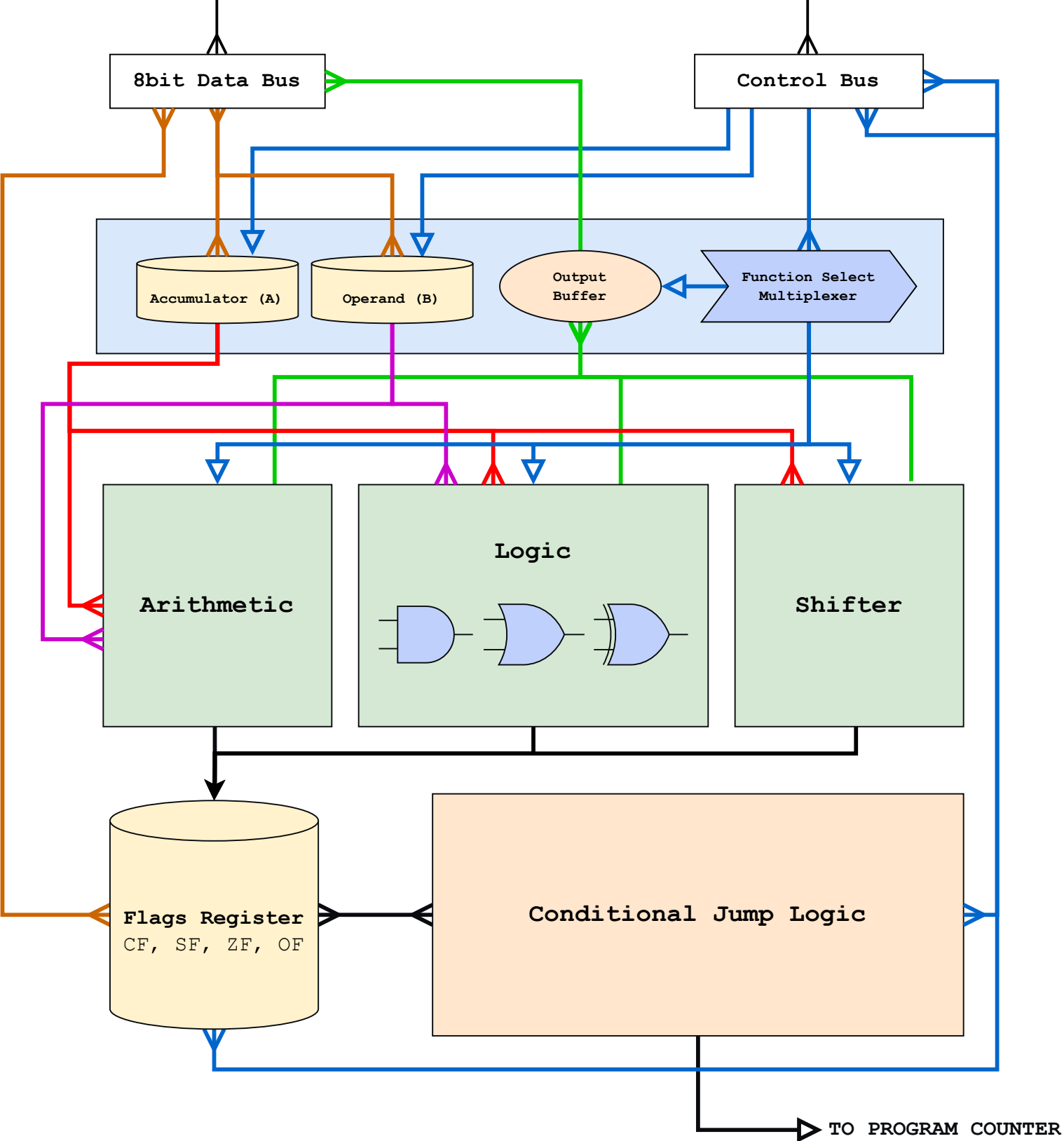


theWickedWebDev/8-bit-computer

Archimedes8 CPU

USER MANUAL

V0.1



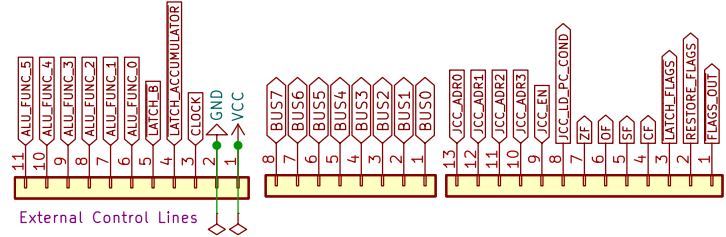
Arithmetic Logic Unit and Conditional Jump

TRUTH TABLE

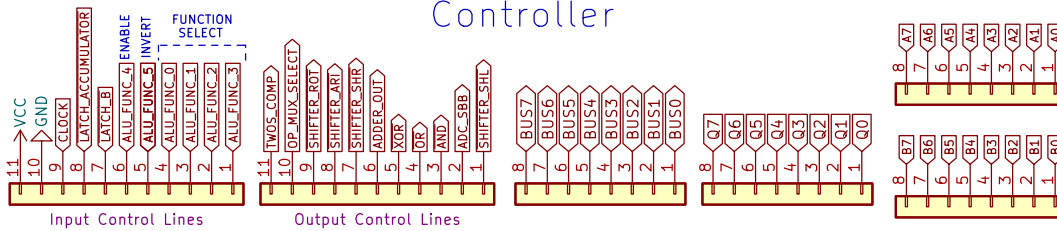
		A*	ADD	ADC	SUB	SBB	INC	DEC	AND	OR	XOR	SHL	SHR	ASL	ASR	ROR	ROL	CMP	TEST	NAND	NOR	XNOR	NOT A
AI	L	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_
BI	L	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_	_/_
FLAG_IN	L	0	0	0	0	0	0	0	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	JG	JGE	JL	JA	JBE	JNB	JB	JNE	JE	JNS	JS	JNO	JO						
				JNG	JNLE	JNL	JNGE	JNBE	JNA	JAE	JNAE	JNZ	JZ										
										JNC	JC												
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

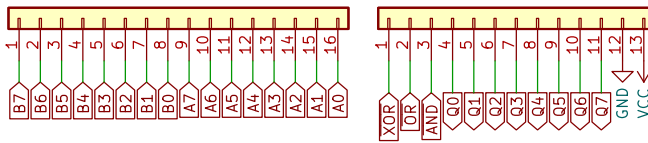
External Connections



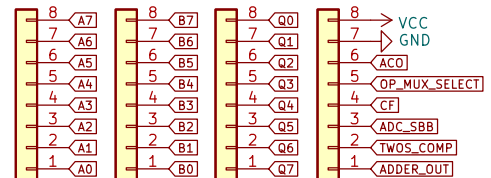
Controller



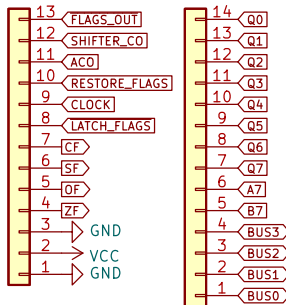
LOGIC



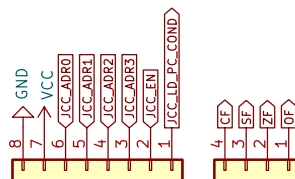
ARITHMETIC



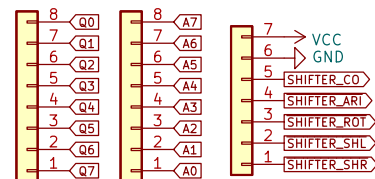
FLAGS



CONDITIONAL JUMP



SHIFTER



theWickedWebDev/8-bit-computer

Sheet: /

File: ALU-Backplane.kicad_sch

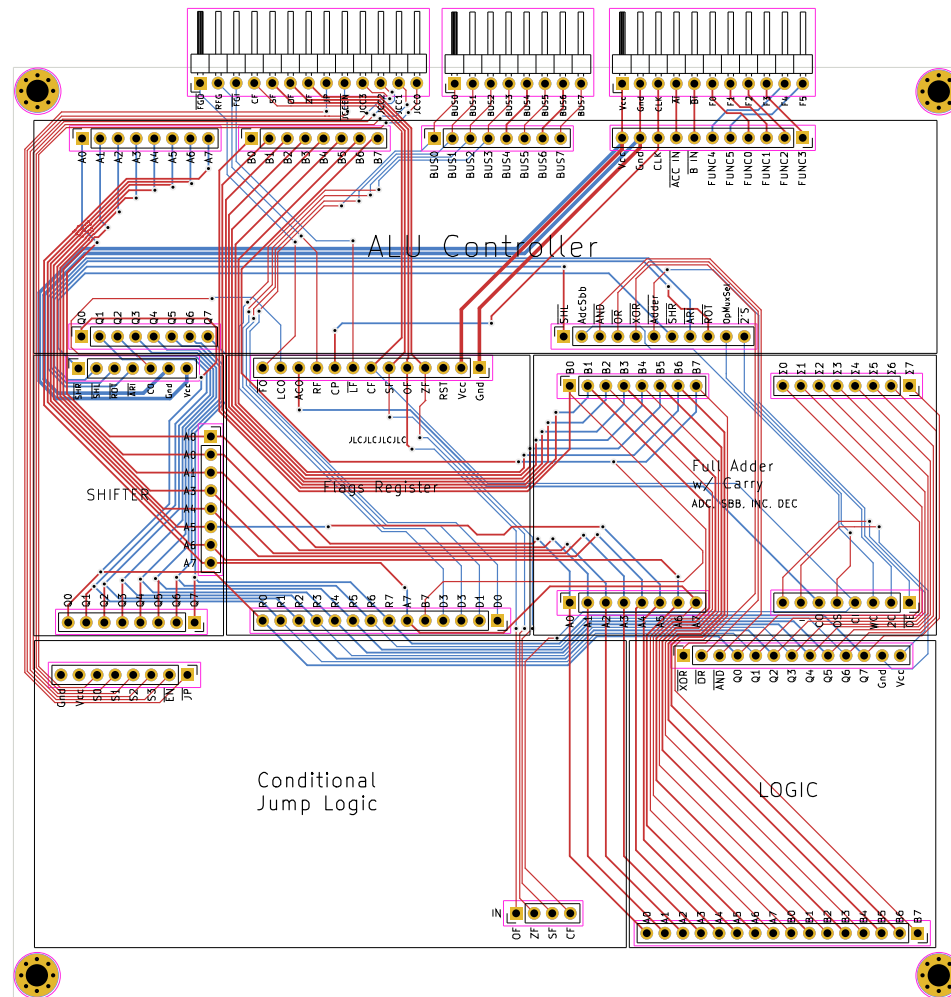
Title: ALU & Conditional Jump Backplane

Size: User Date: 2022-01-17

KiCad E.D.A. kicad (6.0.0-0)

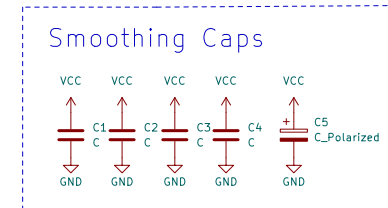
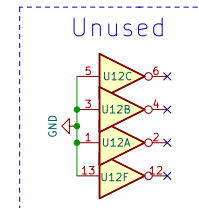
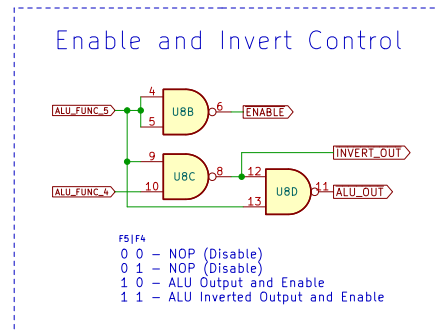
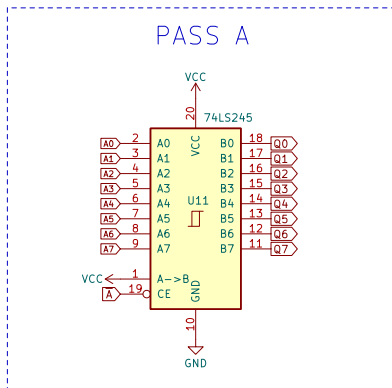
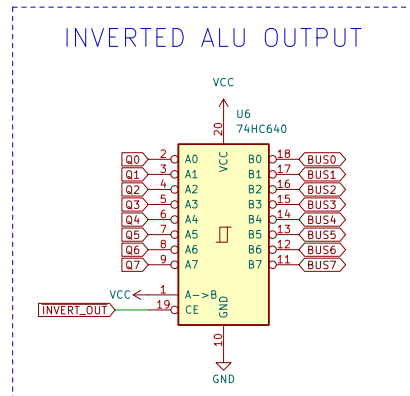
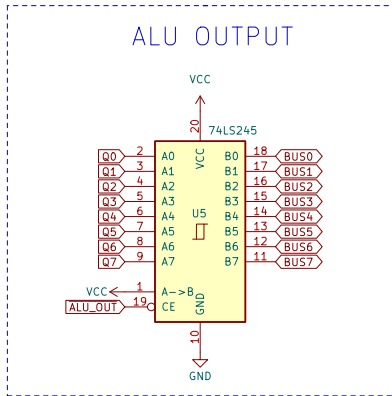
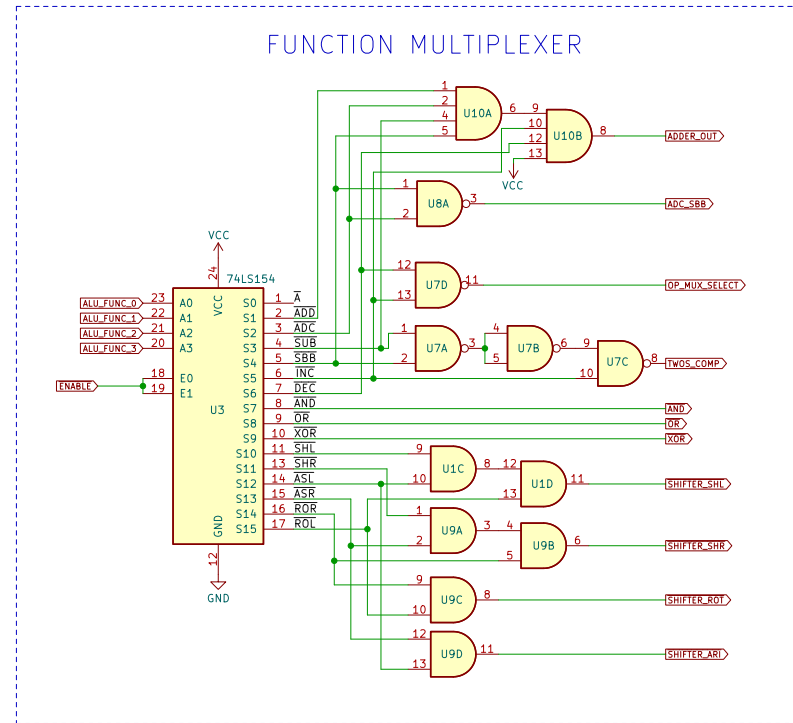
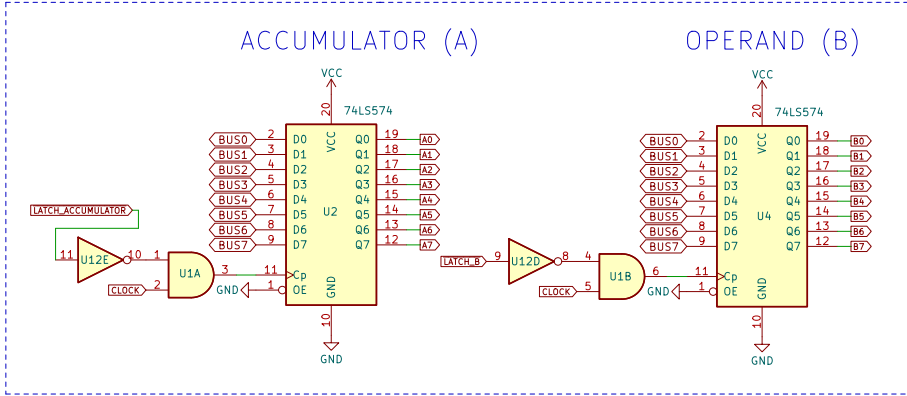
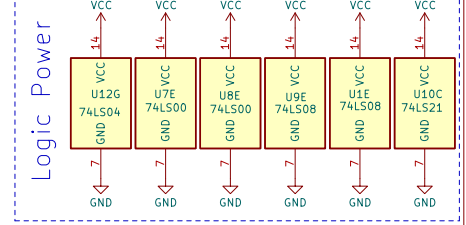
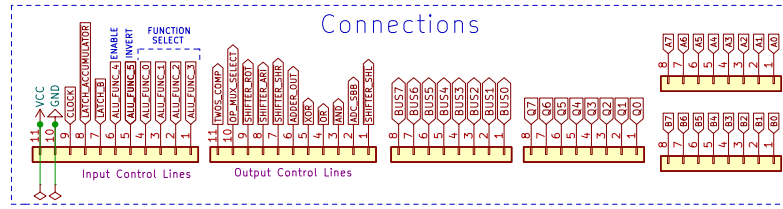
Rev: 4

Id: 1/1



ALU Control Module

$A + 1$ AND SHL
 $A - 1$ NAND SHR
 $A + B$ OR ASL
 $A - B$ NOR ASR
 A XOR ROR
 $\text{NOT } A$ XNOR ROL

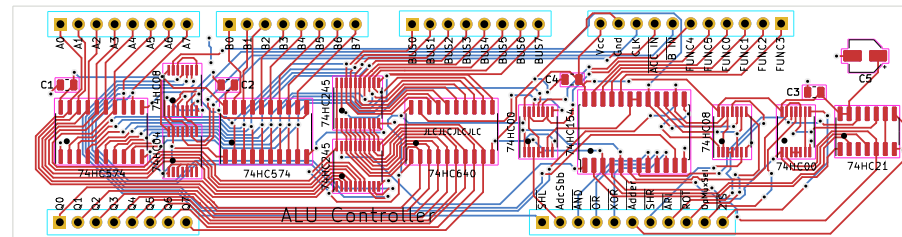


Sheet: /
File: Control_Land_Output.kicad_sch

Title:

Size: User Date:
KiCad E.D.A. kicad (6.0.0-0)

Rev:
Id: 1/1



Arithmetic Module

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

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

A - B: MUX_SEL: 0, TC: 1, Cl: 1

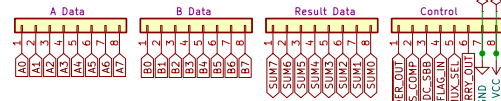
A + B: MUX_SEL: 0, TC: 0, Cl: 0

INC A: MUX_SEL: 1, TC: 1, Cl: 1

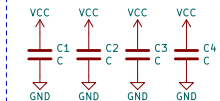
A - B - Cl: MUX_SEL: 0, TC: 1, Cl: ?

A + B + Cl: MUX_SEL: 0, TC: 0, Cl: ?

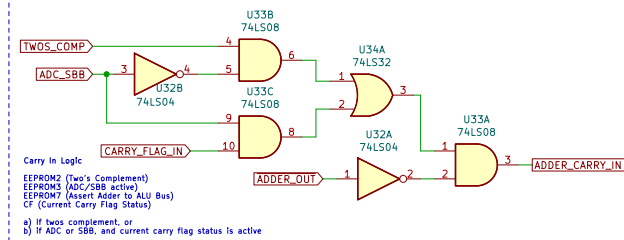
Connectors



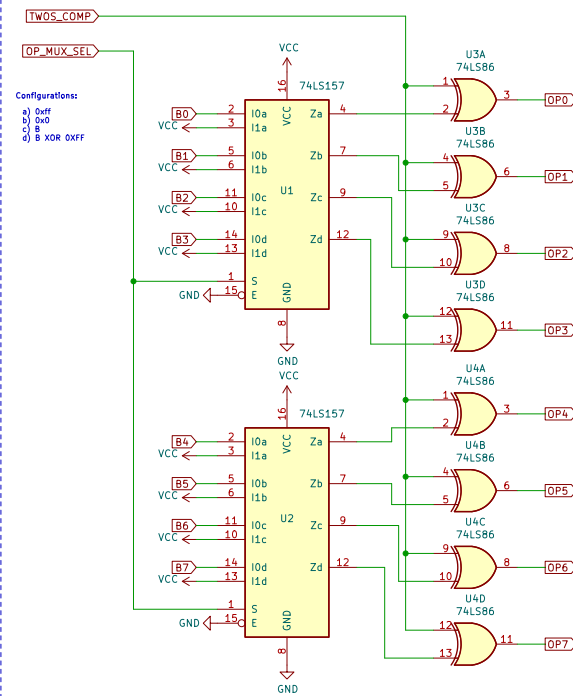
Smoothing Caps



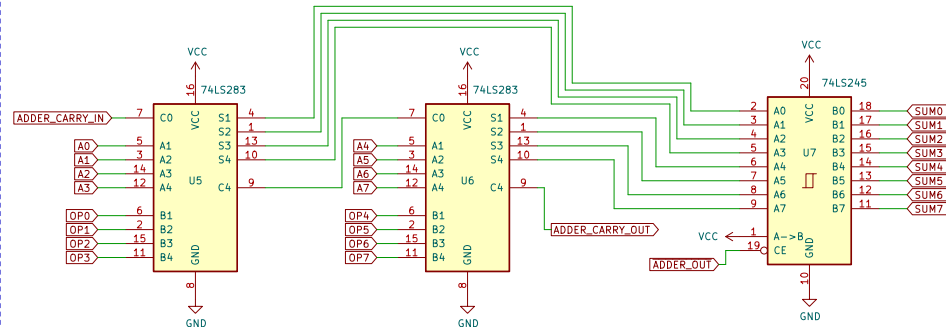
Arithmetic Carry In Logic



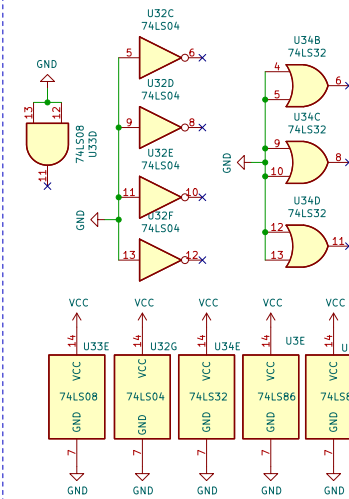
Operand Multiplexer



Full Adder w/ Carry



Logic Power



ADD / SUB / ADC / SBB / INC / DEC

Sheet: /

File: Arithmetic.kicad_sch

Title: Arithmetic Module

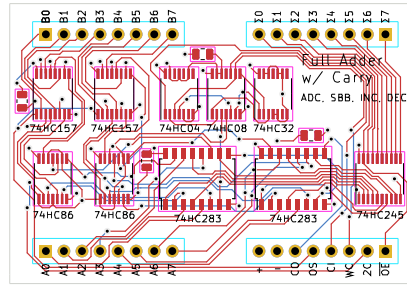
Size: User

Date:

KiCad E.D.A. kicad (6.0.0-0)

Rev: 3

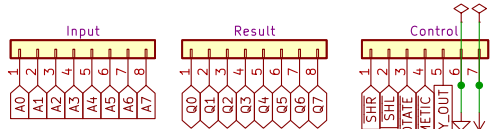
Id: 1/1



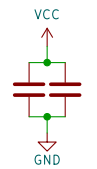
Shift & Rotate Module

Logical / Arithmetic

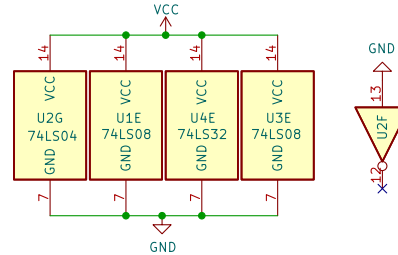
Connections



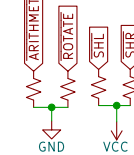
Cap's



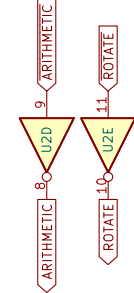
Logic Power



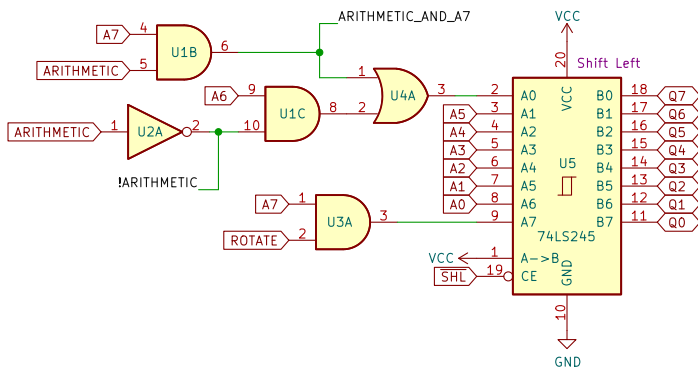
Pull U/D



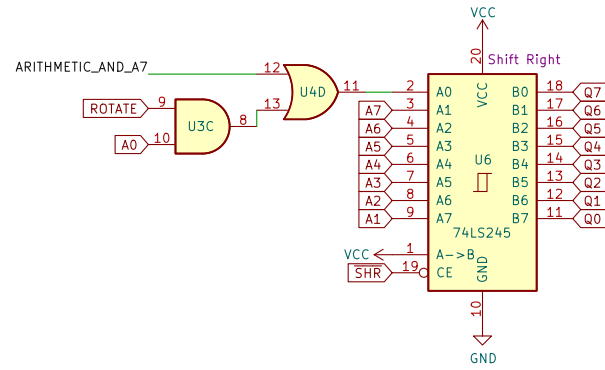
MAKE ACTIVE LOWS



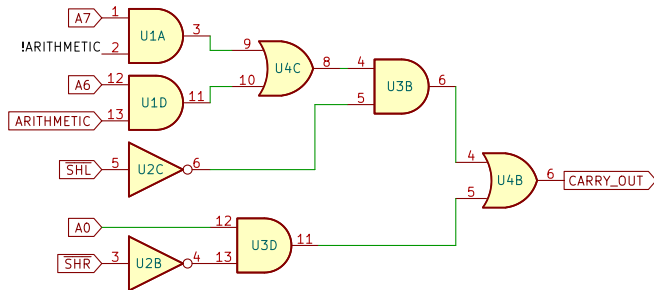
SHIFT LEFT



SHIFT RIGHT



CARRY OUT LOGIC



Rotate Enable / Carry Out

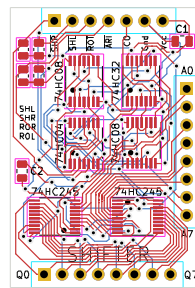
theWickedWebDev/8-bit-computer

Sheet: /
File: Shifter-v6.kicad_sch

Title: Arithmetic & Logical Shifter Module

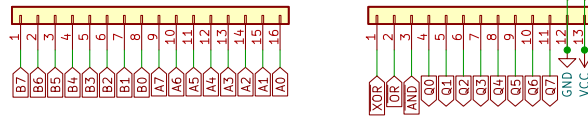
Size: A4 Date: 2022-01-16
KiCad E.D.A. kicad (6.0.0-0)

Rev: 4
Id: 1/1

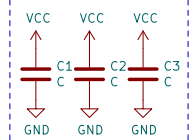


8bit Logic Gate

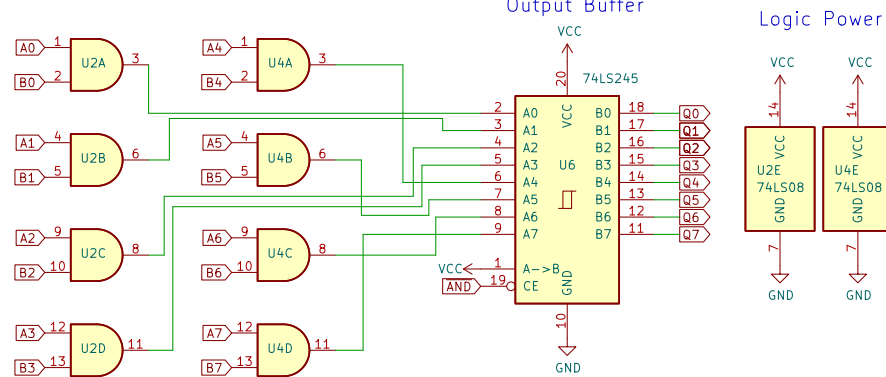
Connectors



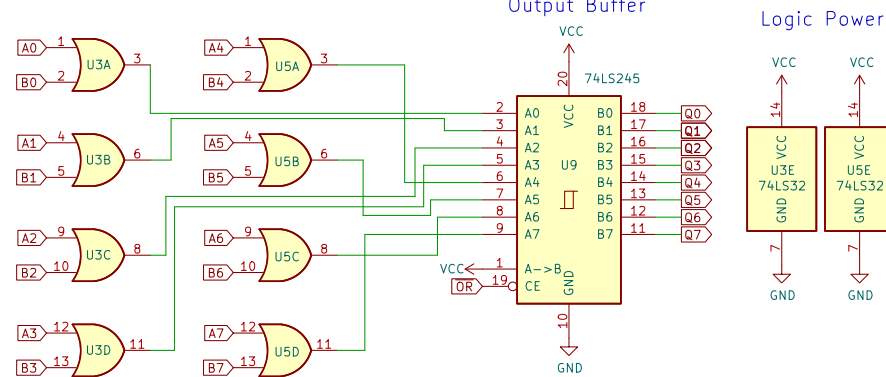
Smoothing Caps



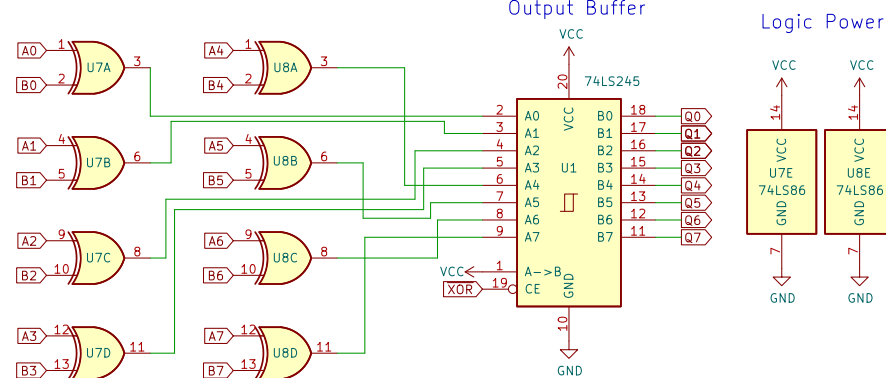
AND



OR



XOR

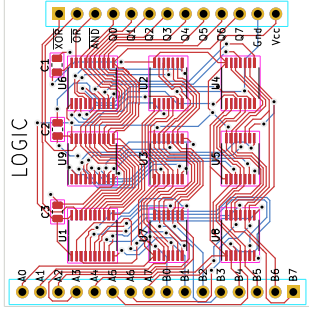


Sheet: /
File: Modules.kicad_sch

Title: 8bit Logic Gate (ALU)

Size: User Date:
KiCad E.D.A. kicad (6.0.0-0)

Rev: 3
Id: 1/1



FLAGS REGISTER

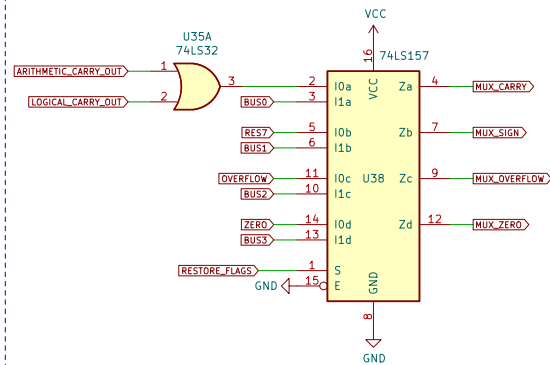
LATCH_FLAGS – A LOW signal will store the data asserted from the multiplexer into the Flags Register (FR)

- **RESTORE: LOW**, uses signals from ALU
- **RESTORE: HIGH**, uses signal asserted on data bus

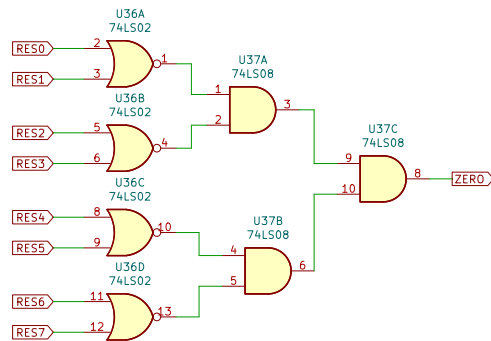
FLAG_OUT – Asserts the current flags statuses onto the Data bus, typically used to push it onto the stack to handle an ISR

Source Multiplexer

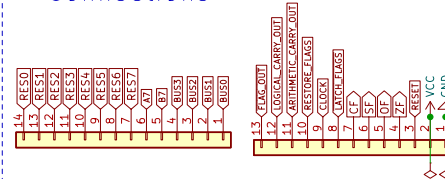
Flags come directly from ALU, or, from the flag/data bus to restore flags from the stack or another location



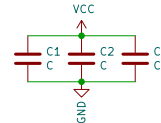
Zero



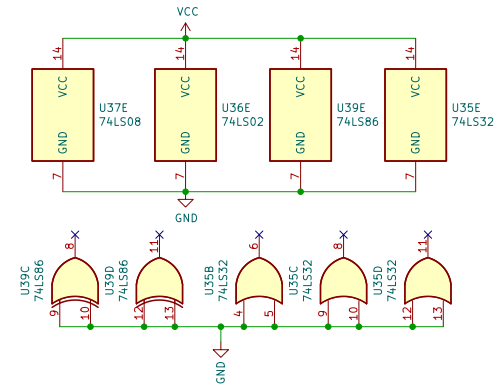
Connections



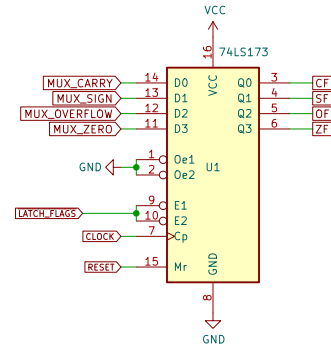
Smoothing Caps



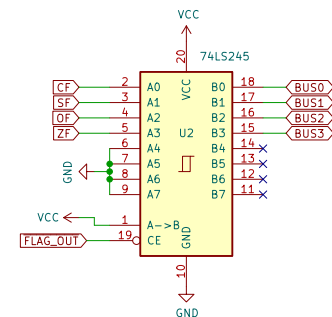
Logic Gate Power



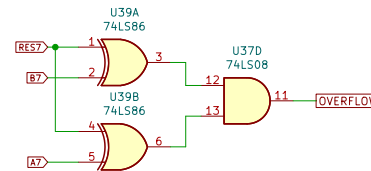
REGISTER



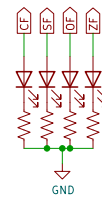
Bus Connection



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
KiCad E.D.A. kicad (6.0.0-0)

Rev: 3
Id: 1/1

