

98243045-فربد فولادی-

98243027-عرفان رفیعی اسکئی-

در ابتدا مقادیر PA, PB, PC, CR را ست می کنیم تا در ادامه از آن ها استفاده کنیم.

همچنین مقدار دهی مقادیر اولیه را انجام می دهیم مثل DS.

```
.MODEL SMALL
.STACK64
.DATA

PA EQU 60H
PB EQU 62H
PC EQU 64H
CR EQU 66H

.CODE
MAIN PROC FAR

    MOV AX, 4000H
    MOV DS, AX

    MOV CX, 0H

    ;Control Register
    MOV AL, 10001001B
    OUT CR, AL
```

عملکرد کلی به این صورت است که هر سطر از کی پد را به عنوان یک سطر از حافظه در نظر می گیریم و با مقدار PC و control_register تشخیص می دهیم که کجای حافظه هستیم تا row مورد نظر را انتخاب کنیم و در ادامه اعداد موجود در آن row را بخوانیم و ذخیره کنیم. در این حین اگر در سطری از حافظه نباشیم وارد busy_waiting می شویم.

```

row_1:
    MOV AL, 0FEH
    OUT PA, AL
    IN AL, PC
    MOV AH, AL

busy_wait_row1:
    IN AL, PC
    CMP AL, 0FFH
    JNE busy_wait_row1

    MOV AL, AH
    CMP AH, 0FEH
    JE ONE
    MOV AH, AL
    CMP AH, 0FDH
    JE TWO
    MOV AH, AL
    CMP AH, 0FBH
    JE THREE
    JMP row_2

```

با مقایسه کردن مقدار AH با مقادیر از پیش تعیین شده تشخیص می دهیم که وارد کدام سطر شویم با دستور جامپ به آن می رویم.

در تگ های 1 تا 9 مقدار آن عدد را در AL ذخیره می کنیم، همچنین یک index داریم که در CX نگهداری می شود و از 0 شروع می شود.

این مقدار CX را در BX می ریزیم تا در ادامه AL را در آدرس BX ذخیره کنیم.

در نهایت CX را یکی اضافه می کنیم به row بعدی می رویم.

```

one:
    MOV AL, 1H
    MOV BX, CX
    MOV [BX], AL
    INC CX
    JMP row_2

two:
    MOV AL, 2H
    MOV BX, CX
    MOV [BX], AL
    INC CX
    JMP row_2

three:
    MOV AL, 3H
    MOV BX, CX
    MOV [BX], AL
    INC CX
    JMP row_2

```

برای دکمه استار نیز مقدار صفر را در DX می ریزیم تا در show از آن استفاده کنیم.

در delay با کم کردن AX و چک کردن آن با صفر به show یا delay می رویم.

```
delay:
    dec AX
    CMP AX, 0H
    JE SHOW
    JMP delay
```

در show با چک کردن DX با CX شرط پایان را بررسی می کنیم که اگر برقرار باشد به done می رویم که AX ریست می شود و به row_1 بر می گردیم ولی اگر برابر نباشد مقدار داخل آدرس BX را در AL می ریزیم و با out در PA قرار می دهیم تا نمایش داده شود و از DX یکی کم می کنیم تا جایی که همه اعداد نمایش داده شوند و به شرط پایان برسیم.

```
show:
    cmp dx, cx
    JE done
    MOV AX, 4000H
    MOV DS, AX
    MOV BX, DX
    MOV BL, [BX]
    MOV BH, 0H
    MOV AX, 0F000H
    MOV DS, AX
    MOV AL, [BX]
    out pb, al
    inc DX

    MOV AX, 0FFFFH
```