Final report on how our microprocessor works

Our device should be able to deal with the memory houses in the same way that our main system exchanges with the program code, i.e. the three stages of fetch, decode, and execute, but this time the work must be done in software form.

1- For address, we put a button that when the user presses it, then the device has the display status, and when the user enters the desired house, the information of that house is displayed, and when the entered numbers are more, the numbers that were entered at the beginning exit from the other side, we also have to start working with the user program from the front houses to prevent the main code from interfering with the memory required by the program.

After the address where the data is written is determined, now we specify the data itself. When we press DATA, we write the data in the location of the address that was known, which is either Operand or Upcode.

In order to save that one house should be moved up or down, we use two positive symbols to move up and negative to move down and change the contents of that house.

A few notes about the code:

To start the work, PC must be entered and the others will be ignored. Enter two numbers, which are the starting address. After entering the second number, the starting address of the code will be shown on two seven segments for a few seconds, and the rest will be ignored again. The starting address must be EXE. Again, any other key that is pressed will have no effect, then the two numbers that are pressed are the first command of the code.

After entering the second number for a few seconds, the starting address of the code will be displayed on the two seven segments. After showing the opcode, you must press EXE to select the next address: press the + key, and immediately after that, the next address will appear on the seven. The segment is shown or the key is pressed, after which the previous address is shown on the seventh segment for a few seconds. In the above two cases, there is no need to press EXE after showing the address.

You can enter two numbers and after displaying it on the seven segment EXE, you can press the END key, which means that entering the code is finished and the entered codes will be executed. Press END 00. After pressing END, a number will be shown on the seven segment, which is the result of code calculations, and if 88 is shown, it means that one of the entered opcodes was wrong.

The pc button is for when data is selected in the address mode, and as a result, the user's program will start from there

Another problem with this program is that the commands of the PC are changed in such a way that the information is lost. We know this from the class and one of the exercises. We can use the stack memory and the PC is saved like this and will not be lost, but the design of the 8051 is limited and cannot be used. If the ram memory is considered for that, a bootloader must be used (TE's suggestion) to write in the ROM and identify the RAM memory as memory.

One of the solutions to solve the input problem is the same as shown, that is, the use of 2 keyboards, where we have shown the values assigned to each:

```
MOV R1, #000H ; the key is 0

MOV R1, #008H ; the key is 8

MOV R1, #010H ; the key is PC


MOV R1, #020H ; The key is F1

MOV R1, #001H ; the key is 1


MOV R1, #009H ; the key is 9

MOV R1, #030H ; the key is ADR

MOV R1, #040H ;

MOV R1, #002H ; the key is 2

MOV R1, #00AH ; the key is A
MOV R1, #050H ; the key is PLUS

MOV R1, #060H ; The key is F3

MOV R1, #003H ; the key is 3

MOV R1, #00BH ; the key is B

MOV R1, #070H ; the key is MINUS

MOV R1, #080H ; The key is F4

MOV R1, #004H ; the key is 4

MOV R1, #00CH ; the key is C

MOV R1, #090H ; the key is DATA

MOV R1, #0A0H ; The key is F5

MOV R1, #005H ; the key is 5

MOV R1, #00DH ; the key is D

MOV R1, #0B0H ; the key is MOD

MOV R1, #0C0H ; The key is F6

MOV R1, #006H ; the key is 6

MOV R1, #00EH ; the key is E

MOV R1, #0D0H ; the key is EXE

MOV R1, #0E0H ; The key is F1

MOV R1, #007H ; the key is 7

MOV R1, #00FH ; the key is F

MOV R1, #0F0H ; the key is END

MOV R1, #0F1H ; The key is F8
```

Because we have a lack of ports, we control the structure by using two BDLs

2- Possible challenge

We use RAM to store the user's program and determine the range for the main code of the program so that the user has the ability to write his program from that address onwards. To run the user's program, we can look at the entire subroutine and press the execute button. Call the program to the specific starting address of the user program and before this, fill the user program with Opcode related to RET so that after the user program is finished, the program will return to its process. Now, none of the conditional commands have any problem, and there is also a problem for using the Call command.
It doesn't happen, the only difference is that instead of starting from the first house of the stack, the program will start from the second house of the stack.

There are a group of commands that disrupt the linear process of the program and create a branch. By using the stack, the operation of these commands can be simulated, and since the stack is independent from the main code, it is not necessary to define a virtual stack.