ELEVATOR INTERFACE

BY
SUBATHRA S

ELEVATOR INTERFACE

OBJECTIVE

To write an assembly language program to interface an elevator with 8085 Microprocessor kit

APPARATUS REQUIRED

- Single port Microcomputer kit (ALS kit).
- Elevator interfacing kit.
- Power Supply.
- Flat Ribbon Cable.

OPERATING INSTRUCTION

A program that illustrates the function of elevator interfaces returns. The elevator starts at the ground floor and makes up the loop. The request are sensed each time the lift position is changed and the presence of request is displayed as in the address field once the elevator has moved to the top all the request are cleared. The speed of movement of the elevator can be changed by varying the number of states times the delay routine is called with some changes in the software power is applied to the interface through a polarized connection.

DESCRIPTION

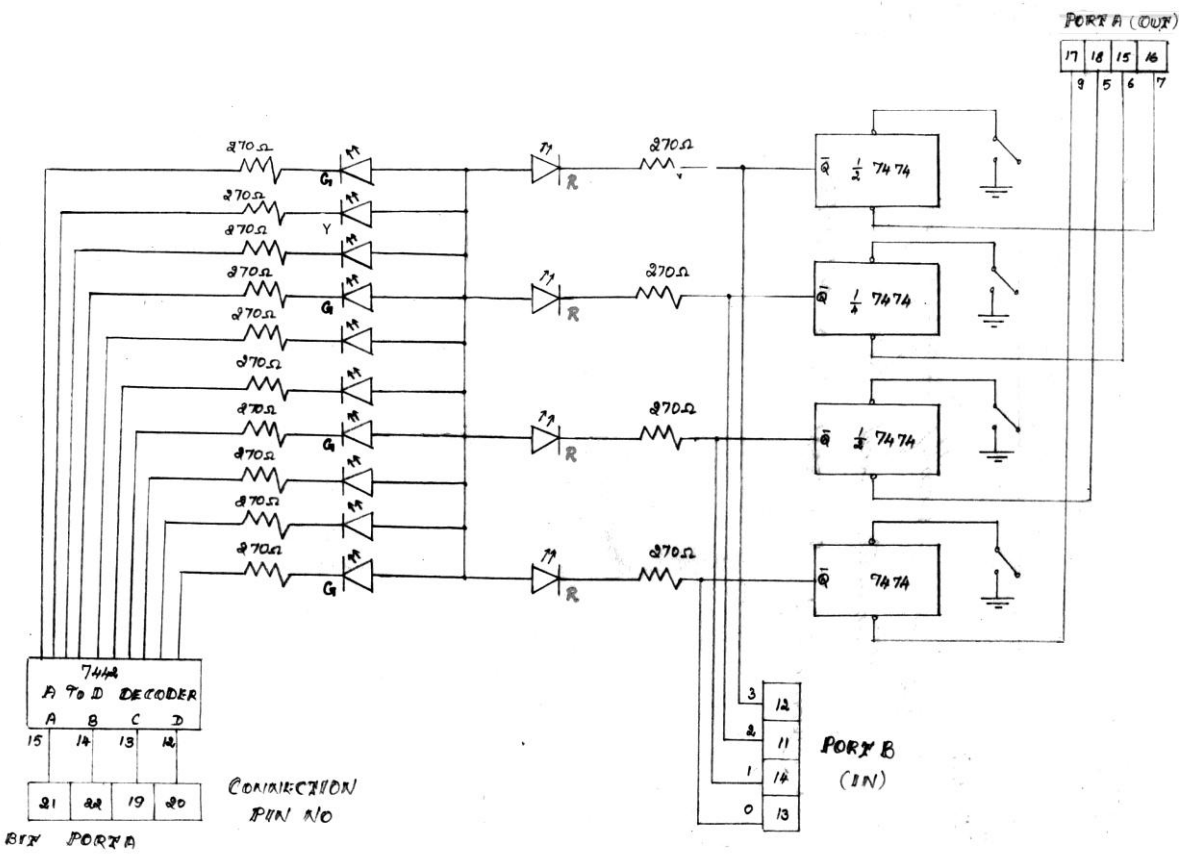
An elevator presents a typical problem to logical designs. A number of situation like the occurrence of simulation requests are frequently encountered. The use of microcomputer enables complex simulation exercise to be performed.

The control and operation of an elevator interfacing functioning within a building having four floors is simulated by this interface. A key and corresponding LED indicated serve as a request button and request indicator (for simplicity the request button within the elevator has not been included).

A row of LED's indicates the position of the elevator within the shaft. The green LED's when indicate that elevator is at a floor. The serving of a request (the appearance of an elevator at a specific floor) can be indicated by clearing the request (turning off the request indicates).

A four line to ten line decoder is used to specify the location of the elevator within the shaft when the inputs of the decoder are requested through binary codes corresponding to (0-9) with a time interval between transistors, the elevator is seen to move. Two dual OFF are used to sense requests and output the states. A request is cleared through the clear input of FIF.

CIRCUIT DIAGRAM



ASSEMBLY LANGUAGE PROGRAM

ADDRESS	LABEL	MNEMONICS	OPCODE/OPERAND
C100		**MVI A,82$_H$**	3E 82
C102		**OUT CWR**	D3 DB
C104		**MVI B,00$_H$**	06 00
C106		**MOVA,B**	78
C107		**ORI F0$_H$**	F6 F0
C108		**OUT PORTA**	D3 D8
C10B	**LOOP1**	**LXI H,C180$_H$**	21 80 C1
C10E	**LOOP2**	**IN PORTB**	DB D9
C110		**ORI F0$_H$**	F6 F0
C112		**MOV C,A**	4F
C113		**SUI FF$_H$**	D6 FF
C115		**JZ LOOP2**	CA 0E C1
C118	**LOOP3**	**MOV A,C**	79
C119		**RAR**	1F
C11A		**MOV C,A**	4F
C11B		**JNC DECODE**	D2 22 C1
C11E		**INX H**	23
C11F		**JMP LOOP3**	C3 18 C1
C122	**DECODE**	**CALL DELAY**	CD 49 C1
C125		**MOV A,M**	7E

C126		**SUB B**	90
C127		**JC DOWN**	DA 36 C1
C12A		**JZ LOOP4**	CA 3F C1
C12D		**INR B**	04
C12E		**MOV A,B**	78
C12F		**ORI $F0_H$**	F6 F0
C131		**OUT PORTA**	D3 D8
C133		**JNZ DECODE**	C2 22 C1
C136	**DOWN**	**DCR B**	05
C137		**MOV A,B**	78
C138		**ORI $F0_H$**	F6 F0
C13A		**OUT PORTA**	D3 D8
C13C		**JMP DECODE**	C3 22 C1
C13F	**LOOP4**	**MVI A,05_H**	3E 05
C141		**ADD L**	85
C142		**MOV L,A**	6F
C143		**MOV A,M**	7E
C144		**OUT PORTA**	D3 D8
C146		**JMP LOOP1**	C3 0B C1
C149	**DELAY**	**MVI D,04_H**	16 04
C14B	**DL1**	**MVI E,FF_H**	1E FF
C14D	**DL2**	**MVI C,FF_H**	0E FF
C14F	**DL3**	**DCR C**	0D
C150		**JNZ DL3**	C2 4F C1
C153		**DCR E**	1D
C154		**JNZ DL2**	C2 4D C1
C157		**DCR D**	15
C158		**JNZ DL1**	C2 4B C1
C15B		**RET**	C9

EXECUTION

ADDRESS	DATA
$C180_H$	00_H
$C181_H$	03_H
$C182_H$	06_H
$C183_H$	09_H
$C184_H$	00_H
$C185_H$	$E0_H$
$C186_H$	$D3_H$
$C187_H$	$B6_H$
$C188_H$	79_H

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