KARAR MEHDI HABIB-1304945

PROJECT: ELECTRONIC DIE

HELSINKI METROPOLIA UNIVERSITY OF APPLIED SCIENCES

DEGREE: BACHELOR

DEGREE PROGRAMME: INFORMATION TECHNOLOGY

COURSE: DIGITAL CIRCUITS

Project Description:

Electronic dice consists of seven LEDs and two buttons. One button is a Test button which turns on all the seven LEDs and when the button is released then all LEDs are switched off. The second button is Operation button which when pressed and released randomly displays a number between 1 and 6. This is accomplished making the dice run in the states 1-6 in circle at very high frequency by pressing the Operation button. Number is displayed until user operates dice again or if test button is pressed.

Requirements:

In this project, I used

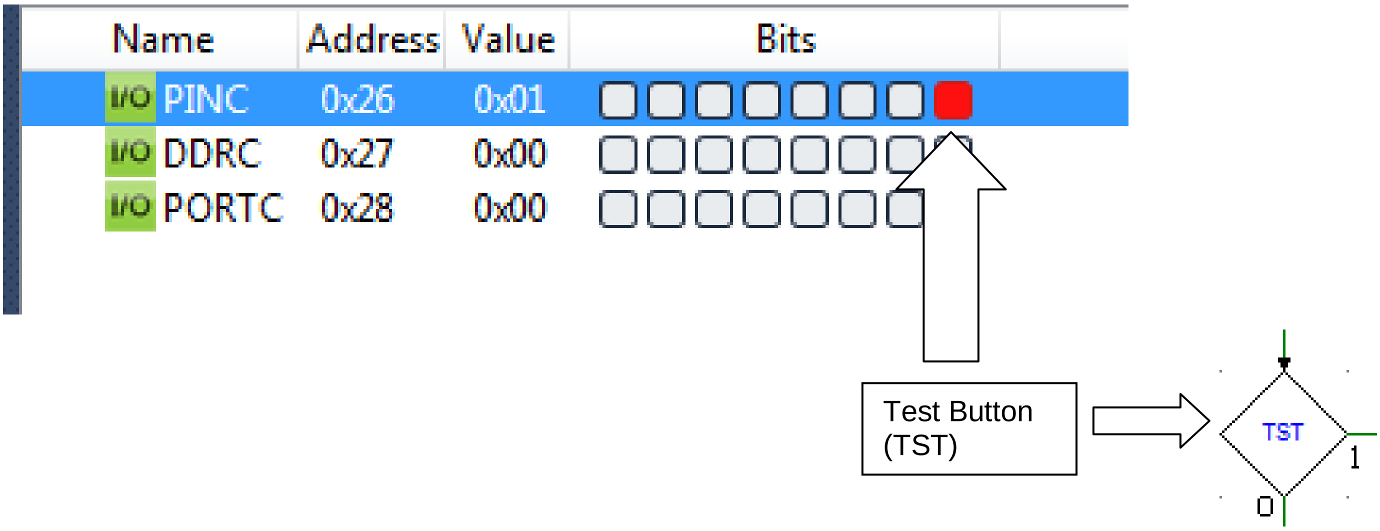
- ATMEGA328P because this board is quite available everywhere and I am quite familiar working with it.
- Assembly Language (AVR Studio-5)

Input Port:

In the project, I used PINC pins of PORTC as inputs. The following binary numbers on PINC pins of PORTC represents the input buttons:

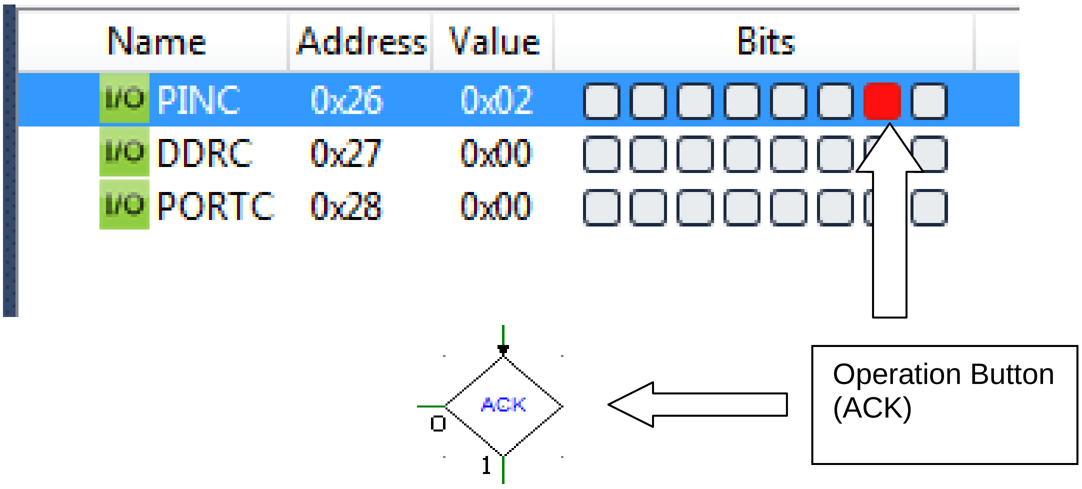
00000001 - represents the Test Button (TST on ASM Chart)

00000010 - represents the Operation Button (ACK on ASM chart)



Name	Address	Value	Bits
I/O PINC	0x26	0x01	□□□□□□□🟥
I/O DDRC	0x27	0x00	□□□□□□□□
I/O PORTC	0x28	0x00	□□□□□□□□

Fig: 1- Test Button (TST on ASM Chart)



Name	Address	Value	Bits
I/O PINC	0x26	0x02	□□□□□□🟥□
I/O DDRC	0x27	0x00	□□□□□□□□
I/O PORTC	0x28	0x00	□□□□□□□□

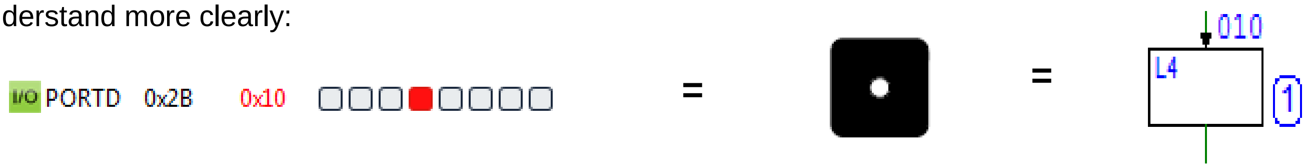
Operation Button (ACK)

Fig: 2- Operation Button (ACK on ASM Chart).

Output Port:

In the project, I used PORTB pins of PORTB as outputs. The following binary numbers on PORTB pins of PORTB represents the LEDs (outputs):

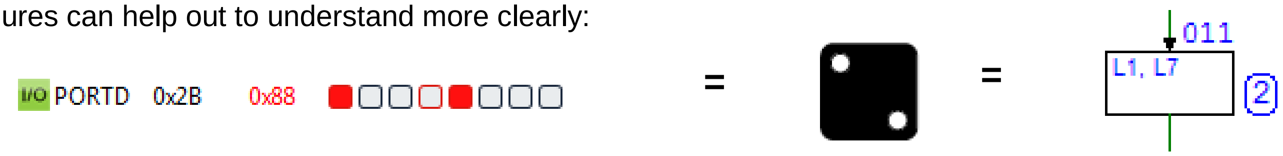
00010000 - represents L4 (on ASM chart) is on. That is only one LED is on which means the die is showing number 1. This represents state named ‘1’ on ASM chart. The following figures can help out to understand more clearly:



I/O PORTD 0x2B 0x10 □□□🟥□□□□

Fig: 3- Only one LED (L4 on ASM) is on. This represents state named ‘1’ on ASM chart.

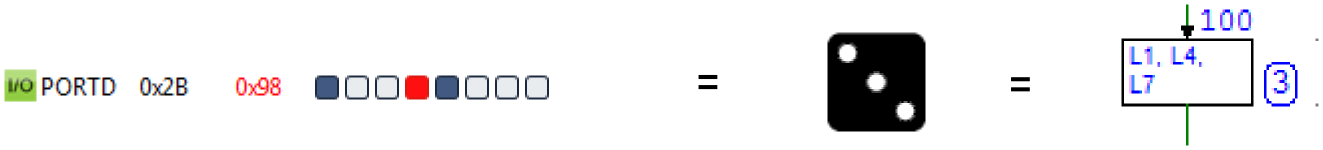
10001000- represents two LEDs (L1 and L7 on ASM chart) are on. That is total two LEDs are on which means the die is showing number 2. This represents state named ‘2’ on ASM chart**.** The following figures can help out to understand more clearly:



I/O PORTD 0x2B 0x88 🟥□□□🟥□□□

Fig: 4- Two LEDs (L1 and L7 on ASM) are on. This represents state named ‘2’ on ASM chart.

10011000 - represents three LEDs (L1, L4 and L7 on ASM chart) are on. That is total three LEDs are on which means the die is showing number 3. This represents state named ‘3’ on ASM chart**.** The following figures can help out to understand more clearly:



I/O PORTD 0x2B 0x98 ■□□🟥■□□□

Fig: 5- Three LEDs (L1, L4 and L7 on ASM) are on. This represents state named ‘3’ on ASM chart.

10101010 - represents four LEDs (L1, L3, L5 and L7 on ASM chart) are on. That is total four LEDs are on which means the die is showing number 4. This represents state named ‘4’ on ASM chart**.** The following figures can help out to understand more clearly:

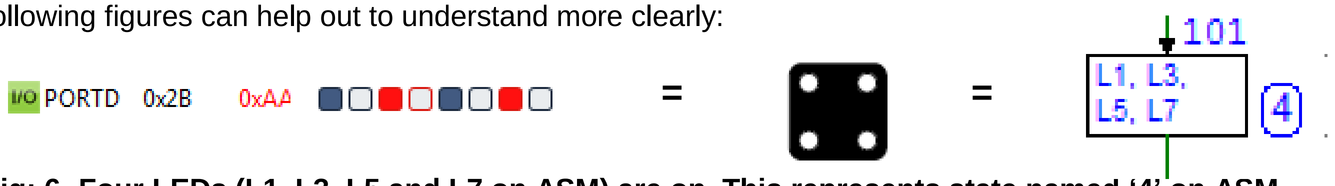


Fig: 6- Four LEDs (L1, L3, L5 and L7 on ASM) are on. This represents state named ‘4’ on ASM chart.

10111010 - represents five LEDs (L1, L3, L4, L5 and L7 on ASM chart) are on. That is total five LEDs are on which means the die is showing number 5. This represents state named ‘5’ on ASM chart**.** The following figures can help out to understand more clearly:

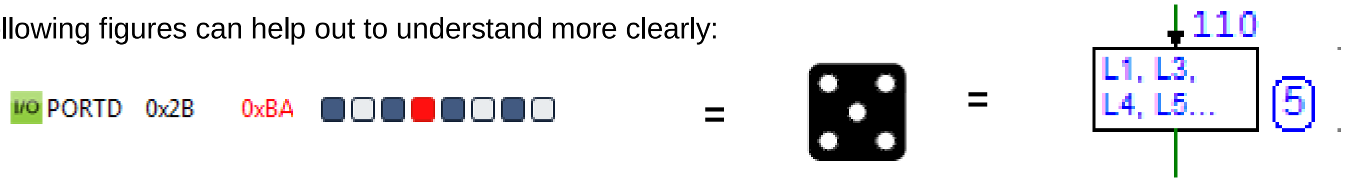


Fig: 7- Five LEDs (L1, L3, L4, L5 and L7 on ASM) are on. This represents state named ‘5’ on ASM chart.

11101110 - represents six LEDs (L1, L2, L3, L5, L6 and L7 on ASM chart) are on. That is total six LEDs are on which means the die is showing number 6. This represents state named ‘6’ on ASM chart. The following figures can help out to understand more clearly:

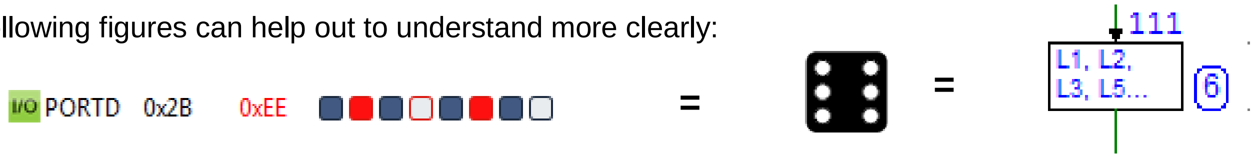


Fig: 8- Six LEDs (L1, L2, L3, L5, L6 and L7 on ASM) are on. This represents state named 6 on ASM chart.

11111110 - represents seven LEDs (L1, L2, L3, L4, L5, L6 and L7 on ASM chart) are on. That is seven LEDs are on which means that the Test Button (TST on ASM) is pressed before pressing Operation Button (ACK on ASM) for the first time. This represents state named ‘7’ on ASM chart**.** The following figures can help out to understand more clearly:

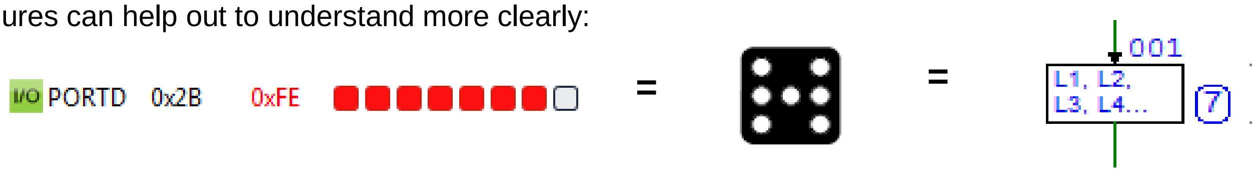
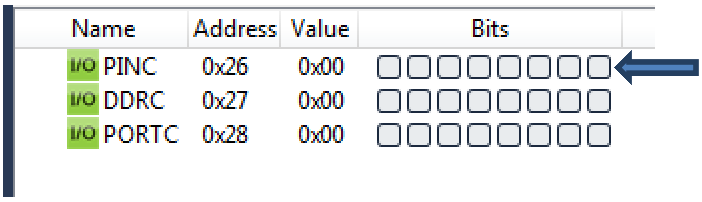


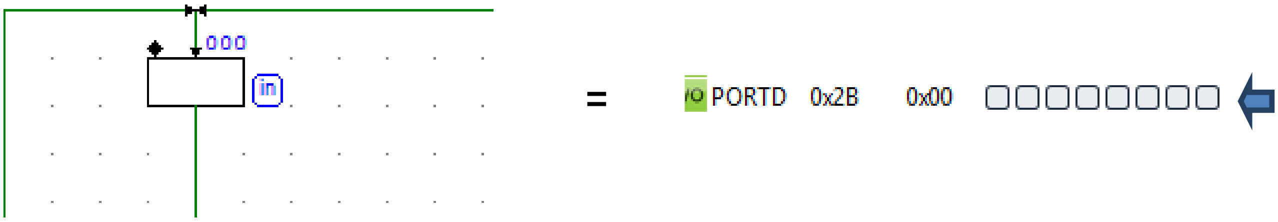
Fig: 9- Seven LEDs (L1, L2, L3, L4, L5, L6 and L7 on ASM) are on. This represents state named ‘7’ on ASM chart.

How the ASM Chart is similar to the program?



Name	Address	Value	Bits
I/O PINC	0x26	0x00	□□□□□□□□
I/O DDRC	0x27	0x00	□□□□□□□□
I/O PORTC	0x28	0x00	□□□□□□□□

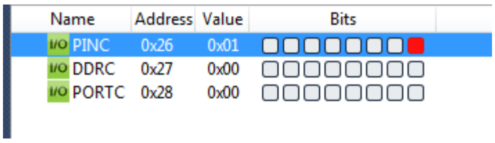
Fig: 10- No input given or Button pressed.



I/O PORTD	0x2B	0x00	□□□□□□□□

Fig: 11- Initial state (state named ‘in’ on ASM Chart)

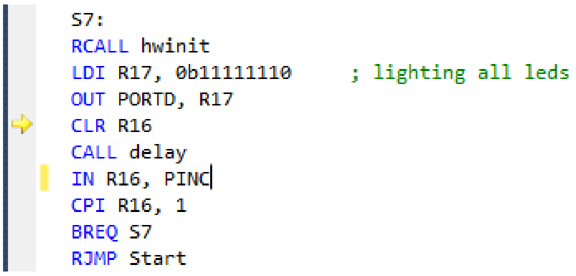
The Fig: 11 above shows that everything is in the initial state because no input is given or input button is pressed. The left figure shows the initial state on ASM chart and the right figure shows that there is no output since no input is given as you can see from Fig: 10.



Name	Address	Value	Bits
I/O PINC	0x26	0x01	□□□□□□□🟥
I/O DDRC	0x27	0x00	□□□□□□□□
I/O PORTC	0x28	0x00	□□□□□□□□

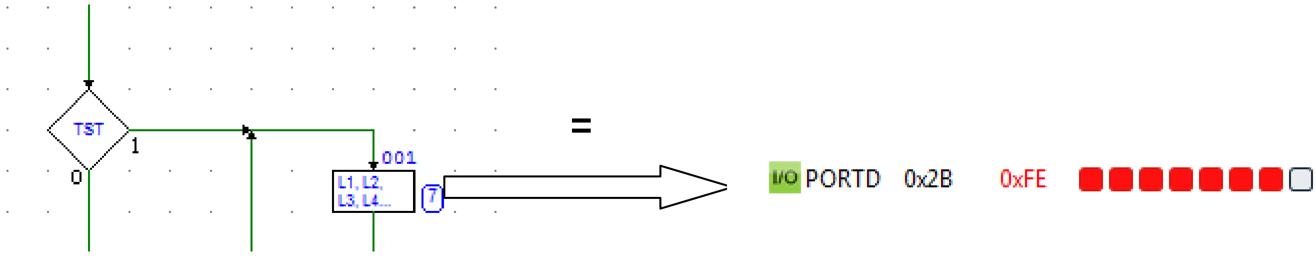
Fig: 12- Test Button (TST on ASM Chart) is pressed.

Fig: 13



```
S7:
RCALL hwinit
LDI R17, 0b11111110     ; lighting all leds
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
CPI R16, 1
BREQ S7
RJMP Start
```

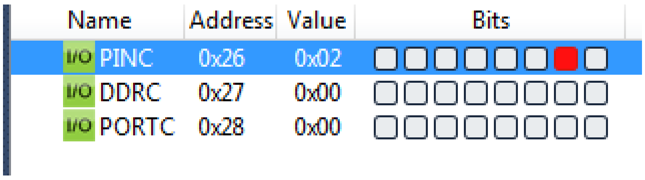
Fig: 13- The arrow jumps to S7 (loop for lighting all 7 LEDs) when Test Button (TST on ASM Chart) is pressed.



I/O PORTD	0x2B	0xFE	🟥🟥🟥🟥🟥🟥🟥□

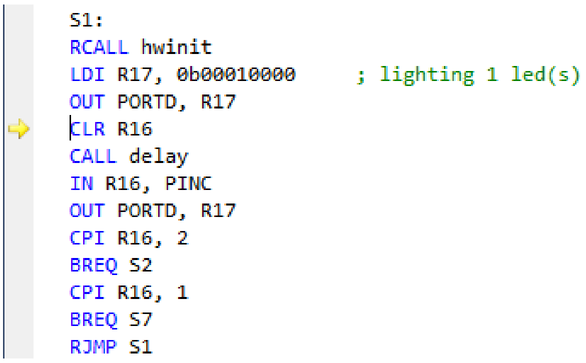
Fig: 14- State named ‘7’ (loop for lighting all 7 LEDs) on ASM Chart and the output after pressing Test Button.

In Fig: 12, it can be seen that one input is given that is Test Button is pressed which causes the arrow in Fig: 13 to jump to S7 (loop for lighting all 7 LEDs) which results in lighting up all 7 LEDs that can be concluded from the right picture of Fig: 14. The left picture on Fig: 14 shows the state named ‘7’ (state where all 7 LEDs light up) after pressing Test Button (TST) and the right picture shows that all 7 LEDs lighting up due to activating the Test Button on the program.



Name	Address	Value	Bits
I/O PINC	0x26	0x02	□□□□□□🟥□
I/O DDRC	0x27	0x00	□□□□□□□□
I/O PORTC	0x28	0x00	□□□□□□□□

Fig: 15- Operation Button (ACK on ASM Chart) is pressed.



```
S1:
RCALL hwinit
LDI R17, 0b00010000     ; lighting 1 led(s)
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
OUT PORTD, R17
CPI R16, 2
BREQ S2
CPI R16, 1
BREQ S7
RJMP S1
```

Fig: 16-The arrow jumps to S1 (loop for lighting 1 LED) when Operation Button (ACK on ASM Chart) is pressed.

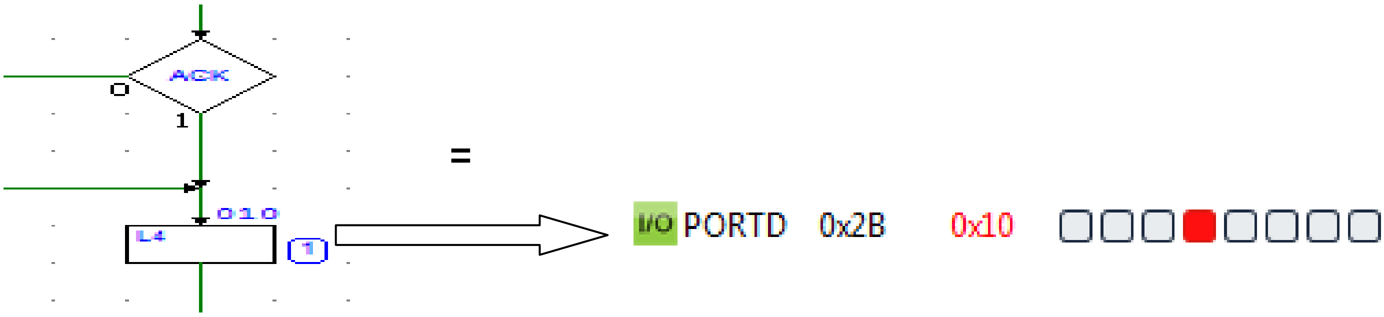
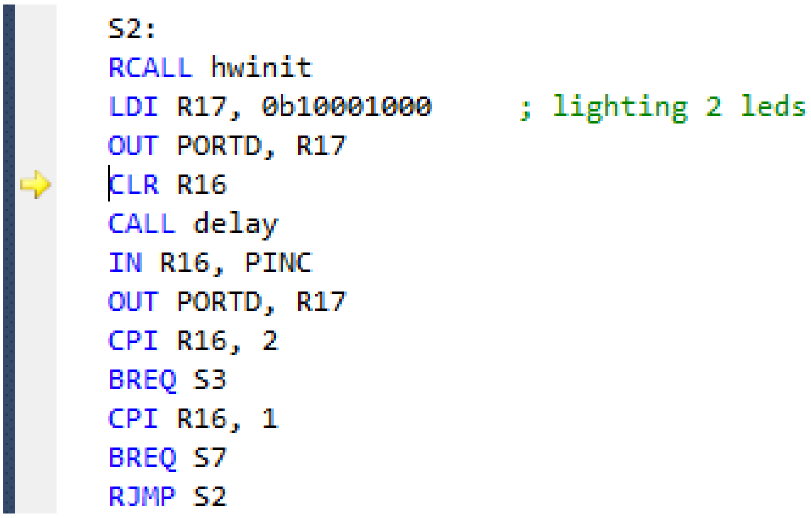


Fig: 17- Shows state named ‘1’ (state for lighting up one LED) and output after pressing Operation Button (ACK on ASM Chart).

In Fig: 15, it can be seen that Operation Button is pressed which causes the arrow in Fig: 16 to jump to S1 (loop for lighting only one LED) which results in lighting up one LED that can be concluded from the right picture of Fig: 17. The left picture on Fig: 17 shows the state named ‘1’ (state where only one LED lights up) after pressing Operation Button (ACK) and the right picture shows that one LED is lighting up due to activating the Operation Button on the program. Moreover, if the Operation Button (ACK) is kept pressing then the program lights up total 2 LEDs followed by 3, 4, 5, 6 and then again starts lighting up 1 LED followed by 2, 3 , 4, 5, 6. The cycle goes on until the Operation Button is kept pressed. Finally, when the Operation Button is released then the program lights up the number of LEDs that it was lighting up just before the Operation Button was released thus enabling the electronic die to view the number. For example- if it lights up one LED then the number is 1, if it light up total two LEDs then the number is ‘2’ and so on. The fact can be more understandable by seeing the following figures:

On Fig:18 (Next Page)



```
S2:
RCALL hwinit
LDI R17, 0b10001000      ; lighting 2 leds
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
OUT PORTD, R17
CPI R16, 2
BREQ S3
CPI R16, 1
BREQ S7
RJMP S2
```

Fig: 18- The arrow jumps from S1 (loop for lighting up total one LED) to S2 (loop for lighting up total two LEDs) when Operation Button is kept pressing.

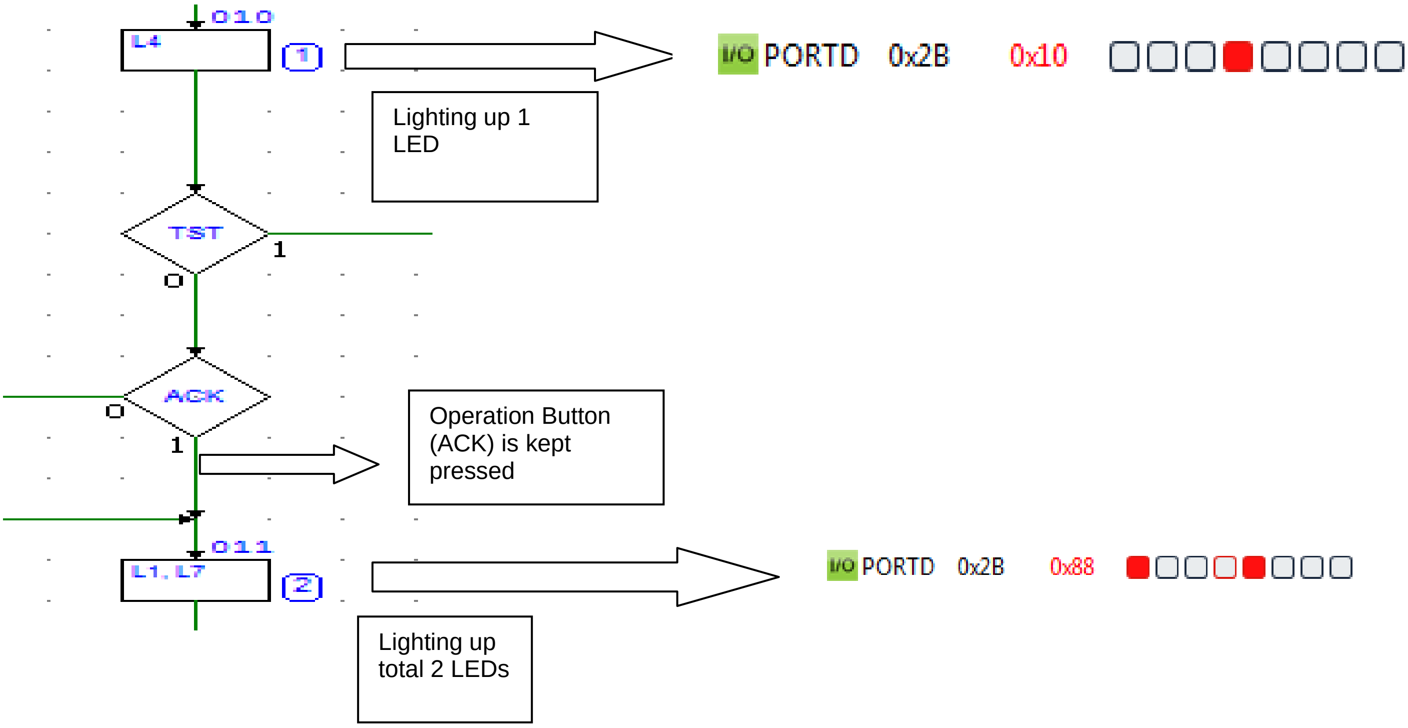
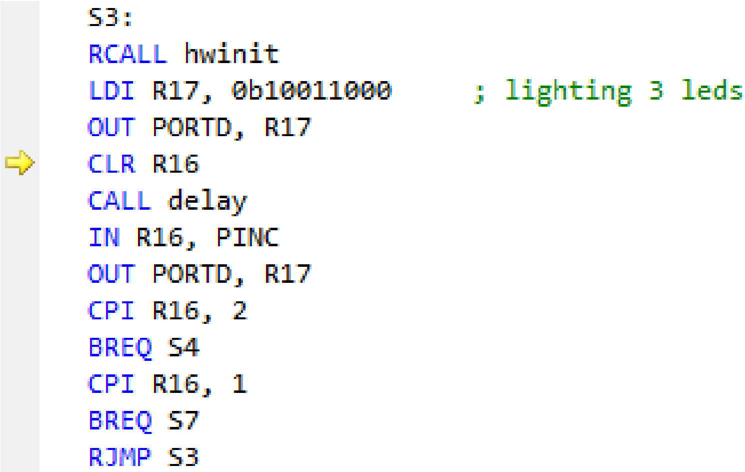


Fig: 19- State named ‘1’ (state for lighting up 1 LED) and when Operation Button (ACK) is kept pressing then the program jumps to the state named ‘2’ (state for lighting up 2 LEDs).

On Fig:20 (Next Page)



```
S3:
RCALL hwinit
LDI R17, 0b10011000     ; lighting 3 leds
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
OUT PORTD, R17
CPI R16, 2
BREQ S4
CPI R16, 1
BREQ S7
RJMP S3
```

Fig: 20- The arrow jumps from S2 (loop for lighting up total 2 LEDs) to S3 (loop for lighting up total three LEDs) when Operation Button is still kept pressed.

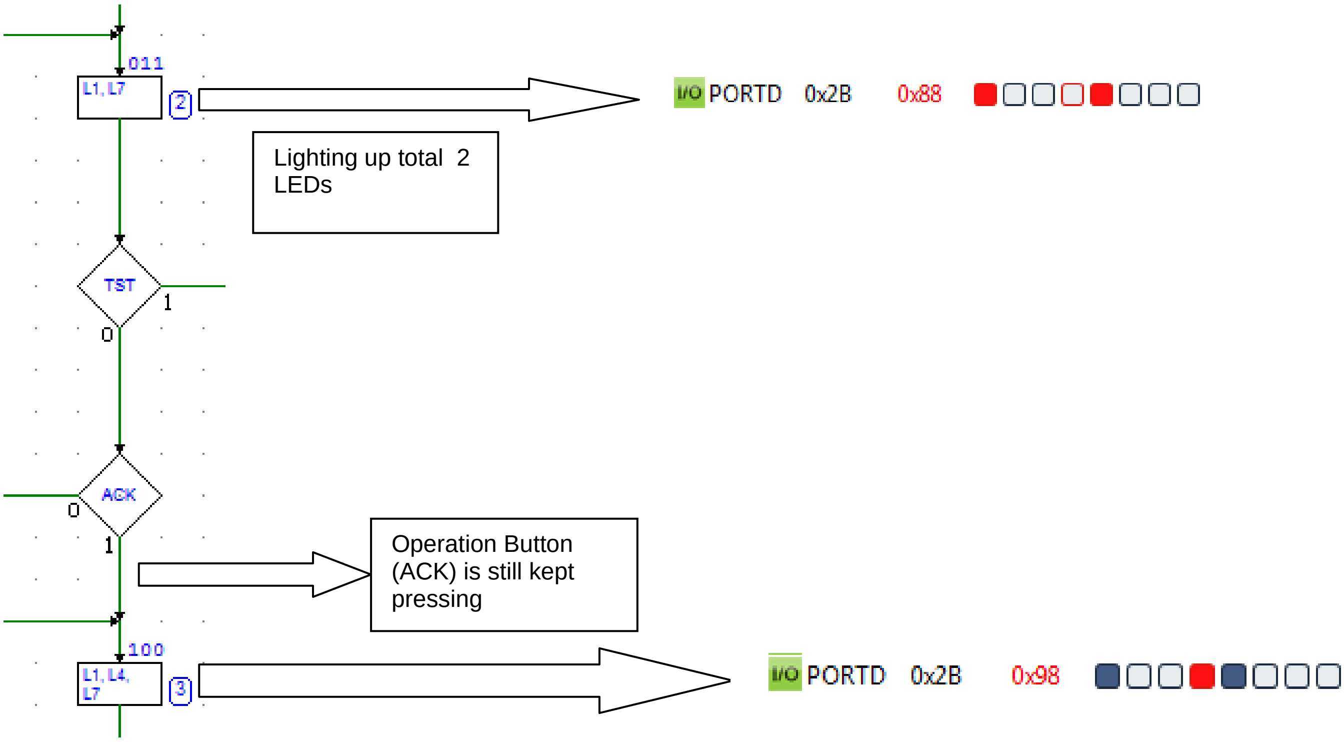
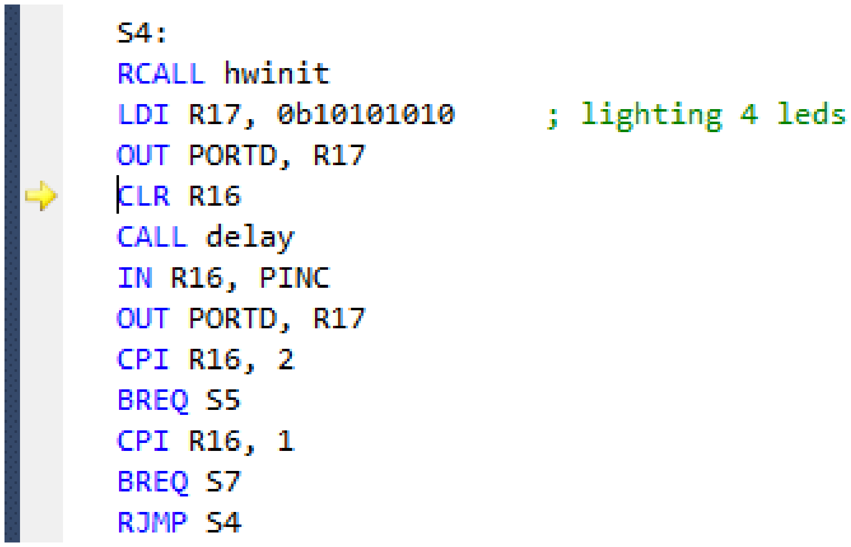


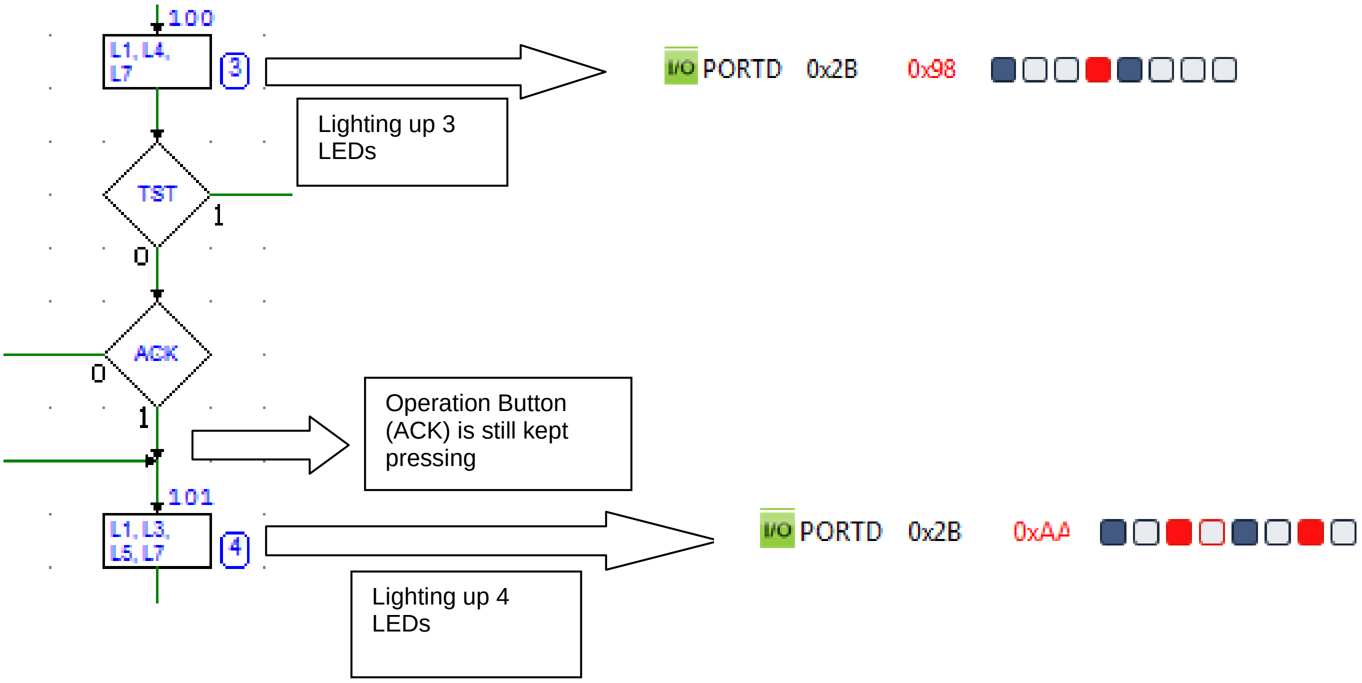
Fig: 21- State named ‘2’ (state for lighting up total 2 LEDs) and when Operation Button (ACK) is still kept pressing then the program jumps to the state named ‘3’ (state for lighting up total 3 LEDs).

On Fig:22 (Next Page)



```
S4:
RCALL hwinit
LDI R17, 0b10101010      ; lighting 4 leds
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
OUT PORTD, R17
CPI R16, 2
BREQ S5
CPI R16, 1
BREQ S7
RJMP S4
```

Fig: 22- The arrow jumps from S3 (loop for lighting up total 3 LEDs) to S4 (loop for lighting up total 4 LEDs) when Operation Button is still kept pressed.



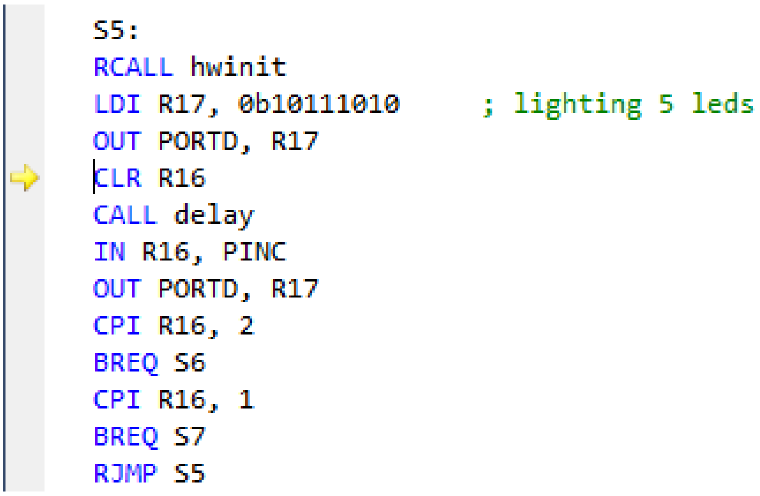
Lighting up 3 LEDs

Operation Button (ACK) is still kept pressing

Lighting up 4 LEDs

Fig: 23- State named ‘3’ (state for lighting up 3 LEDs) and when Operation Button (ACK) is still kept pressing then the program jumps to the state named ‘4’ (state for lighting up 4 LEDs).

On Fig:24 (Next Page)



```
S5:
RCALL hwinit
LDI R17, 0b10111010     ; lighting 5 leds
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
OUT PORTD, R17
CPI R16, 2
BREQ S6
CPI R16, 1
BREQ S7
RJMP S5
```

Fig: 24- The arrow jumps from S4 (loop for lighting up total 4 LEDs) to S5 (loop for lighting up total 5 LEDs) when Operation Button is still kept pressing.

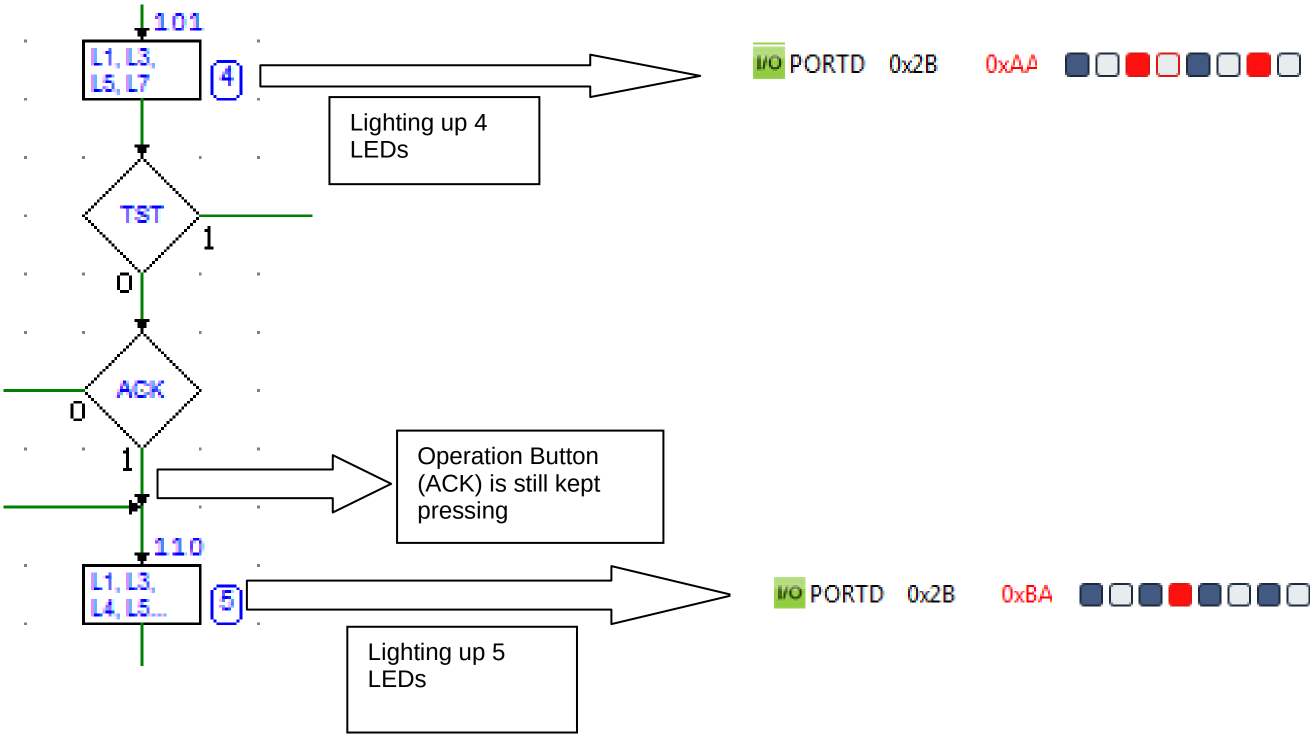
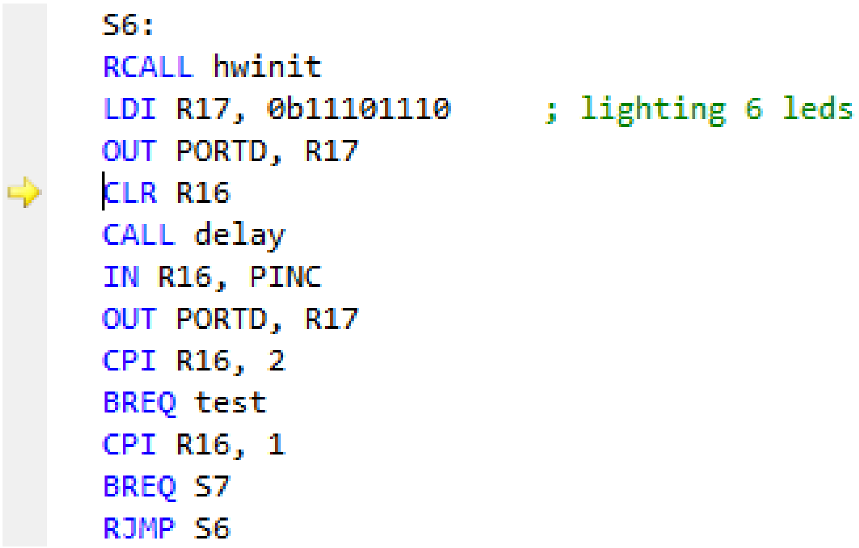


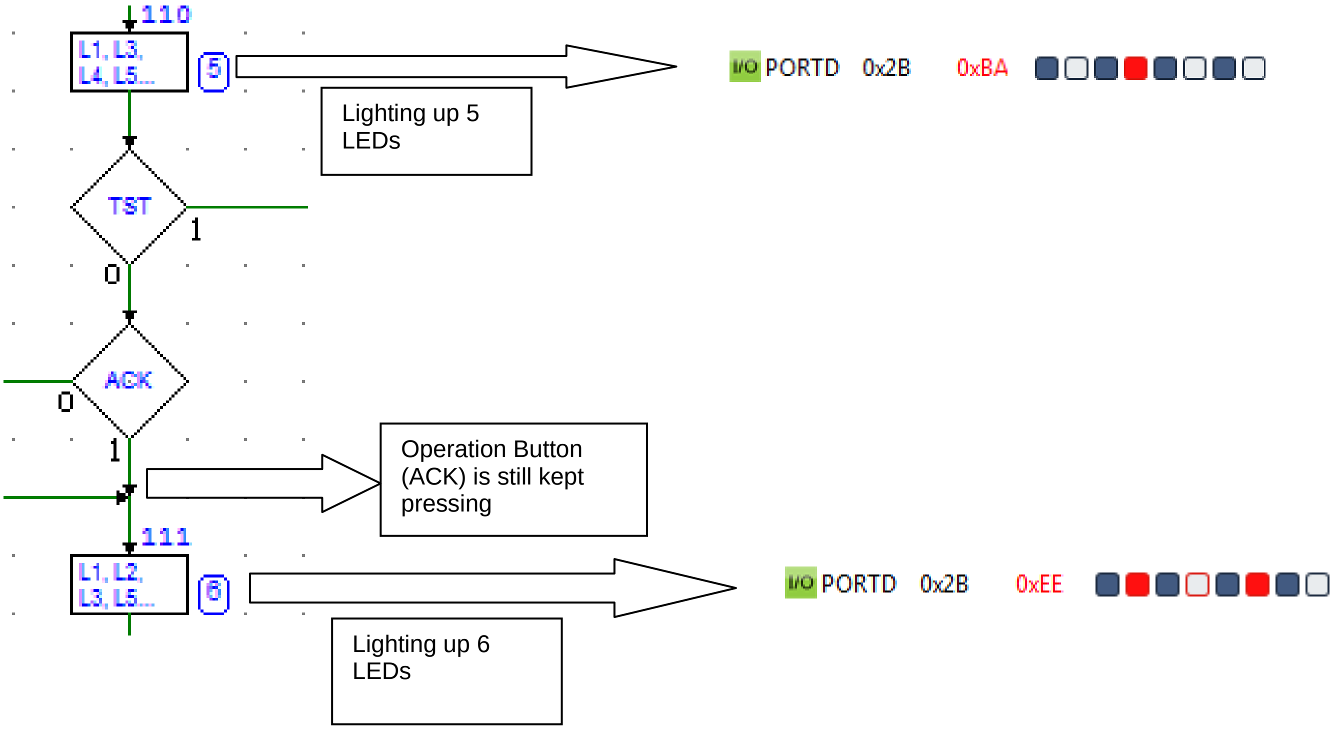
Fig: 25- State named ‘4’ (state for lighting up total 4 LEDs) and when Operation Button (ACK) is still kept pressing then the program jumps to the state named ‘5’ (state for lighting up 5 LEDs).

On Fig:26 (Next Page)



```
S6:
RCALL hwinit
LDI R17, 0b11101110     ; lighting 6 leds
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
OUT PORTD, R17
CPI R16, 2
BREQ test
CPI R16, 1
BREQ S7
RJMP S6
```

Fig: 26- The arrow jumps from S5 (loop for lighting up total 5 LEDs) to S6 (loop for lighting up total 6 LEDs) when Operation Button is still kept pressed.



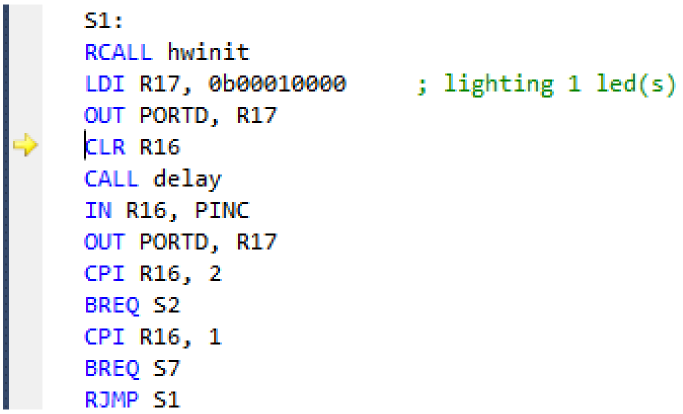
Lighting up 5 LEDs

Operation Button (ACK) is still kept pressing

Lighting up 6 LEDs

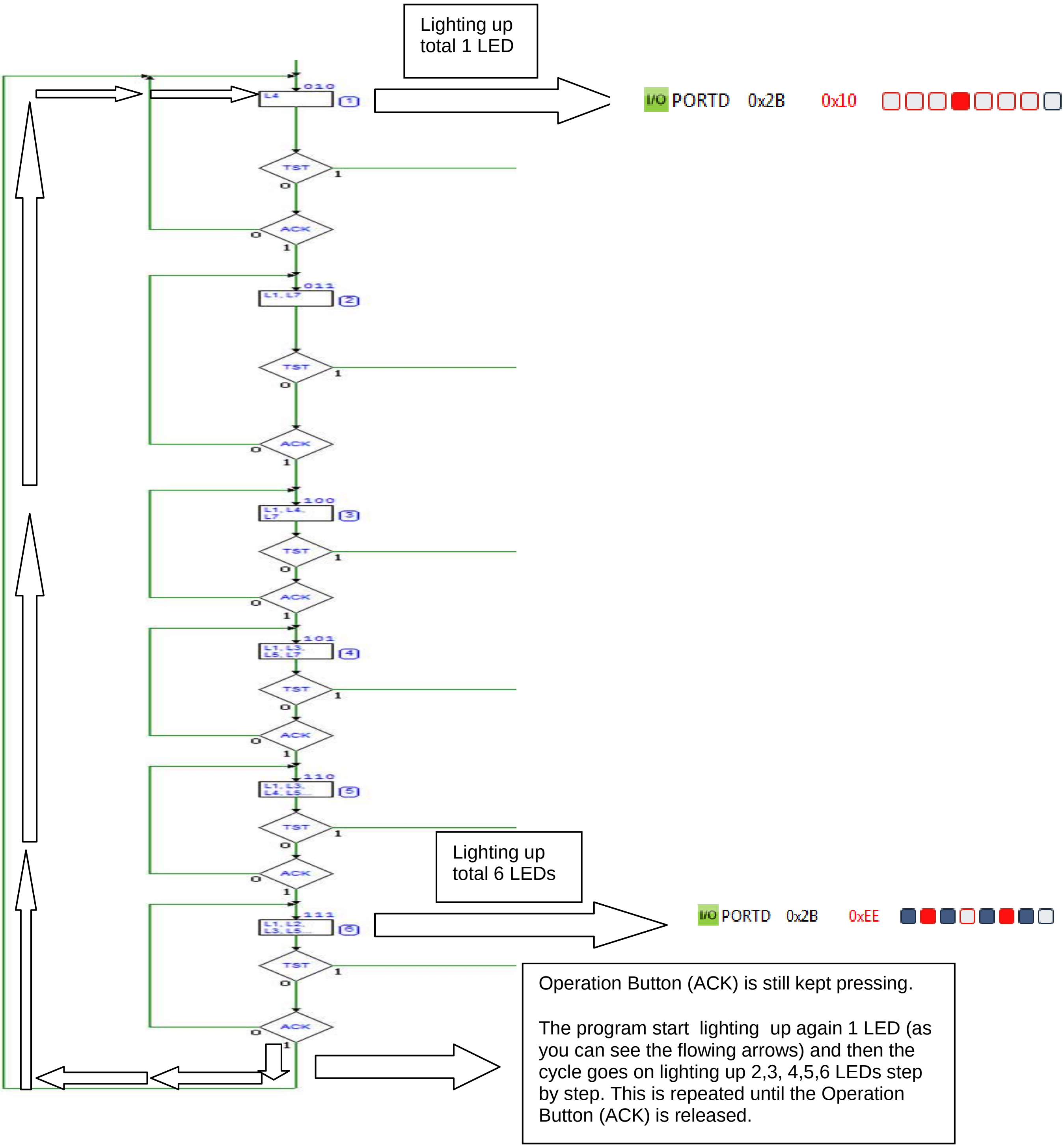
Fig: 27- State named ‘5’ (state for lighting up total 5 LEDs) and when Operation Button (ACK) is still kept pressing then the program jumps to the state named ‘6’ (state for lighting up total 6 LEDs).

On Fig:28 (Next Page)



```
S1:
RCALL hwinit
LDI R17, 0b00010000     ; lighting 1 led(s)
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
OUT PORTD, R17
CPI R16, 2
BREQ S2
CPI R16, 1
BREQ S7
RJMP S1
```

Fig: 28- The arrow jumps from S6 (loop for lighting up total 6 LEDs) to S1 (loop for lighting up total 1 LED) when Operation Button is still kept pressing.



Lighting up total 1 LED

Lighting up total 6 LEDs

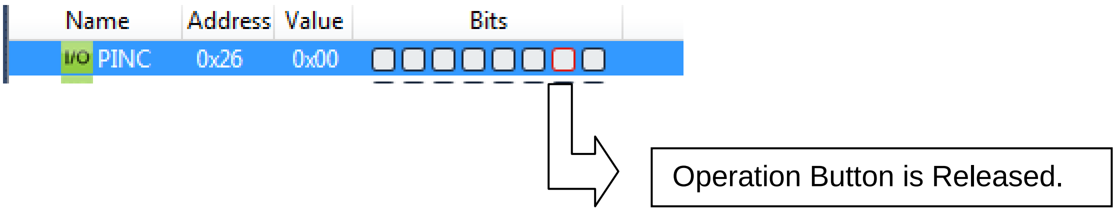
Operation Button (ACK) is still kept pressing.

The program start lighting up again 1 LED (as you can see the flowing arrows) and then the cycle goes on lighting up 2,3, 4,5,6 LEDs step by step. This is repeated until the Operation Button (ACK) is released.

Fig: 29- State named '6' (state for lighting up 6 LEDs) and when Operation Button (ACK) is still kept pressing then the program jumps again to the state named '1' (state for lighting up total 1 LED) and then the cycle goes on lighting up 2, 3,4,5,6 LEDs step by step. The cycle is repeated until the Operation Button (ACK) is kept pressed.

The Fig: 29 explains that when the program light up total 6 LEDs then the program again starts lighting up number of LEDs starting from 1 followed by 2, 3, 4, 5 and again 6 step by step and this cycle is repeated till the Operation Button (ACK) is released.

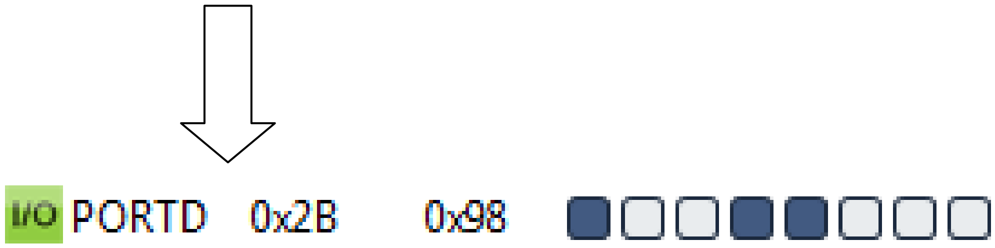
What happens when Operation Button is released at an instance?



Name	Address	Value	Bits
I/O PINC	0x26	0x00	□□□□□□□□

Operation Button is Released.

Fig: 30- Operation Button is Released.



I/O PORTD	0x2B	0x98	■□□■■□□□

Fig: 31- Shows total three LEDs lighting up as the Operation Button is Released that is the die is showing number ‘3’ randomly.

Fig: 30 show that the Operation Button is released at an instance. When the Operation Button is released, the die shows the number by lighting up the number of LEDs that were switched on just before the Operation Button was released. For example- in Fig: 31, total three LEDs are kept on which means the number that the die is viewing is ‘3’ as the Operation Button is released and the three LEDs are switched on until the Test Button or Operation Button is pressed again.

Code for the program:

```
/*
 * AVRAssembler1.asm
 *
 *  Created: 29.4.2014 14:39:27
 *   Author: kararha
 */
  ; in R16 first 2 bits represent the state of the buttons
; 00000001 - value of 1d is going to represent the ACK button
; 00000010 - value of 2d is going to represent the TST button
; register R17 - outputs - first 7 leds connected to them
.include "m328Pdef.inc"
.org 0x0000

ldi R16, high(ramend)
out SPH, R16
ldi R16, low(ramend)
out SPL, R16
ldi R17, high(ramend)
out SPH, R17
ldi R17, low(ramend)
out SPL, R17
```

```
jmp Start
hwinit:
ldi R16, 0
sts UCSR0B, R16 ; disable UASRT0
ldi R17, 0
sts UCSR0B, R17

LDI    R16,  0x00
OUT    DDRC, R16          ; Pins on DDRC as inputs
LDI             R16, 0x03
OUT             PORTC, R16
LDI    R17,  0xFF
OUT    DDRD, R17                                    ; Pins on PORTD as outputs

ret



LDI R20, 0xFF
RJMP Start

delay:
LDI R18, 0
loop:
LDI R19, 0
INC R18
TST R18
BREQ out1
loop2:
INC R19
TST R19
BREQ loop                                                         ; setting a 256*256 delay loop
RJMP loop2
out1:
RET

Start:
RCALL hwinit
CLR R16
CALL delay
LDI R17, 0
OUT PORTD, R17
CALL delay
IN R16, PINC
CPI R16, 2
BREQ S1
CPI R16, 1
BREQ S7
RJMP Start
S1:
RCALL hwinit
LDI R17, 0b00010000                     ; lighting 1 led(s)
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
OUT PORTD, R17
CPI R16, 2
BREQ S2
CPI R16, 1
BREQ S7
RJMP S1
S2:
RCALL hwinit
LDI R17, 0b10001000                     ; lighting 2 leds
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
OUT PORTD, R17
CPI R16, 2
BREQ S3
CPI R16, 1
```

```
BREQ S7
RJMP S2
S3:
RCALL hwinit
LDI R17, 0b10011000                     ; lighting 3 leds
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
OUT PORTD, R17
CPI R16, 2
BREQ S4
CPI R16, 1
BREQ S7
RJMP S3
S7:
RCALL hwinit
LDI R17, 0b11111110                     ; lighting all leds
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
CPI R16, 1
BREQ S7
RJMP Start
S4:
RCALL hwinit
LDI R17, 0b10101010                     ; lighting 4 leds
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
OUT PORTD, R17
CPI R16, 2
BREQ S5
CPI R16, 1
BREQ S7
RJMP S4
S5:
RCALL hwinit
LDI R17, 0b10111010                     ; lighting 5 leds
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
OUT PORTD, R17
CPI R16, 2
BREQ S6
CPI R16, 1
BREQ S7
RJMP S5
S6:
RCALL hwinit
LDI R17, 0b11101110                     ; lighting 6 leds
OUT PORTD, R17
CLR R16
CALL delay
IN R16, PINC
OUT PORTD, R17
CPI R16, 2
BREQ test
CPI R16, 1
BREQ S7
RJMP S6
test:
JMP S1

stop:
rjmp stop
```

ASM Chart:

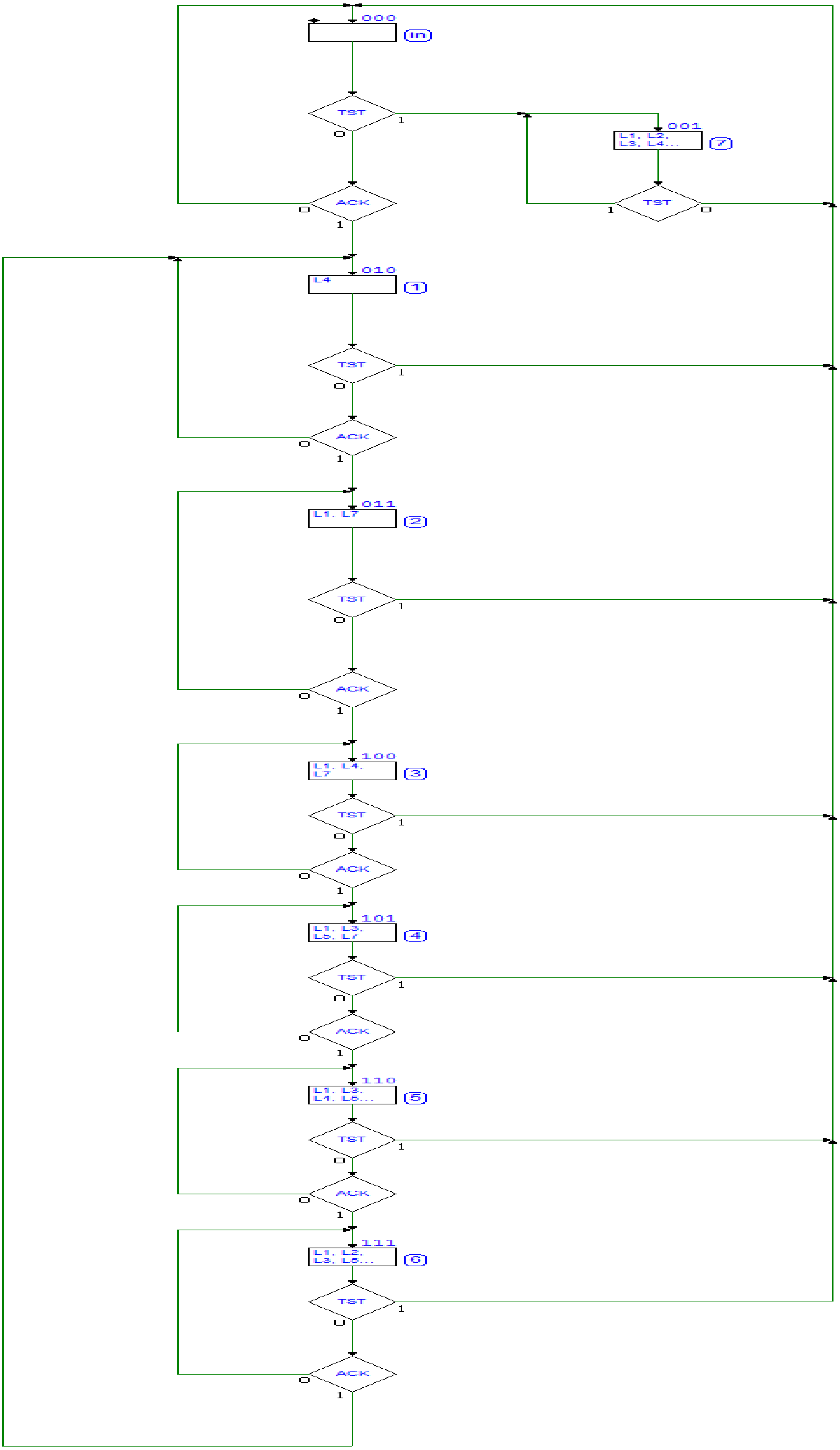
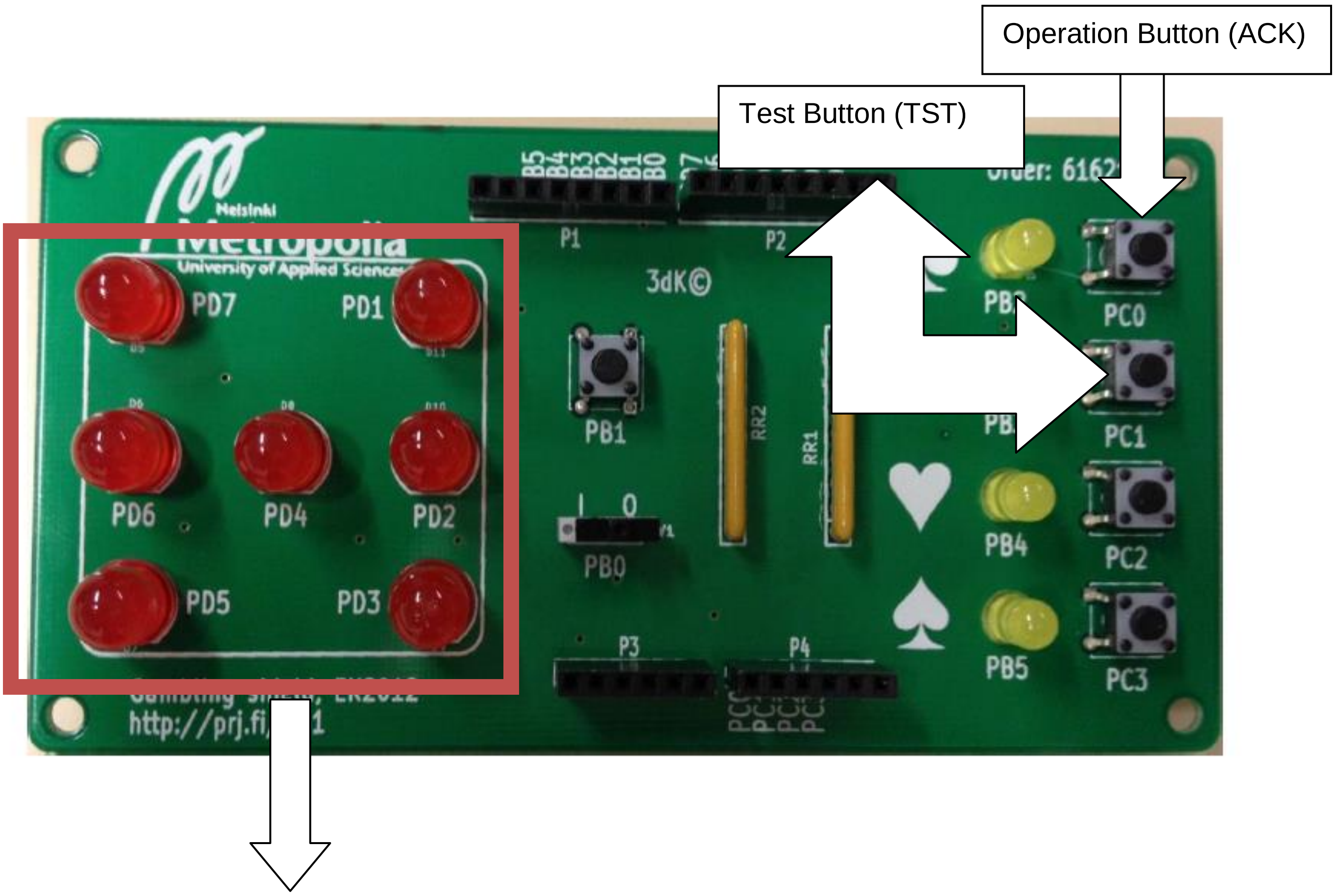


Fig: 32- ASM CHART

Board Used:

- Gambling shield



Operation Button (ACK)

Test Button (TST)

L1		L5
L2	L4	L6
L3		L7

Fig: 33- Gambling Shield Board with the assigned inputs and outputs according to ASM Chart.

In Fig: 33 you can see the input buttons in the board and the output lights similar to ASM Chart.

Conclusion:

Overall, I have developed a program for an electronic die using simple assembly language that displays a random number by lighting up different number of LEDs randomly. It was a great experience to work using assembly language and thanks to Sir **Keijo Länsikunnas** for his materials and guidelines. I hope this report will help to understand the program clearly.