Fig 3.8 Main (Solution)

Main program

```
START:  Move        R2, #LINE
        Store       R2, PNTR        Initialize buffer pointer.
        Clear       R2
        Store       R2, EOL         Clear end-of-line indicator.
        Move        R2, #2          Enable interrupts in
        StoreByte   R2, KBD_CONT     the keyboard interface.
        MoveControl R2, IENABLE
        Or          R2, R2, #2      Enable keyboard interrupts in
        MoveControl IENABLE, R2      the processor control register.
        MoveControl R2, PS
        Or          R2, R2, #1
        MoveControl PS, R2          Set interrupt-enable bit in PS.
        next instruction
```

1. What is the purpose of EOL?

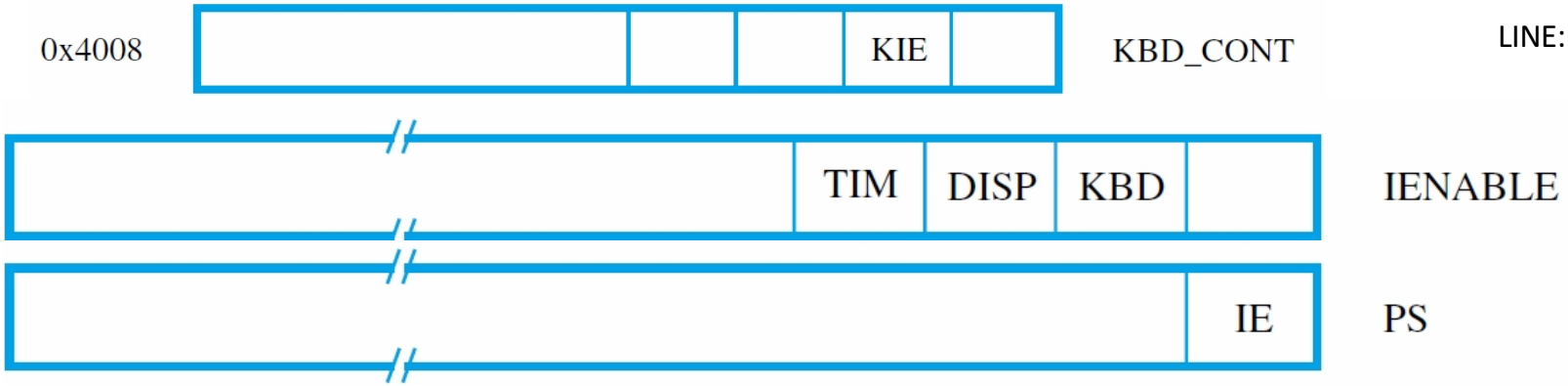
1. Indicate to other programs EOL

2. Why these EIs use different instructions?

3. Why these two Writes to Control Registers use different instructions?

3. Local KBD is memory mapped so using Store while PS is a control register in CPU

2. a) KBD local using memory mapped I/O;
b) Or the other bits so as not to lose them in control R;
c) Or the other bits so other interrupts remain EI/DI

Memory

PNTR	LINE
EOL	0
	LINE: 1st Char