Brian Eckam
Amrit Gautam

Comp Org
Project1

Instruction Set Architecture using 16 bits

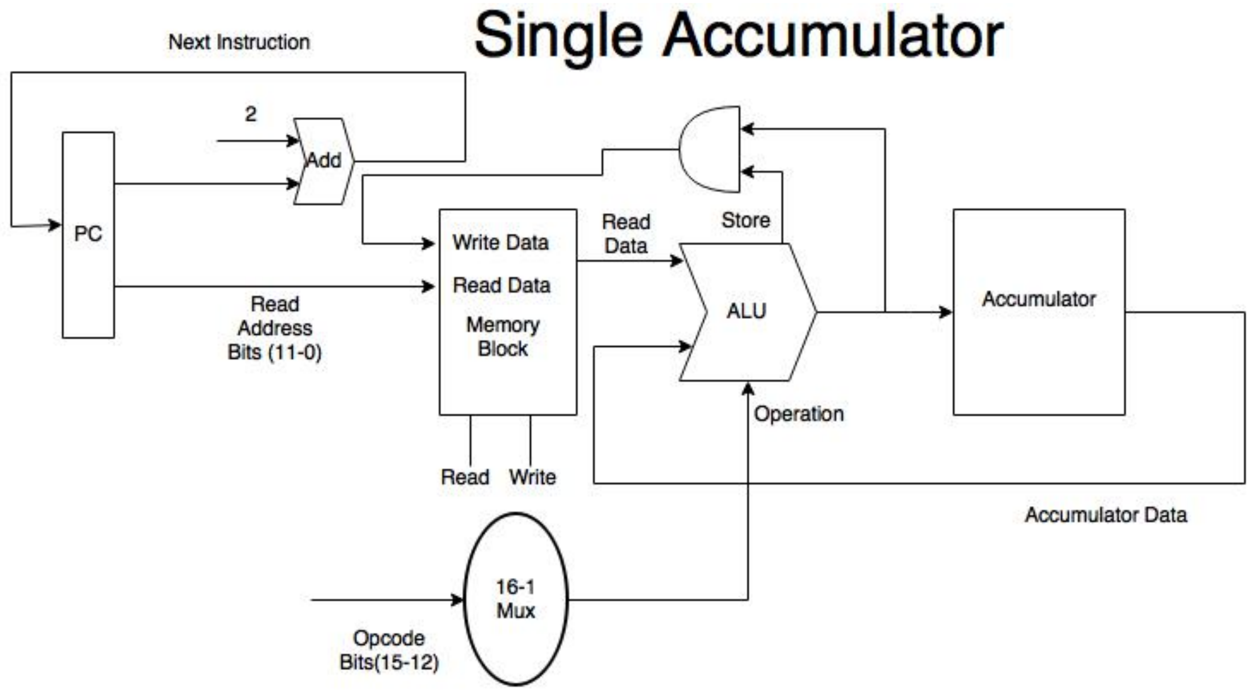


Figure 1: Single Accumulator ISA

In the single accumulator ISA the user has 16 bits to work with. The first four bits are the Opcode. The bottom twelve bits are the address of the operand. x

bits (15 downto 12)	bits (11 downto 0)
Opcode	Op1 Address

Figure 2: Instruction Format (encoding)

Code Sequence:

A = ((B + C) – D)*E ÷ F

1) Load B *Acc* ⟵ *Mem[B]*

2) Add C Acc ← Acc + C

3) Subtract D Acc ← Acc - D

4) Multiply E Acc ← Acc * E

5) Divide F Acc ← Acc ÷ F

6) Store A Acc → Mem[A]

Instruction	*RTN*
ADD (0000)	Acc ← Acc + Mem[Op1]
SUB (0001)	Acc ← Acc – Mem[Op1]
MULT (0010)	Acc ← Acc * Mem[Op1]
DIV (0011)	Acc ← Acc ÷ Mem[Op1]
AND (0100)	Acc ← Acc & Mem[Op1]
OR (0101)	Acc ← Acc | Mem[Op1]
LOAD(1110)	Acc ← Mem[Op1]
STORE(1111)	Acc → Mem[Op1]

Table 1: Instruction => RTN

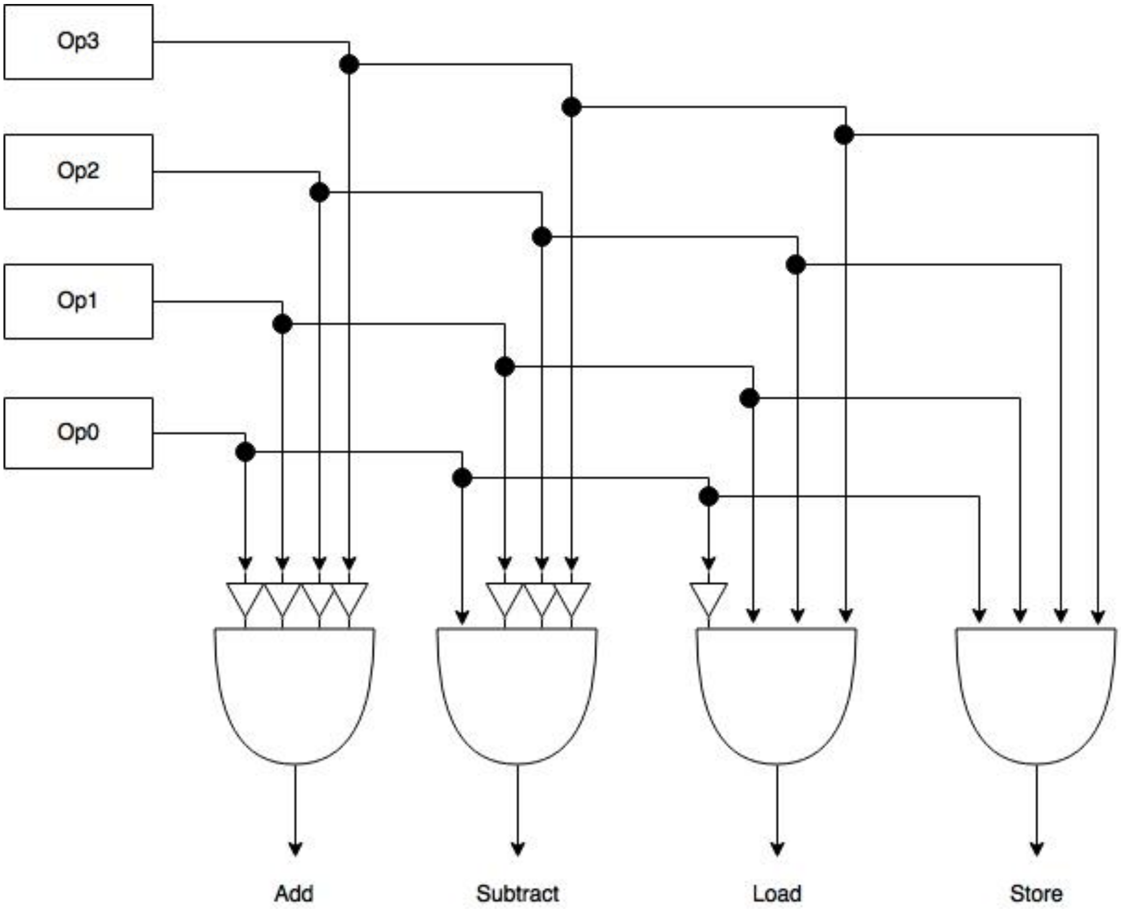
To implement each instruction, a 16-to-1 mux is needed. The opcode will specify which operation must be done at each instruction. The ALU will get this operation along with the accumulator data and the data from memory from the instruction. Four control signals are used: read, write, store and operation. In order to store the information back in memory a 2 input and-gate is used. If the ‘store’ signal = ‘1’, the output from the ALU will be put back into memory. More operations could be added than the ones that are specified above.

Control Signals:

Instruction	Read	Write	Store	Operation
Add	1	0	0	0000
Subtract	1	0	0	0001
Multiply	1	0	0	0010
Divide	1	0	0	0011
And	1	0	0	0100
Or	1	0	0	0101
Load	1	0	0	1110
Store	0	1	1	1111

Table 2: Control Signals

Programmable Logic Array:



All other operations have similar behavior

Figure 3: PLA of Accumulator

Information Size:

Op Code	4 bits
Read Address	12 bits
Write Address	12 bits
Read Data	16 bits
Accumulator Data	16 bits
Next Instruction	16 bits

Table 3: Size of Information Signals

Estimated Cycle:

To calculate the clock cycle each piece of hardware needs to be accounted for. The instruction fetch takes 2 ns. The ALU operation takes 2 ns. Pushing to memory takes 2 ns. This gives us an estimated clock cycle of 6 ns.