DIGITAL CLOCK

BY
SUBATHRA S

DIGITAL CLOCK

AIM

To write an assembly language program to generate a digital clock.

APPARATUS REQUIRED

- 8085 Microprocessor kit
- Power Supply

ALGORITHM

1. Hours, minutes and seconds are initialized to zero.
2. Increment seconds and decimal adjust it, when it reaches 60, increment a minute counter.
3. Proceed the second step till minute reaches 60. When minute reaches 60, increment a hour counter.
4. Previous two steps are repeated till the clock hour reaches 24.
5. When the hour reaches 24, the entire set up is reset and starts from first step counting the seconds.
6. Continue the execution.

ASSEMBLY LANGUAGE PROGRAM

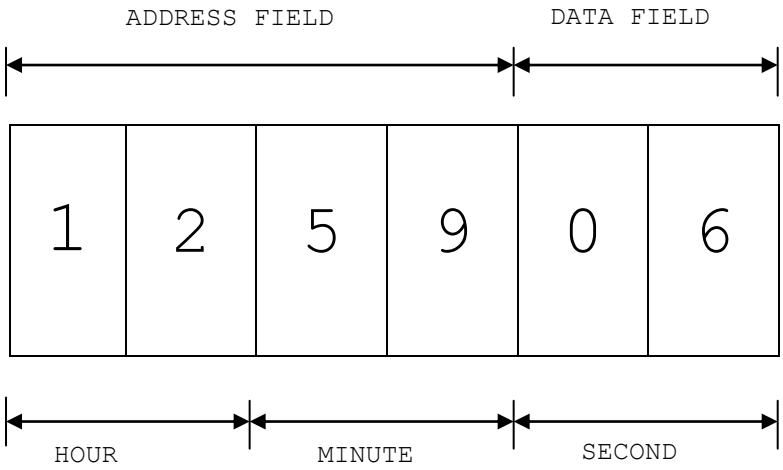
ADDRESS	LABEL	MNEMONICS	OPCODE/OPERAND	COMMENT
C000		**LXI SP,C500$_H$**	31 00 C5	Initialize the stack pointer for storage purpose
C003		**MVI B,24$_H$**	06 12	Initialize the hour to be displayed.
C005		**MVI C,59$_H$**	0E 59	Initialize the minute to be displayed
C007		**MOV A,B**	78	Store the hour at the address field of the display
C008		**STA FFF8$_H$**	32 F8 FF	
C00B		**PUSH B**	C5	Store the hour in stack since while calling the monitor routine the register and flag contents gets destroyed
C00C		**CALL UPDAD**	CD BC 06	Call the address field memory location using monitor routine
C00F		**POP B**	C1	Retrieve the stacked hour
C010		**MOV A,C**	79	Store the minute at the address field of the display
C011		**STA FFF7$_H$**	32 F7 FF	
C014		**PUSH B**	C5	Store the minute in stack since while calling the monitor routine the register and flag contents gets destroyed
C015		**CALL UPDAD**	CD BC 06	Call the address field memory location using monitor routine

C018		**POP B**	C1	Retrieve the stacked minute
C019		**JMP HOUR**	C3 07 C1	Jump to Label HOUR, to start the digital count
C100	**RESET**	**LXI SP,C500$_H$**	31 00 C5	When the 24 hours are over, the entire set up is reset i.e all the counters are initialized to 00_H , to start afresh
C103		**MVI B,00$_H$**	06 00	
C105		**MVI C,00$_H$**	0E 00	
C107	**HOUR**	**MVI A,00$_H$**	3E 00	
C109	**MIN**	**STA FFF9$_H$**	32 F9 FF	Accumulator content is stored as the second at the data field of the display for each looping
C10C		**PUSH PSW**	F5	Calculate the second using suitable delay subprogram and display it by calling the address field of display using monitor routine. Also stack the hour content present in B register
C10D		**PUSH B**	C5	
C10E		**CALL UPDDT**	CD D3 06	
C111		**CALL DELAY**	CD 00 C2	
C114		**POP B**	C1	
C115		**POP PSW**	F1	
C116		**ADI 01$_H$**	C6 01	Increment the seconds.
C118		**DAA**	27	The 8 bit number in the accumulator is adjusted to form two four bit BCD digits. This uses the auxiliary flag (internally) to perform the binary to BCD conversion. DAA is used, in order to avoid displaying of the hexcode in the digital clock.
C119		**CPI 60$_H$**	FE 60	Compare whether the second has reached 60.
C11B		**JNZ MIN**	C2 09 C1	When 60 seconds are over ,then minute counter is incremented by one.
C11E		**MOV A,C**	79	Minute is moved to accumulator
C11F		**ADI 01$_H$**	C6 01	Increment the minute
C121		**DAA**	27	Perform the binary to BCD conversion.
C122		**MOV C,A**	4F	Minute is stored in C register
C123		**STA FFF7$_H$**	32 F7 FF	Accumulator content is stored as the minute at the address field of the display for each looping
C126		**PUSH B**	C5	To display minute, call the address field of display using monitor routine, by stacking the hour content present in B register
C127		**CALL UPDAD**	CD BC 06	
C12A		**POP B**	C1	
C12B		**MOV A,C**	79	Minute is moved to

				accumulator
C12C		**CPI 60_H**	FE 60	Compare whether the minute has reached 60.
C12E		**JNZ HOUR**	C2 07 C1	When 60 minutes are over ,then hour counter is incremented by one.
C131		**MVI C,00_H**	0E 00	In order to reset the minute counter, C register is initialized to 00_H.
C133		**MOV A,C**	79	Minute is moved to accumulator
C134		**STA FFF7_H**	32 F7 FF	Accumulator content is stored as the minute at the address field of the display for each looping
C137		**PUSH B**	C5	To display minute, call the address field of display using monitor routine, by stacking the hour content present in B register
C138		**CALL UPDAD**	CD BC 06	
C13B		**POP B**	C1	
C13C		**MOV A,B**	78	Hour is moved to accumulator
C13D		**ADI 01_H**	C6 01	Increment the hour
C13F		**DAA**	27	Perform the binary to BCD conversion.
C140		**MOV B,A**	47	Hour is moved to B register
C141		**STA FFF8_H**	32 F8 FF	Accumulator content is stored as the hour at the address field of the display for each looping
C144		**PUSH B**	C5	Stack the hour & display it by calling the address field of the display using monitor routine.
C145		**CALL UPDAD**	CD BC 06	
C148		**POP B**	C1	
C149		**MOV A,B**	78	Hour is moved to accumulator
C14A		**CPI 25_H**	FE 13	Compare whether the 24 hours are completed.
C14C		**JNZ HOUR**	C2 07 C1	When 24 hours are not over ,then hour counter is incremented by one.
C14F		**MOV B,01_H**	06 01	Reinitialize the hour
C151		**MOV A,B**	78	Hour is moved to accumulator
C152		**STA FFF8_H**	32 F8 FF	Accumulator content is stored as the hour at the address field of the display for each looping
C155		**PUSH B**	C5	Stack the hour and display it by calling the address field of the display using monitor routine.
C156		**CALL UPDAD**	CD BC 06	
C159		**POP B**	C1	
C15A		**JMP RESET**	C3 00 C1	Jump to label RESET.
C200		**MVI B,02_H**	C6 02	The delay time is actually

C202	**DELAY**	**LXI D,A000$_H$**	11 00 A0	the calculation of the total execution time for the program. Delay subprogram is used to generate the frequency which is inversely proportional to time period. In order to make the visibility of the digits in the display.
C205	**X1**	**DCX D**	1B	
C206		**MOV A,E**	7B	
C207		**ORA D**	B2	
C208		**JNZ X1**	C2 05 C2	
C20B		**DCR B**	05	
C20C		**JNZ DELAY**	C2 02 C2	
C20F		**RET**	C9	

EXECUTION



REFERENCE

1. Ramesh S.Gaonkar, Microprocessor Architecture, Programming, and Applications, Fourth Edition, Penram International Publishing (India), 2000.

2. S.Subathra, “Advanced Microprocessor Laboratory”, Record work, Adhiparashakthi Engineering College, Melmaruvathur, October 2002

3. S.Subathra, “Programming in 8085 Microprocessor and its applications – An Innovative Analysis”, Technical Report, Adhiparashakthi Engineering College, Melmaruvathur, March 2003

4. Micro-85 EB, User Manual, Version – 3.0, CAT #M85 EB-002, VI Microsystems Pvt. Ltd., Chennai.