

XMUT202 Digital Electronics

Tutorial 5: Stack (Solution)

A. Stacks in 8051

1. What is a stack? [2 marks]

Answer

The stack is a section of RAM used by the CPU to store information temporarily. This information could be data or an address. The CPU needs this storage area since there are only a limited number of registers.

2. What register is used for stack programming? [2 marks]

Answer

The register used to access the stack is called the SP (stack pointer) register. The stack pointer is 8 bits wide and it can take values of 00 to FFH.

3. What is the default memory address of stack pointer? [2 marks]

Answer

Locations 08H to 1FH is used by default for the stack, in this case the upper limit of stack is 1FH.

When the 8051 is powered up, the SP register contains value 07H. This means that RAM location 08H is the first location used for the stack by the 8051 microcontrollers.

4. What is the relationship between the stack pointer and the register Bank 1 in the 8051 microcontrollers? [2 marks]

Answer

Since SP = 07 when the 8051 microcontroller is powered up, the first location of the stack is RAM location 08, which also belongs to register R0 of register bank 1.

In other words, register bank 1 and the stack are using the same memory space. If in a given program we need to use register banks 1 and 2, we can reallocate another section of RAM to the stack.

5. What do you do when you need more space to store for your stack in the 8051 microcontrollers? [2 marks]

Answer

If in a given program, we need more than 24 bytes (08 to 1FH = 24 bytes) of stack, we can change the SP to point to RAM locations 30 – 7FH. This is done with the instruction “MOV SP, #xxH”.

6. Describe two mechanisms for interacting with the content in the stack? [2 marks]

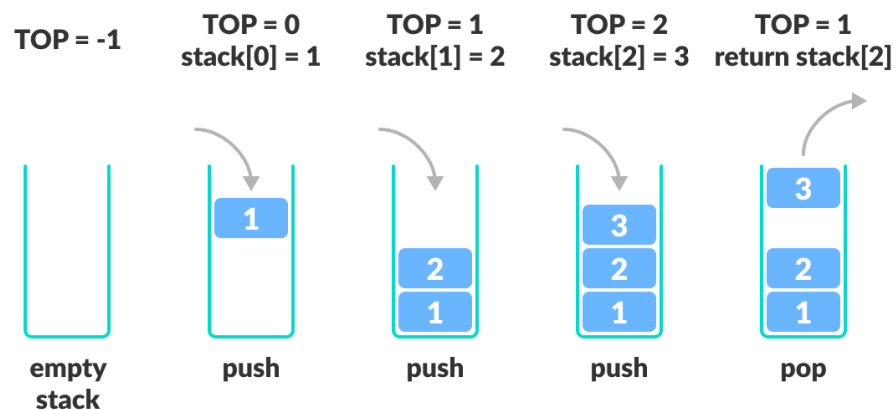
Answer

The storing of a CPU register in the stack is called a PUSH, and pulling the contents off the stack back into a CPU register is called a POP.

7. Describe the process involved in storing and retrieving items to and from the stack. [4 marks]

Answer

The processes in storing and retrieving items to and from the stack are as illustrated in the figure below.



The operations have the following steps listed as follows:

1. A pointer called TOP is used to keep track of the top element in the stack.
2. When initializing the stack, we set its value to -1 so that we can check if the stack is empty by comparing $TOP == -1$.
3. On pushing an element, we increase the value of TOP and place the new element in the position pointed to by TOP.
4. On popping an element, we return the element pointed to by TOP and reduce its value.
5. Before pushing, we check if the stack is already full
6. Before popping, we check if the stack is already empty

B. Stack Programming

1. Describe the mechanism of storing data to the stack in 8051. Write a short assembly language code that stores data onto the stack [4 marks]

Solution

When data is stored onto the stack, the stack pointer is incremented by 1, and the data is stored at the new location pointed to by the stack pointer.

For example, store a value 55H into a register and store it into stack.

Program:

```
; Example of pushing data onto the stack
MOV A, #55H ; Load A with 55H
PUSH ACC ; Push the contents of A onto the stack
```

2. Describe the mechanism of retrieving data from the stack in 8051. Write a short assembly language code in 8051 that retrieves data from the stack. [4 marks]

Solution

Retrieving (popping) data from the stack is the reverse operation of storing (pushing).

For example, the stack pointer is decremented by 1, and the data at the location pointed to by the stack pointer is copied to the specified register i.e. A.

Program:

```
; Example of popping data from the stack
POP ACC ; Pop the top of the stack into A
```

3. Write a code that moves values 0x25H and 0x45H to registers R1 and R4 and store and retrieve them to the stack to the registers R3 and R6 respectively. Describe line-by-line the code in the program. [4 marks]

Solution

Program:

```
MOV R1, #25H
MOV R4, #45H
PUSH R1
```

PUSH R4

POP R3

POP R6

Programming descriptions:

- the instruction "PUSH R1" increments SP to 08H and pushes register R1 onto the stack location 08H.
- the "PUSH R4" increments SP to 09H and pushes register R4 onto the stack location 09H.
- the instruction "POP R3" copies the top of stack (location 09H) to register R3 and decrements SP to 08H.
- the "POP R6" copies the top of stack (location 08H) to register R6 and decrements SP to 07H.