Interrupt Control Instructions

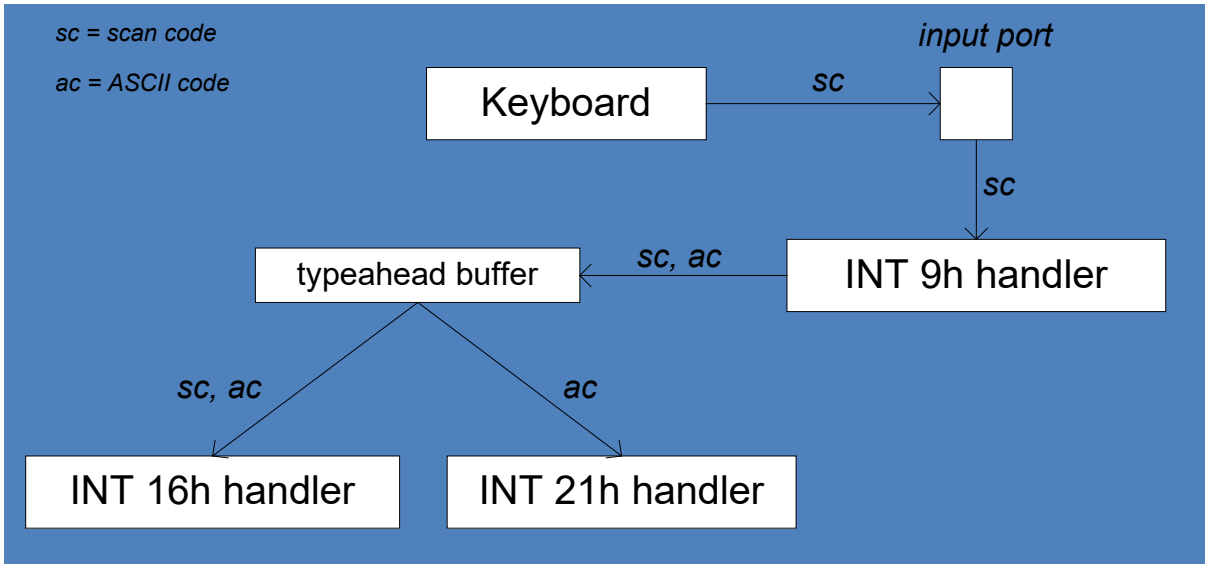
- STI – set interrupt flag
  - enables external interrupts
  - always executed at beginning of an interrupt handler
- CLI – clear interrupt flag
  - disables external interrupts
  - used before critical code sections that cannot be interrupted
  - suspends the system timer

Keyboard Input with INT 16h

- How the Keyboard Works
- INT 16h Functions

How the Keyboard Works

- Keystroke sends a scan code to the keyboard serial input port
- Interrupt triggered: INT 9h service routine executes
- Scan code and ASCII code inserted into keyboard type ahead buffer



Keyboard Flags

16-bits, located at 0040:0017h – 0018h.

Bit	Description
0	Right Shift key is down
1	Left Shift key is down
2	Either Ctrl key is down
3	Either Alt key is down
4	Scroll Lock toggle is on
5	Num Lock toggle is on
6	Caps Lock toggle is on
7	Insert toggle is on
8	Left Ctrl key is down

Bit	Description
9	Left Alt key is down
10	Right Ctrl key is down
11	Right Alt key is down
12	Scroll key is down
13	Num Lock key is down
14	Caps Lock key is down
15	SysReq key is down

INT 16h Functions

- Provide low-level access to the keyboard, more so than MS-DOS.
- Input-output cannot be redirected at the command prompt.
- Function number is always in the AH register
- Important functions:
  - set typematic (Repeat Delay and Repeat Rate) rate
  - push key into buffer
  - wait for key
  - check keyboard buffer
  - get keyboard flags

Function 10h: Wait for Key

If a key is waiting in the buffer, the function returns it immediately. If no key is waiting, the program pauses (blocks), waiting for user input.

```
.data
scanCode  BYTE ?
ASCIICode BYTE ?

.code
mov ah,10h
int 16h
mov scanCode,ah
mov ASCIICode,al
```

Function 12h: Get Keyboard Flags

Retrieves a copy of the keyboard status flags from the BIOS data area.

```
.data
keyFlags WORD ?

.code
mov ah,12h
int 16h
mov keyFlags,ax
```

Clearing the Keyboard Buffer

Function 11h clears the Zero flag if a key is waiting in the keyboard typeahead buffer.

```
L1: mov ah,11h              ; check keyboard buffer
    int 16h                 ; any key pressed?
    jz  noKey               ; no: exit now
    mov ah,10h              ; yes: remove from buffer
    int 16h
    cmp ah,scanCode         ; was it the exit key?
    je  quit                ; yes: exit now (ZF=1)
    jmp L1                  ; no: check buffer again

noKey:                      ; no key pressed
    or  al,1                ; clear zero flag
quit:
```