Date: 27-03-2023 **Name:** Ashvath S.P **Reg No**: 2162014

COA LAB
Experiment – 7

Write a program in Assembly Language to perform Linear Search, Binary Search.

Program no. 7.1 (Linear Search)

Algorithm:

1. Set the value of the key to be searched in register R00.
2. Initialize the values of registers R01 to R05 with the array of values to be searched.
3. Perform a linear search by iterating through the array of values and comparing each value with the key using the CMP instruction.
4. If the value in any of the registers R01 to R05 matches the key, jump to the "keyfound" label and set the value of register R10 to 1.
5. The "keyfound" label sets the value of register R10 to 1 and halts the program.

Assembly Language code:

```
MOV #15, R01 ;Store value of 15 in register R01
MOV #7, R02 ;Store value of 7 in register R02
MOV #11, R03 ;Store value of 11 in register R03
MOV #12, R04 ;Store value of 12 in register R04
MOV #9, R05 ;Store value of 9 in register R05
MOV #7, R00 ;Store value of 7 in register R00
CMP R00, R01 ;Compare value of register R01 with value of register R00
JEQ $keyfound ;If register R01 value is equal to R00, jump to the ' keyfound' label
CMP R00, R02 ;Compare value of register R02 with value of register R00
JEQ $keyfound ;If register R02 value is equal to R00, jump to the ' keyfound' label
CMP R00, R03 ;Compare value of register R03 with value of register R00
JEQ $keyfound ;If register R03 value is equal to R00, jump to the ' keyfound' label
CMP R00, R04 ;Compare value of register R04 with value of register R00
JEQ $keyfound ;If register R04 value is equal to R00, jump to the ' keyfound' label
CMP R00, R05 ;Compare value of register R05 with value of register R00
JEQ $keyfound ;If register R05 value is equal to R00, jump to the ' keyfound' label
HLT  ;Halts the simulator
keyfound: ;Label for identifying the key value
MOV #1, R10  ;Store value of 1 in register R10
HLT  ;Halts the simulator
```

Result:

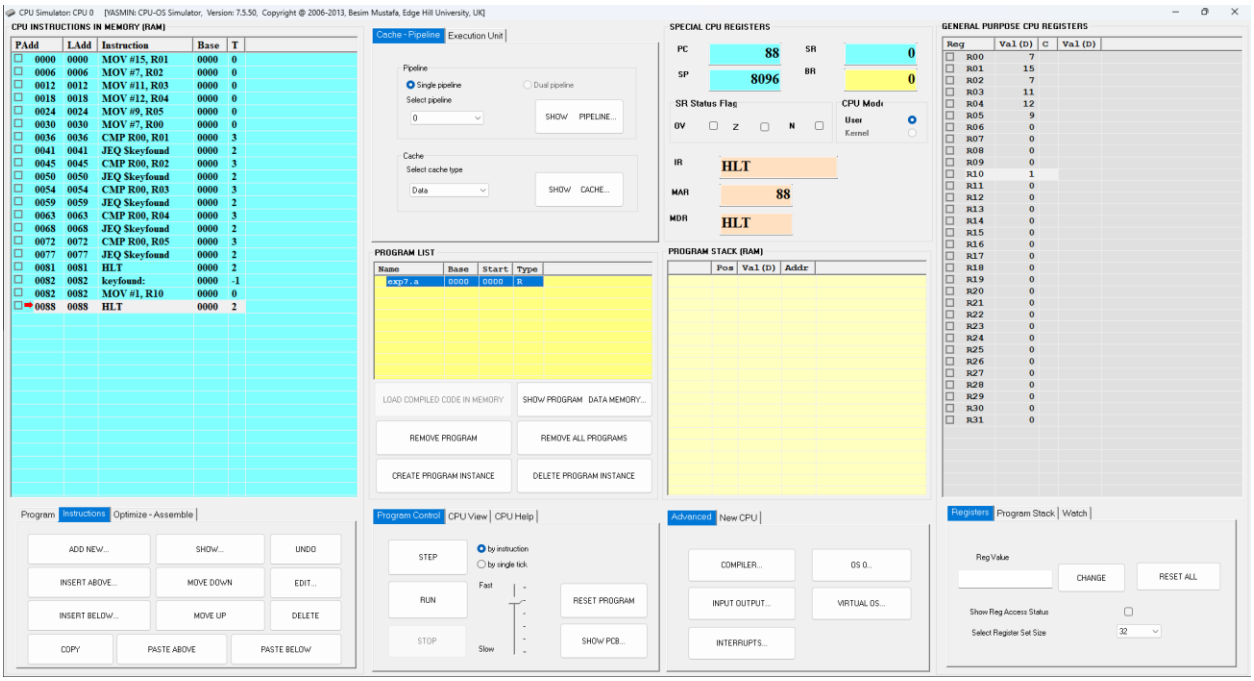

Fig7.1.a: CPU Simulator Window (key found)

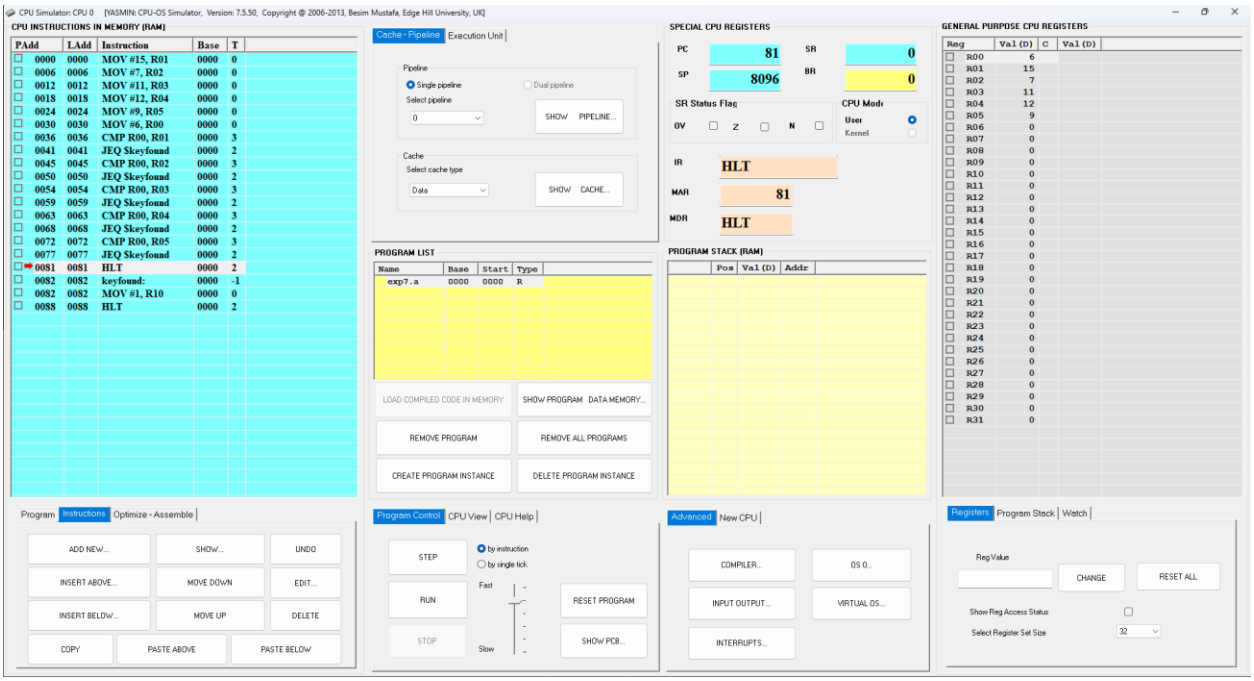

Fig.7.1.b: CPU Simulator Window (key not found)

Program no. 7.2 (Binary Search)

Algorithm:

1. Set the value to be searched in a register (R00).
2. Initialize the array of values to be searched in ascending order (e.g., in registers R01 to R05).
3. Initialize the lower bound of the search range (R06) to 0 and the upper bound (R07) to the index of the last element in the array.
4. Calculate the initial midpoint as the average of the lower and upper bounds.
5. Enter a loop to perform the binary search:
    a. Compare the value at the current midpoint with the key (R00) using CMP.
    b. If the midpoint value is greater than the key, set the new upper bound to the midpoint - 1 and recalculate the midpoint.
    c. If the midpoint value is less than the key, set the new lower bound to the midpoint + 1 and recalculate the midpoint.
    d. If the midpoint value is equal to the key, set the value of a register (e.g., R10) to 1 and halt the program.
6. Continue the loop until either the key is found or the search range is exhausted (i.e., lower bound > upper bound).
7. If the key is not found, the program will halt without setting the value of the register to 1.

Assembly Language code:

```
; Move constants into registers
MOV #3, R01 ; Move 3 into register R01
MOV #5, R02 ; Move 5 into register R02
MOV #7, R03 ; Move 7 into register R03
MOV #10, R04 ; Move 10 into register R04
MOV #12, R05 ; Move 12 into register R05
MOV #15, R06 ; Move 15 into register R06
MOV #17, R07 ; Move 17 into register R07
MOV #12, R00 ; Move 12 into register R00

; Compare values and jump to appropriate labels
CMP R04, R00 ; Compare value in register R04 with value in register R00
JEQ $FOUND ; Jump to label $FOUND if they are equal
JGT $GREATER1 ; Jump to label $GREATER1 if R04 is greater than R00
JMP $LESSER1 ; Jump to label $LESSER1 if R04 is less than R00

GREATER1
CMP R06, R00 ; Compare value in register R06 with value in register R00
JEQ $FOUND ; Jump to label $FOUND if they are equal
JGT $GREATER2 ; Jump to label $GREATER2 if R06 is greater than R00
```

```
JMP $LESSER2 ; Jump to label $LESSER2 if R06 is less than R00

LESSER1
CMP R02, R00 ; Compare value in register R02 with value in register R00
JEQ $FOUND ; Jump to label $FOUND if they are equal
JGT $GREATER3 ; Jump to label $GREATER3 if R02 is greater than R00
JMP $LESSER3 ; Jump to label $LESSER3 if R02 is less than R00

GREATER2
CMP R07, R00 ; Compare value in register R07 with value in register R00
JEQ $FOUND ; Jump to label $FOUND if they are equal
JMP $NOTFOUND ; Jump to label $NOTFOUND

LESSER2
CMP R00, R05 ; Compare value in register R00 with value in register R05
JEQ $FOUND ; Jump to label $FOUND if they are equal
JMP $NOTFOUND ; Jump to label $NOTFOUND

GREATER3
CMP R00, R03 ; Compare value in register R00 with value in register R03
JEQ $FOUND ; Jump to label $FOUND if they are equal
JMP $NOTFOUND ; Jump to label $NOTFOUND

LESSER3
CMP R00, R01 ; Compare value in register R00 with value in register R01
JEQ $FOUND ; Jump to label $FOUND if they are equal
JMP $NOTFOUND ; Jump to label $NOTFOUND

FOUND
MOV #1, R10 ; Move 1 into register R10
HLT ; Halt program

NOTFOUND
MOV #1, R11 ; Move 1 into register R11
HLT ; Halt program
```

Result:

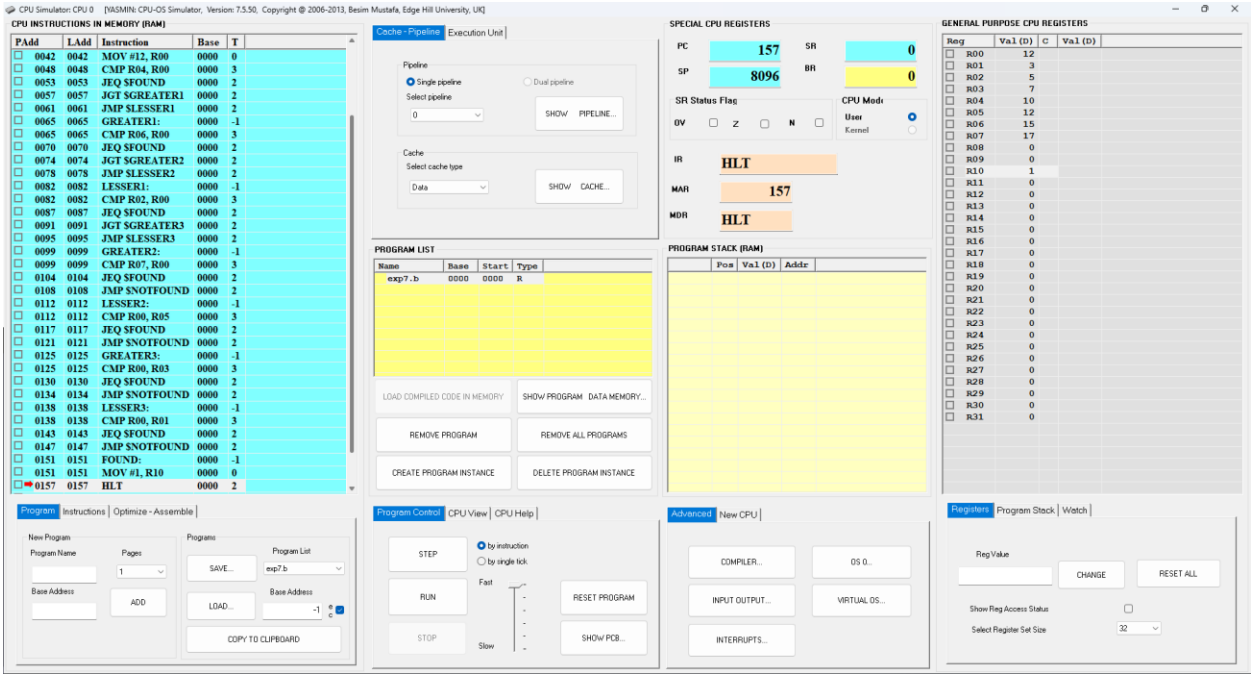

Fig7.2.a: CPU Simulator Window (key found)

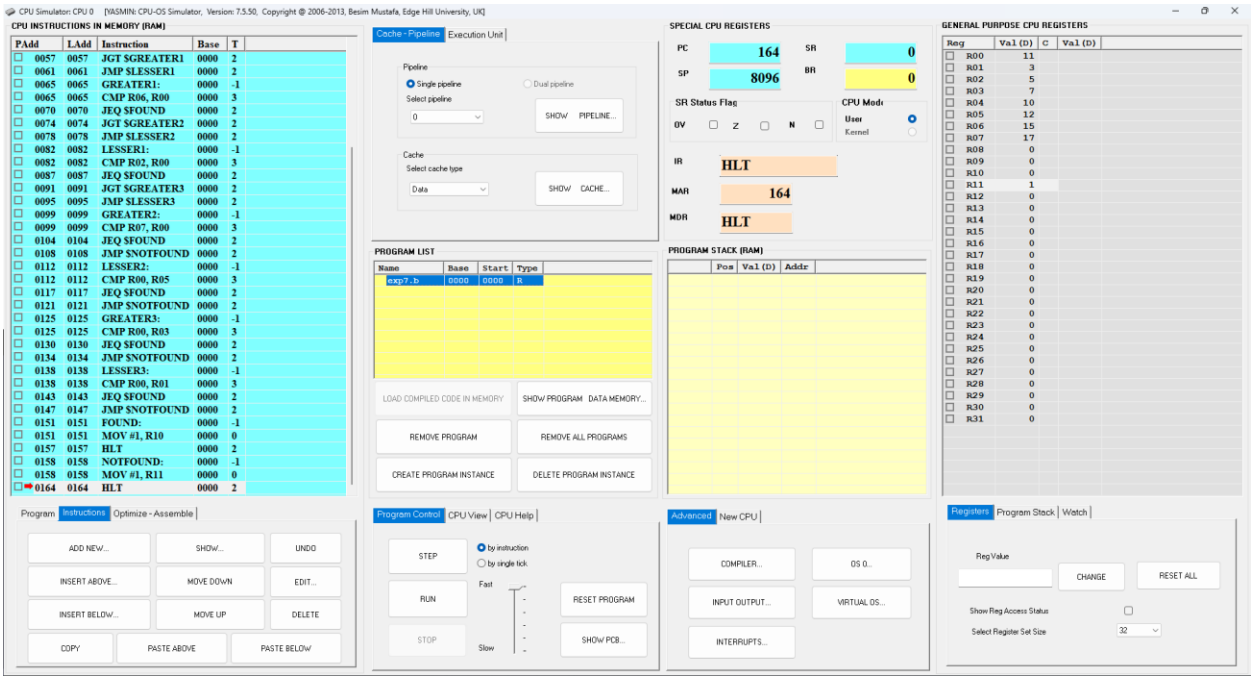

Fig.7.2.b: CPU Simulator Window (key not found)