Arithmetic Logic Unit (ALU)

This portion of the CPU is responsible for all of the arithmetic, logic, and boolean functionality. It includes a *flags register* as well as a *conditional jump logic* module. Using a *6-bit control word*, and the *A and B* data busses, it can perform *22 different functions*. It is connected to the main 8-bit data bus to transfer values to the accumulator and operand registers as well as output the result of the given function. It is also used to restore the flags register during an interrupt return.

Power: 5V Vcc

Functionality

- Add w/ Carry
- Subtract w/ Borrow
- Invert
- Shift Left / Right
- Arithmetic Shift Left / Right
- Rotate Left / Right
- Compare / Test
- AND / NAND
- OR / NOR
- XOR / XNOR

Control & Instructions

ƒ	Description	Flags Affected	Hex Code
A	Asserts accumulator to the data bus		0x20
INC	Increment, A + 1	CF, SF, OF, ZF	0x25
DEC	Decrement, A – 1	CF, SF, OF, ZF	0x26
ADD	Addition, A + B	CF, SF, OF, ZF	0x21
ADC	Addition w/ Carry, A + B + Carry Flag	CF, SF, OF, ZF	0x22
SUB	Subtract, A – B	CF, SF, OF, ZF	0x23
SBB	Subtract w/ Carry, A – B – Carry Flag	CF, SF, OF, ZF	0x24
NOT	Invert A, ~A		0x30
AND	Boolean AND, A & B	SF, ZF, OF: Cleared, CF: Cleared	0x27
NAND	Boolean NAND, ~(A & B)	SF, ZF, OF: Cleared, CF: Cleared	0x35
OR	Boolean OR, A | B	SF, ZF, OF: Cleared, CF: Cleared	0x28

NOR	Boolean NOR, ~(A | B)	SF, ZF, OF: Cleared, CF: Cleared	0x38
XOR	Boolean XOR, A ^ B	SF, ZF, OF: Cleared, CF: Cleared	0x29
XNOR	Boolean XNOR, ~(A ^ B)	SF, ZF, OF: Cleared, CF: Cleared	0x39
SHL	Shift Left, A << 1	CF: Bit shifted out, SF, ZF, OF	0x2a
SHR	Shift Right, A >> 1	CF: Bit shifted out, SF, ZF, OF	0x2b
ASL	Signed Arithmetic Shift Left, A * 2	CF: Bit shifted out, SF, ZF, OF	0x2c
ASR	Signed Arithmetic Shift Riht, A / 2	CF: Bit shifted out, SF, ZF, OF	0x2d
ROR	Rotate Right	CF: Bit shifted out, SF, ZF, OF	0x2e
ROL	Rotate Left	CF: Bit shifted out, SF, ZF, OF	0x2f
CMP	Compare (SUB) - Without flags		0x23*
TST	Test (AND) - Without flags		0x27*

Accumulator and Operand Registers (8-bit)

- Latch Accumulator – Loads the value present on the data bus into the Accumulator (A) Register to be used by the various ALU functions.

- Latch Operand – Loads the value present on the data bus into the Operand (B) Register to be used by the various ALU functions.

Arithmetic Module

This module is built with two 4-bit ***Full Adders*** to perform addition on two 8-bit values. It uses **two's-complement** for Subtraction by XOR'ing the operand (B) with 0xff and adding 1. The accumulator can be incremented or decremented.

Arithmetic Control Truth Table

Instruction	Mux	Two's Comp	Carry In
A + B	0	0	0
A – B	0	1	1
A + B + Carry	0	0	Carry Flag Value
A – B – Carry	0	1	Carry Flag Value
INC (A + 1)	1	1	1
DEC (A – 1)	1	0	0

Arithmetic Examples

Instruction	A	B	CF	Result
ADD	0b0011	0b0001	n.c.	0b0100
ADC	0b0011	0b0001	1	0b0101
SUB	0b0011	0b0001	n.c.	0b0010
SBB	0b0011	0b0001	1	0b0001
INC	0b0011	n.c.	n.c.	0b0100
DEC	0b0011	n.c.	n.c.	0b0010

Logic Gate Module

This module is responsible for executing ***logical bitwise operations***. It contains only the AND, OR, and NOR functions. To acheive NAND, NOR, and XNOR the result is inverted at the ALU control unit.

Logic Gate Examples

Instruction	A	B	Result
AND	0b0101	0b0011	0b0001
NAND*	0b0101	0b0011	0b1110
OR	0b0101	0b0011	0b0111
NOR*	0b0101	0b0011	0b1000
XOR	0b0101	0b0011	0b0110
XNOR*	0b0101	0b0011	0b1001

Logic Gate Examples

**1 – Uses same function, however, the invert control bit is set to active. ie. and => nand*

Shift & Rotate Module

This module provides the ability to shift bits to the left or right. Can be done ***logically*** or ***arithmetically***. The logical shift is unsigned (ignores MSB), whereas Arithmetic Shift is signed, and maintains the most significant bit. This is the equivalent of multiplying/dividing by 2.

Shift & Rotate Truth Table

Instruction	SHR	SHL	Rotate	Arithmetic
SHL	1	0	1	1
SHR	0	1	1	1
ASL	1	0	1	0
ASR	0	1	1	0
ROR	0	1	0	1
ROL	1	0	0	1

Shift & Rotate Examples

Instruction	Operand	Result
SHL	0b10001001	0b00010010
SHR	0b10001001	0b01000100
ASL	0b10001001	0b10010010
ASR	0b10001001	0b11000100
ROR	0b10001001	0b11000100
ROL	0b10001001	0b00010011

Conditional Jump Logic

This module provides all of the conditions for calculating ***jumps*** based on ALU Flags. It uses a 4-bit control bus with enable, which is connected to the main CPU's microcode control logic, the ALU flags and outputs a single control line which is LOW if a jump is active.

Instruction	Description	Flags	Hex Code
jp	Jump	n/a	0x0
jle / jng	Jump if Less Than or Equal / Jump Not Greater	ZF = 1 or SF <> OF	0x1
jg / jnle	Jump if Greater / Jump if Not Less Than or Equal	ZF = 0 and SF = OF	0x2
jge / jnl	Jump if Greater Than or Equal / Jump if Not Lower	SF = OF	0x3
jl / jnge	Jump if Less Than / Jump if Not Greater Than or Equal	SF <> OF	0x4
ja / jnbe	Jump if Above / Jump if Not Below	CF = 0 and ZF = 0	0x5
jbe / jna	Jump if Below / Jump if Not Above	CF = 1 or ZF = 1	0x6
jnb / jae / jnc	Jump if not below / Jump if above or equal / Jump if not carry	CF = 0	0x7

jb / jnae / jc	Jump if below / Jump if not above or equal / Jump if carry	CF = 1	0x8
jne / jnz	Jump if not equal / Jump if not zero	ZF = 0	0x9
je / jz	Jump if equal / Jump if zero	ZF = 1	0xa
jns	Jump if not sign	SF = 0	0xb
js	Jump if sign	SF = 1	0xc
jno	Jump if not overflow	OF = 0	0xd
jo	Jump if overflow	OF = 1	0xe

Parts & Components List

Digital Logic, IC's

Part #	Description	Qty	Datasheet	Link
74HC574	8-Bit, Edge-Triggered, flip-flops	2	DATA	Mouser
74HC245	8-bit, Tri-State Transceiver	9	DATA	Mouser
74HC640	8-bit, Tri-State Transceiver w/ inverted outputs	1	DATA	Mouser
74HC154	4 to 16 line, decoder	2	DATA	Mouser
74HC00	Quad, 2-input NAND gates	2	DATA	Mouser
74HC04	6, 1-input, NOT (invert) gates	5	DATA	Mouser
74HC08	Quad, 2-input AND gates	11	DATA	Mouser
74HC21	Dual, 4-input AND gates	1	DATA	Mouser
74HC32	Quad, 2-input OR gates	7	DATA	Mouser
74HC86	Quad, 2-input XOR gates	6	DATA	Mouser
CD4072	Dual, 4-input OR gates	2	DATA	Mouser

74LS283	4-bit Binary Adder w/ Fast Carry	2	DATA	Mouser
Note: This is LS series, which has a HIGH output voltage of 3.6V, which feeds a 74HC245 that has an input HIGH voltage of 3.15V (at 4.5V VCC). I should have used 74HC283, however, it seems to be working fine for me - but this is def a point to evaluate with any issues				
74HC157	Quad 2-input multiplexer	3	DATA	Mouser
74HC173	Quad D-Type Flip-Flop, ThreeState	1	DATA	Mouser
	Polarized, Electrolyic Capacitor	1		
	Ceramic Capacitor	14		
	Resistor (LED)	14		
	Resistor (Pull U/D)	9		
	Light Emitting Diodes (LED)	14		
	Header Pins	-		