Step 1: - initialization:

* Set up the initial values of the RAM, program counter (PC) and other registers.
* Initialize the control signals and ensure the system is in a known state.

Step 2: -

* The program counter (PC) contains the address of the next instruction.
* The output of program counter is the input to the select lines to ram.
* Use the PC to access the memory location in RAM pointed to by the PC.
* Since select line of 8 sixteen by 1 mux are same so Q0 to Q7 values of a 74hc595 register is loaded to IR. The content of the addressed memory location is loaded into the instruction register (IR).
* Increment the PC to prepare for the next instruction.

Step 3: -

* The upper 4 bits is the instruction in the IR that is decoded using Arduino as sequencer controller to determine the operation to be performed.
* Control signals are generated based on the decoded instruction to coordinate the subsequent steps of the cycle.

Step 4: - Decoding of Instructions

- The content of the accumulator (AC) is stored in the memory location specified in the instruction.

- The content of the memory location specified in the instruction is loaded into the B register. The binary adder/subtracter adds or subtracts the content of the B register from the accumulator, and the result is stored back in the accumulator.

- The memory address specified in the instruction is loaded into the program counter (PC), causing a jump to that address.

- The memory address specified in the instruction is loaded into additional register and compared with the contents of accumulator the output is given to Arduino controller.

- if comparison is true then the memory address specified in the instruction is loaded into the program counter (PC), causing a jump to that address.

Step 5: - Repeat

Go back to the fetch stage, where the PC is incremented, and the next instruction is fetched from memory.

This step-wise breakdown provides an overview of how the SAP-1 architecture, with the specified components, performs the fetch-decode-execute cycle.