
NOT A	XNOR	ROL

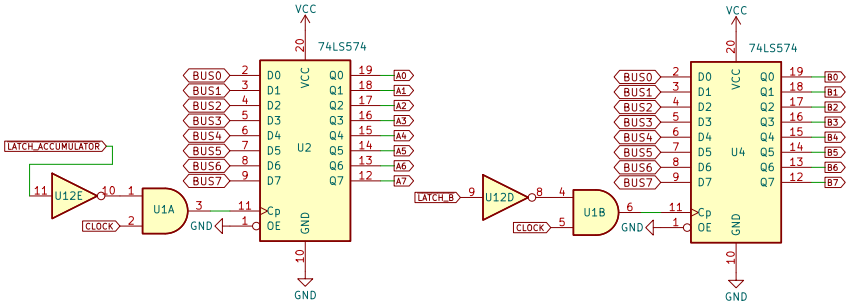
Connections



Logic Power

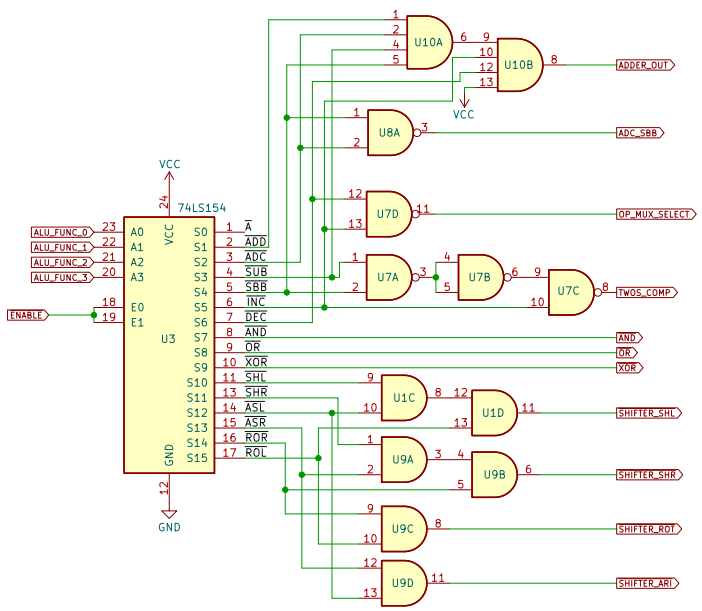


ACCUMULATOR (A)

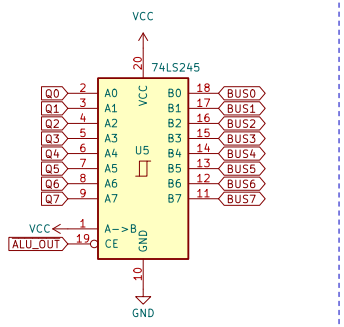


OPERAND (B)

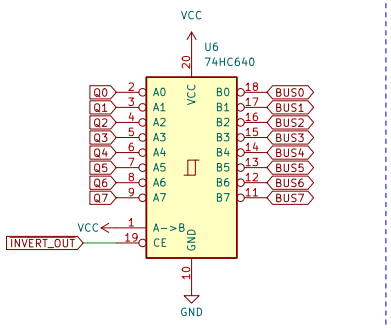
FUNCTION MULTIPLEXER



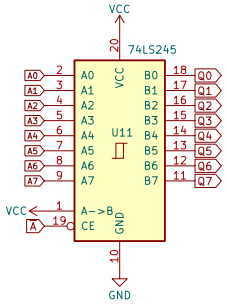
ALU OUTPUT



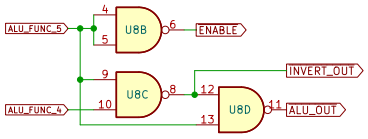
INVERTED ALU OUTPUT



PASS A

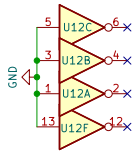


Enable and Invert Control

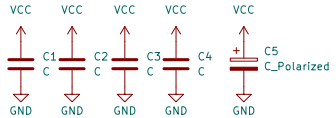


F5|F4
0 0 – NOP (Disable)
0 1 – NOP (Disable)
1 0 – ALU Output and Enable
1 1 – ALU Inverted Output and Enable

Unused



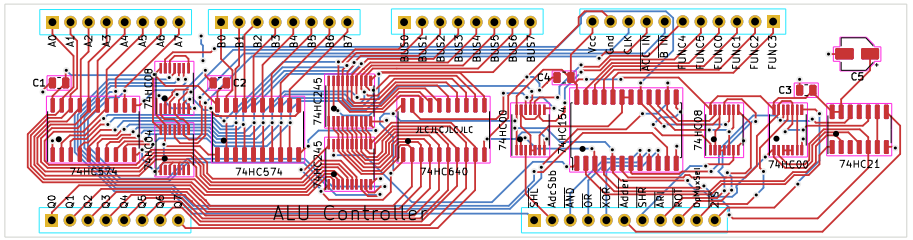
Smoothing Caps



Sheet: /
File: Control_and_Output.kicad_sch

Title:

Size: User	Date:	**Rev:**
KiCad E.D.A. kicad (6.0.0-0)		Id: 1/1


ALU Controller
74HC574
74HC574
74HC640
74HC00
74HC21
C1
C2
C3
C4
C5

Arithmetic Module

8bit Arithmetic Module provides
ADD, ADC, SUB, SBB, INC, DEC

DEC A: MUX_SEL: 1, TC: 0, CI: 0

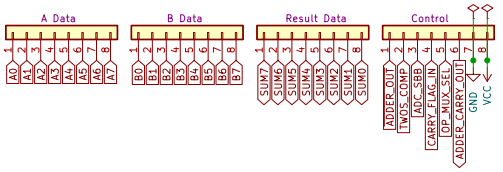
A − B: MUX_SEL: 0, TC: 1, CI: 1

A + B: MUX_SEL: 0, TC: 0, CI: 0

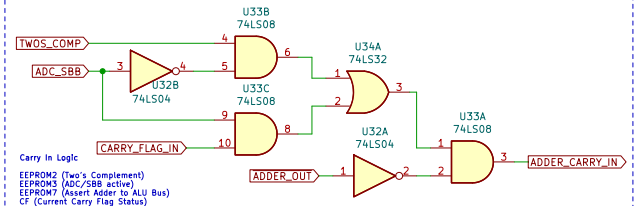
INC A: MUX_SEL: 1, TC: 1, CI: 1

A − B − Ci: MUX_SEL: 0, TC: 1, CI: ?
A + B + Cf: MUX_SEL: 0, TC: 0, CI: ?

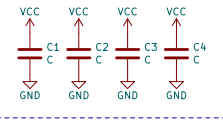
Connectors



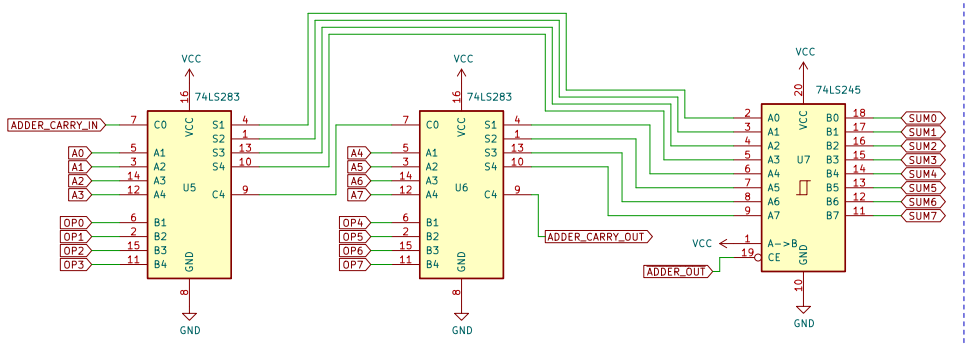
Arithmetic Carry In Logic



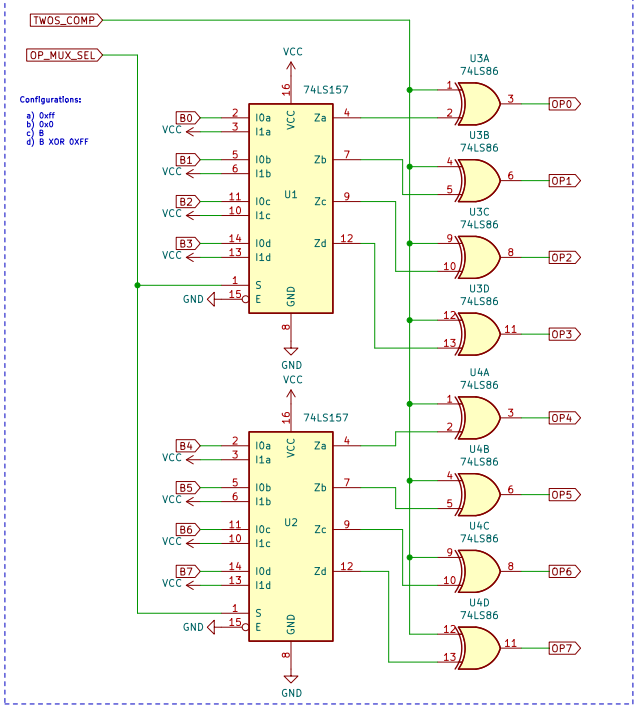
Smoothing Caps



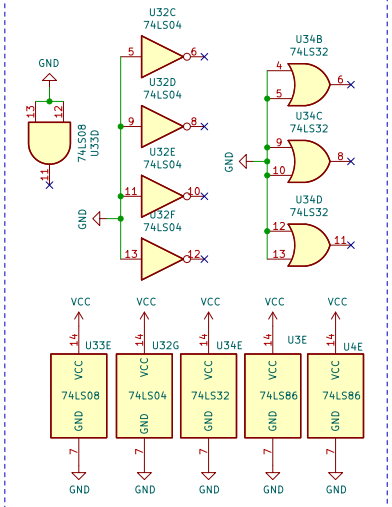
Full Adder w/ Carry



Operand Multiplexer



Logic Power

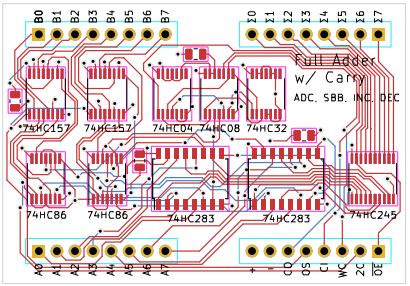


ADD / SUB / ADC / SBB / INC / DEC

Sheet: /
File: Arithmetic.kicad_sch

Title: Arithmetic Module

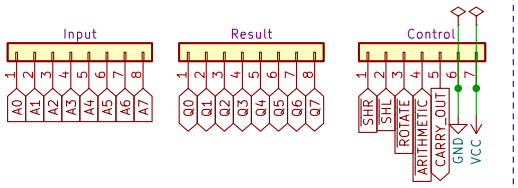
Size: User	Date:	Rev: 3
KiCad E.D.A. kicad (6.0.0−0)		Id: 1/1


B0
B1
B2
B3
B4
B5
B6
B7
I0
I1
I2
I3
I4
I5
I6
I7
Full Adder
w/ Carry
ADC. SBB. INC. DEC
74HC157
74HC157
74HC04
74HC08
74HC32
74HC86
74HC86
74HC283
74HC283
74HC245
A0
A1
A2
A3
A4
A5
A6
A7
+
-
CO
dS
CI
WC
2C
OE

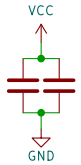
Shift & Rotate Module

Logical / Arithmetic

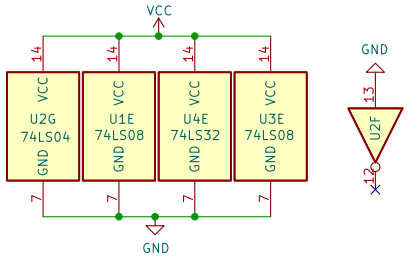
Conections



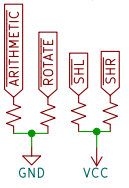
Cap's



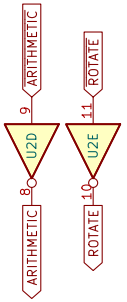
Logic Power



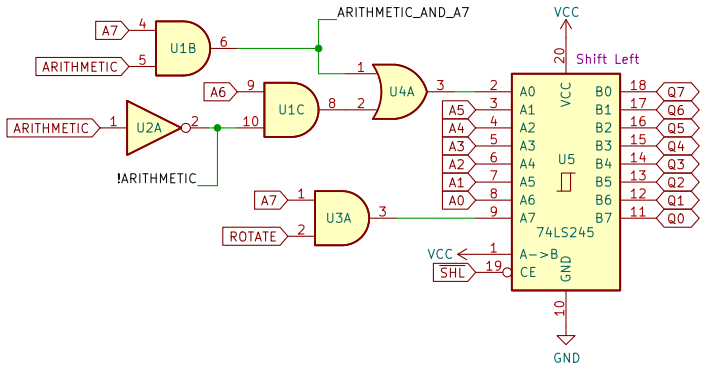
Pull U/D



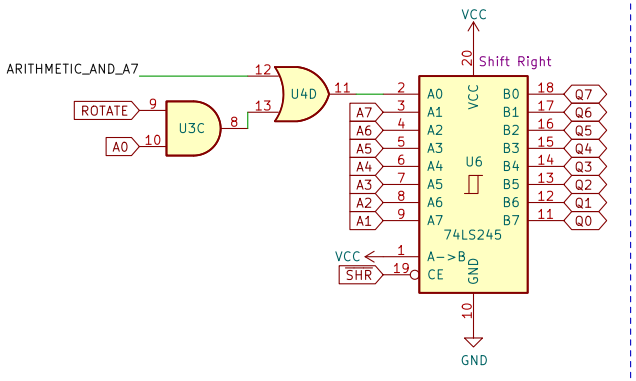
MAKE ACTIVE LOWS



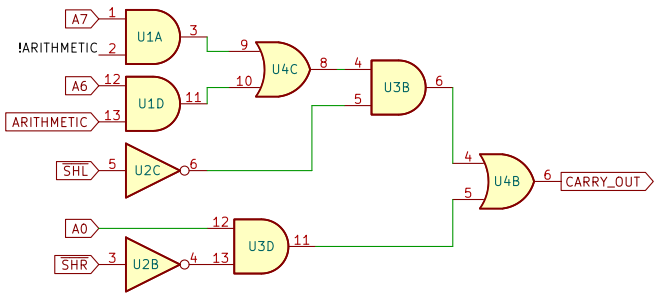
SHIFT LEFT



SHIFT RIGHT



CARRY OUT LOGIC



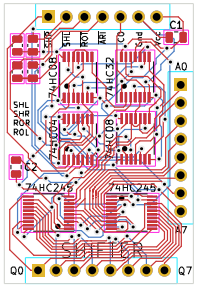
Rotate Enable / Carry Out

theWickedWebDev/8-bit-computer

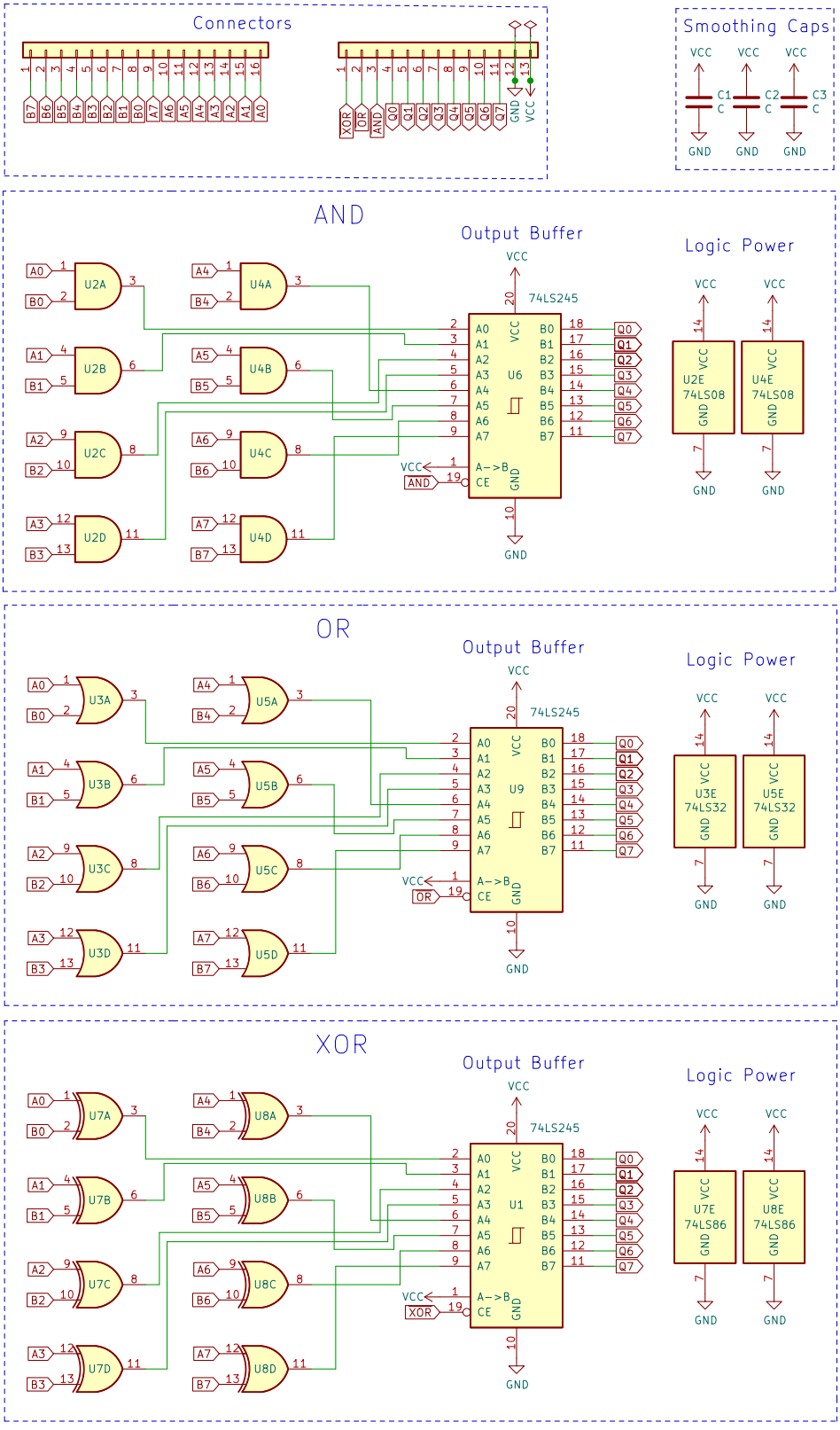
Sheet: /
File: Shifter-v6.kicad_sch

Title: Arithmetic & Logical Shifter Module

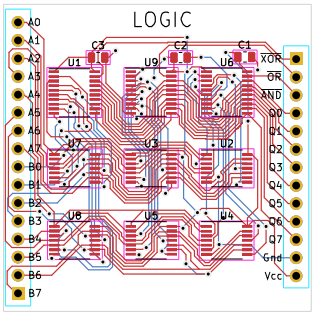
Size: A4	Date: 2022-01-16	**Rev: 4**
KiCad E.D.A. kicad (6.0.0-0)		Id: 1/1


C1
A0
A7
Q0
Q7
C2
74HC245
74HC245
74HC299
74HC299
74HC299
74HC299
SHL
SHR
RO0
RO7

8bit Logic Gate



Sheet: /	
File: Modules.kicad_sch	
Title: 8bit Logic Gate (ALU)	
Size: User	Date:
KiCad E.D.A. kicad (6.0.0-0)	**Rev: 3**
	Id: 1/1


LOGIC
A0
A1
A2
A3
A4
A5
A6
A7
B0
B1
B2
B3
B4
B5
B6
B7
C3
C2
C1
U1
U9
U6
U7
U3
U2
U8
U5
U4
XOR
OR
AND
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Gnd
Vcc

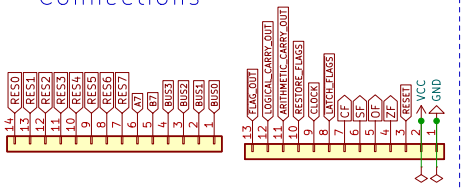
FLAGS REGISTER

$\overline{\text{LATCH\_FLAGS}}$ – A LOW signal will store the data asserted from the multiplexer into the Flags Register (FR)

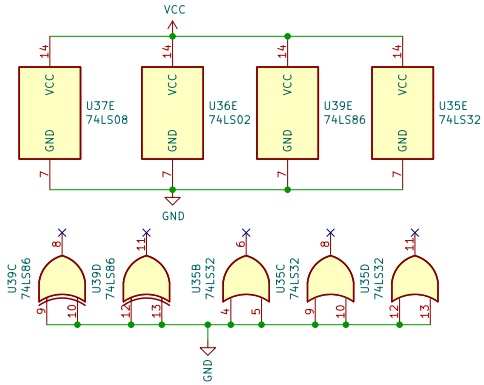
- $\overline{\text{RESTORE}}$: LOW, uses signals from ALU
- RESTORE: HIGH, uses signal asserted on data bus

$\overline{\text{FLAG\_OUT}}$ – Assers the current flags statuses onto the Data bus, typically used to push it onto the stack to handle an ISR

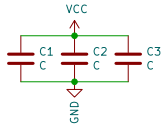
Connections



Logic Gate Power

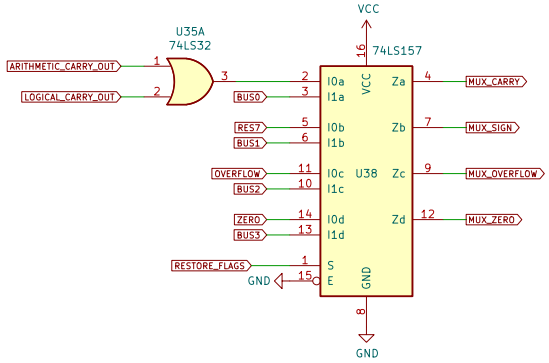


Smoothing Caps

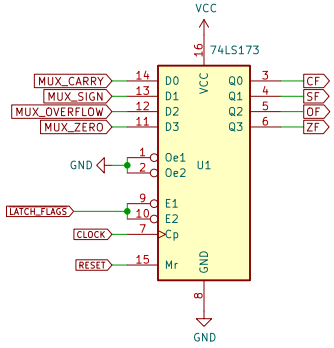


Source Multiplexer

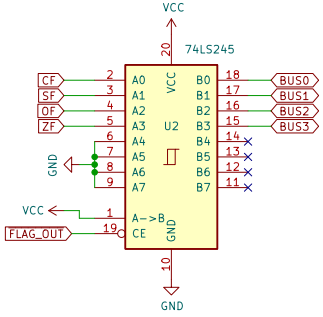
Flags come directly from ALU, or, from the flag/data bus to restore flags from the stack or another location



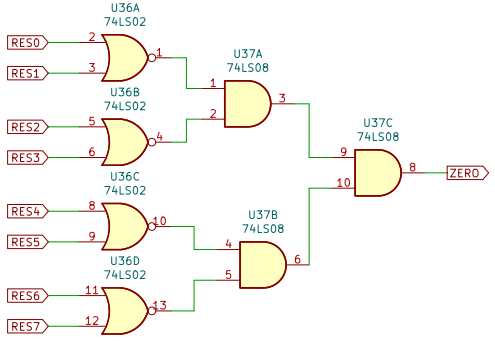
REGISTER



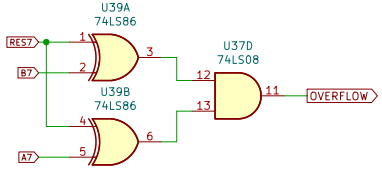
Bus Connection



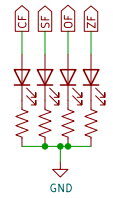
Zero



OVERFLOW



Leds



For storing and asserting current flag statuses from ALU

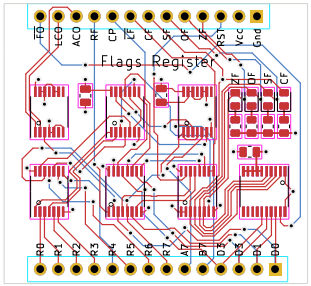
theWickedWebDev/8-Bit-Computer

Sheet: /

File: Flags Register.kicad_sch

Title: Flags Register

Size: User	Date: 2022-01-03	**Rev: 3**
KiCad E.D.A. kicad (6.0.0-0)		Id: 1/1


Flags Register
IEO
LEO
ACO
AF
CP
ZF
CF
SF
OF
IF
RST
Vcc
Gnd
ZF
OF
SF
CF
R0
R1
R2
R3
R4
R5
R6
R7
A7
B7
D7
D6
D5
D4
D0

Memory

ROM & RAM

Input & Output (I/O)

CPU Control