Seven-Segment Display

The MDA 8086 trainer kit uses the PIO 8255, which has its ports internally connected to a **seven-segment display**. We can use binary, decimal or hexadecimal values as shown in the table below to use the seven-segment display. The 8 bits are provided to PA0 through PA7 from right to left respectively.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| dp | g | f | e | d | c | b | a | Decimal | Hex |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 80 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | F9 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 2 | A4 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 3 | B0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 4 | 99 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 5 | 92 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 6 | 82 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 7 | F8 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 80 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 9 | 90 |

Note that a **binary 0** will cause a segment to remain **on** while a **binary 1** will cause a segment to remain **off**. The dot-point value is not important.

To display a single character, all we have to do is firstly set the **mode** in the **control register**, so that the trainer kit knows we are trying to control the 8255 PPI, and secondly pass the value to the appropriate port.

MOV AL, 10000000b ; Mode set for Control Word to control 8255 PPI  
OUT 1Fh, AL ; Set mode in Control Register  
MOV AL, 10110000b ; Set the value for ‘3’  
OUT 19h, AL ; Pass value to Port A

ASM