

Computer System Architecture

Third Lecture

The Stack

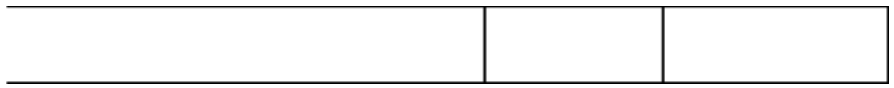
- The stack is a place in RAM where data is temporarily stored. The SS and SP registers point to that place like this: SS:SP So the SS register is the segment and the SP register contains the offset.
- There are a few instructions that make use of the stack. **POP and PUSH are the most basic ones.**
- PUSH can "push" a value on the stack and POP can retrieve that value from the stack.

The Stack

- As data is pushed onto the stack **SP decremented**.
- As data is popped off the stack into the CPU, **SP incremented**.
- **There are no instructions such as "PUSH AL" or "PUSH AH"**.

The Stack

- Assume AX=026BH BX=04E3H and SP=36H



- PUSH AX(decrement the SP by 2)

SP=36 top of the stack



SP=34

Note the operation reverses the sequence stored bytes so that 02B6 becomes 6B02.

- PUSH BX(decrement SP by 2) and stores the contents of BX in the stack as E304



SP=32

The Stack

- **POP BX:** Restores the word from where SP points in the stack(E304) to BX register and increment SP by 2(to 34) . BX now contains 04E3.
- **POP AX:** Restores the word from where SP points in the stack(6B02) to AX register and increment SP by 2(to 36) . BX now contains 026B.
- The last value added to a stack is the first one to be removed or popped from the stack so we can call it as a LIFO structure(Last Input First Output)

Using the INT instruction

- INT instruction will exit from a program, enter a DOS or BIOS routine, performs the requested function, and return to a program.
- There are different types of INT operations some of which require a function code in the AH register to request a specific action.

Using INT instruction

- BIOS Interrupt Routines
 - INT 10h :for screen operations
 - INT 16h :for keyboard operations
- DOS interrupt
 - INT 21h

for both screen and keyboard operations.

Each of the above interrupt routines performs several functions depending on **the value stored in register AH** at the time the interrupt is invoked.

INT 21H instruction

➤ INT 21H with

- Functions 01H to accept character and display it.
- functions 02H to display a character
- function 09H to display a string

INT 21H instruction

Accept a single character from keyboard and display it.

- **Step 1: Set AH =01H**
- **Step 2: Call INT 21H to display the character**
 - The ASCII of entered character will set in the AL register.

AL = the character inputted from keyboard.

• Ex:

```
MOV AH,01
```

```
INT 21H
```

INT 21h

Display a single character .(You must place the character to be displayed in DL register)

- Step 1: Set AH =02H**
- Step 2: Load the character to DL**
- Step 3: Call INT 21H to display the character**

Example:

MOV AH,02

MOV DL,41H

INT 21H

The ASCII Character Set

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

- CR = “carriage return” (MSDOS: move to beginning of line)
- LF = “line feed” (MSDOS: move directly one line below)
- SPC = “blank space”

Display a character

Example:

```
100 MOV AH, 02H
```

```
102 MOV DL, 41H
```

```
104 MOV CX,0005
```

```
107 INT 21H
```

```
109 LOOP 107
```

➤ This program will display capital letter **A** five time.

LOOP Instruction

- The **LOOP** instruction is the easiest way to repeat a block of statements a specific number of times.
- **CX** register is automatically used as a counter and is decremented each time the **LOOP** repeats.
- Its Syntax: **LOOP** *target*.
- First the **LOOP** instruction subtracts 1 from **CX**.
- Then if **CX** is not equal to zero, control transfer to *target*.

Debug Program that Print Capital letters:(A to Z).

A 100

- **100 MOV AH,02H**
- **102 MOV DL,41H**
- **104 MOV CX,001AH**
- **107 INT 21H**
- **109 INC DL**
- **10B LOOP 107**

Debug program that Print Capital letters:(A to Z).

In this program we:

- Move 02 to AH register.
- Move the ASCII code of the first character to DL register which is 41H .
- Move the total number of looping iteration to CX register which is 1AH.(1AH equal 26 decimal)
- Call INT 21H.
- Increment DL register.
- Iterate from INT 21H.

Using INT 21H to Display string

- To display a character string on screen.
- **Step 1: Set AH =09H**
- **Step 2: set DX with offset address of the string.**
- **Step 3: Call INT 21H to display the string**

Using INT 21H to Display string

➤ Example:

```
100 MOV AH, 09H
```

```
102 MOV DX, 108
```

```
105 INT 21H
```

```
107 NOP
```

```
108 DB 'your name', '$'
```

Using INT 21 H to Display string

- AH register contain **function code (09)** which tell **INT 21H** to display and **DX register contain starting address**.
- **DB** means **define byte** . Following your name string is a \$ also in quotes which tells **the INT to end the display**.
- key in R to display the registers and first instruction, and key in T commands for the two MOVs. Key in P to execute INT 21 and MY NAME IS YOUR NAME will display on the screen.