1DT301 lab1

Songho Lee Sarpreet Singh

September 8, 2016

1 Introduction

This report provides the solutions for the first laboratory of the course 1DT301.

2 Task 1

Hereby, code for the first task is presented, which to light LED 2

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;     1DT301, Computer Technology I
;     Date: 2016-09-05
;     Author:
;     Student name Songho Lee
;     Student name Sarpreet Singh
;
;
;     Lab number: 1,
;     Title: Task 1
;
;     Hardware: STK600, CPU ATmega2560
;
;     Function: It enlights the LED lamp number 2, that is connected to the portB
;
;     Input ports: None. We programme directly on the asm
;
;     Output ports: On-board LEDs are connected to PORTB.
;
;     Subroutines: If applicable.
;     Included files: m2560def.inc
;
;     Other information:
;     Changes in program: (Description and date)
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

;     Load pre-configured files for the ports, and memory adresses.
.include "m2560def.inc"

ldi r16, 0xFF  ;Set data direction registers.
out DDRB, r16  ;Set B port as output ports

;LED have 0 as on, 1 as off which is opposite to normal lamps
;Hereby we command second position of the lED to be on.
ldi r16, 0b11111011

out portB, r16
```

In order to light LED light, the minimum lines of code are four. First line should store 0b11111111 into register, second should load that value into data direction register of the desired port. The fourth line should load into the register the desired output value, and the last line should put the desired value into the port so that LED lights on. In such case, the flowchart would look as following:

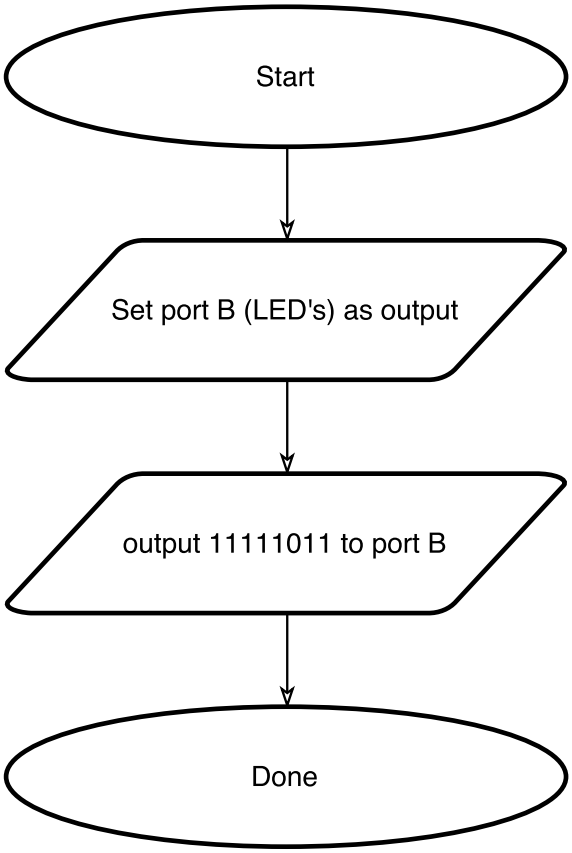

Figure 1: Flowchart of task 1

3 Task 2

Code and a flowchart for the second task is presented.

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;    1DT301, Computer Technology I
;    Date: 2016-09-05
;    Author:
;    Student name Songho Lee
;    Student name Sarpreet Singh
;
;
;    Lab number: 1
;    Title: Task 2
;
;    Hardware: STK600, CPU ATmega2560
;
;    Function: It enlights the LED lamp number 2, that is connected to the portB
;
;    Input ports: On-board Switches are connected to PORTD
;
;    Output ports: On-board LEDs are connected to PORTB.
;
;    Subroutines: If applicable.
;    Included files: m2560def.inc
;
;    Other information:
;    Changes in program: (Description and date)
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

;Load pre-configured files for the ports, and memory adresses.
.include "m2560def.inc"

ldi r16, 0xFF  ;Set data direction registers.
out DDRB, r16  ;Set B port as output ports

ldi r16, 0x00  ;Set data direction registers.
out DDRD, r16  ;Set D port as input ports

;LED have 0 as on, 1 as off which is opposite to normal lamps
;Hereby we command second position of the lED to be on.

ldi r16, 0xFF
out portB, r16

;initialisation is done. All LEDs are off now.

;We create a loop which constantly listens on the switch ports

listening_to_switch_loop:

     in r15, PIND   ;Read switch D and saves into register number 15
     out PORTB, r15 ;

rjmp listening_to_switch_loop
```

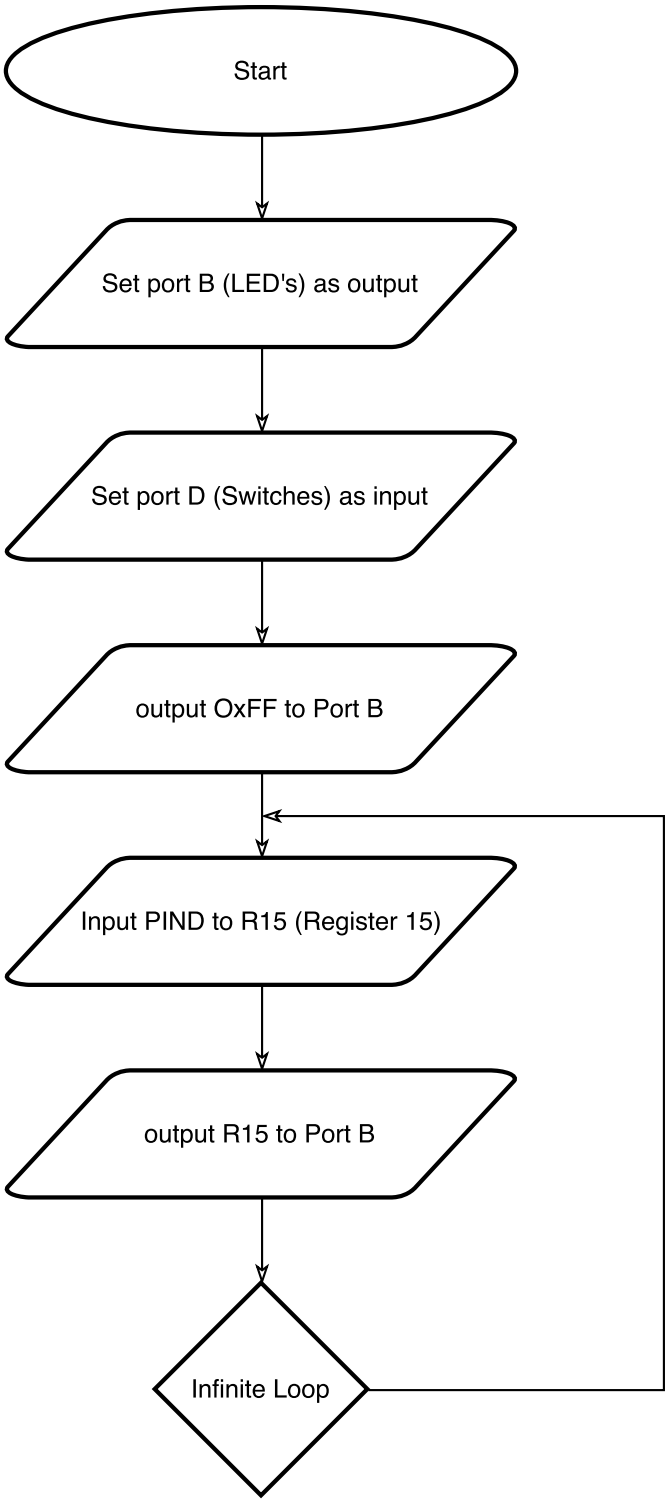

Figure 2: Flowchart of task 2

4 Task 3

Code and a flowchart for the third task is presented.

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;     1DT301, Computer Technology I
;     Date: 2016—09—05
;     Author:
;     Songho Lee
;     Sarpreet Singh
;
;
;     Lab number: 1
;     Title: Task 3
;
;     Hardware: STK600, CPU ATmega2560
;
;     Function: It lights the LED lamp number 0, if switch 5 is pressed
;
;     Input ports: On—board Switches are connected to PORTD
;
;     Output ports: On—board LEDs are connected to PORTB.
;
;     Subroutines: If applicable.
;     Included files: m2560def.inc
;
;     Other information:
;     Changes in program: (Description and date)
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

;Load pre—configured files for the ports, and memory adresses.
.include "m2560def.inc"

ldi r16, 0xFF  ;Set data direction registers.
out DDRB, r16  ;Set B port as output ports

ldi r16, 0x00  ;Set data direction registers.
out DDRD, r16  ;Set D port as input ports

;LED have 0 as on, 1 as off which is opposite to normal lamps
;Hereby we command second position of the lED to be on.

ldi r16, 0xFF
out portB, r16
;initialisation is done. All LEDs are off now.

;We create a loop which constantly listens on the switch ports

ldi r17, 0b11011111 ;Our desired input value
ldi r18, 0b11111110 ;Command that lights LED 0
listening_to_switch_loop:
     in r16, PIND   ;Read switch D and saves into register number 15

     cp r16, r17    ;Do compare with our desired value, which is pressing SWITCH 5
     breq equal
rjmp listening_to_switch_loop

equal: out portB, r18
```

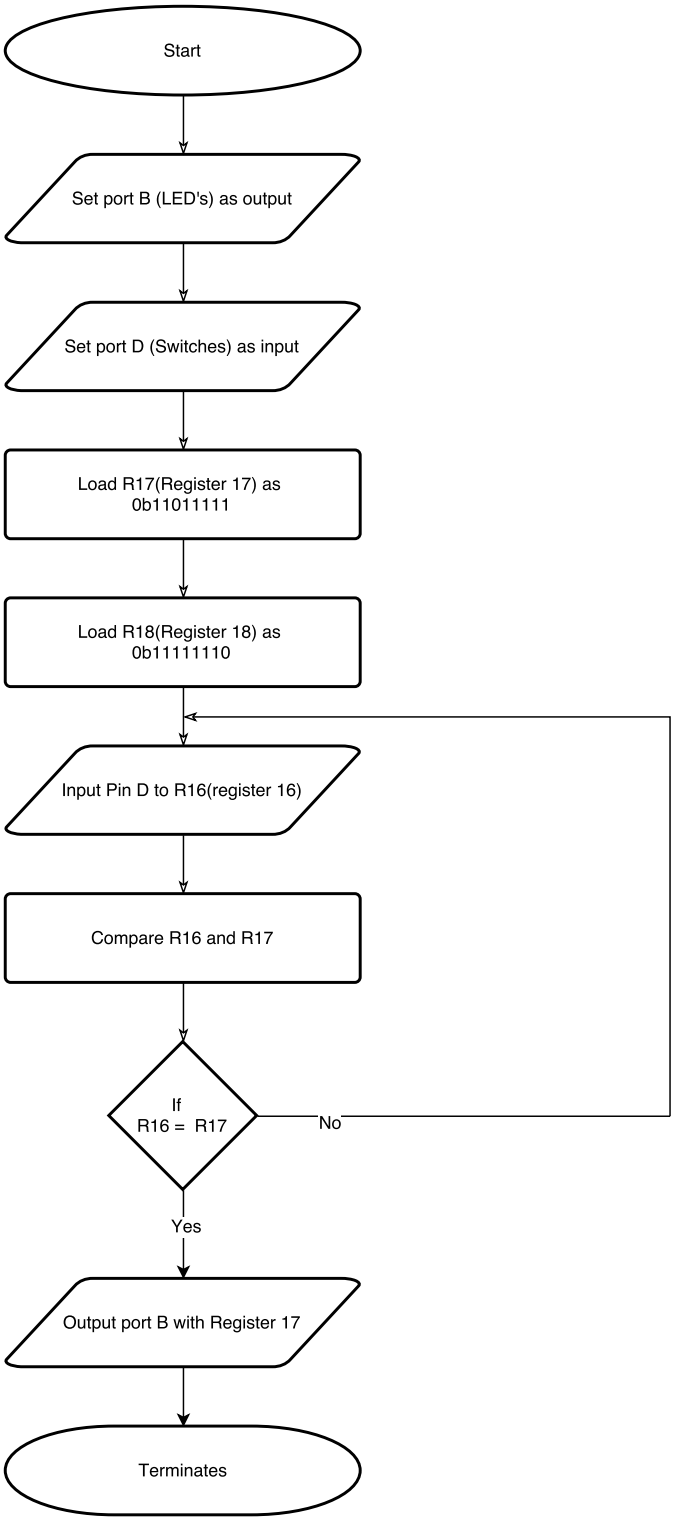

Figure 3: Flowchart of task 3

5 Task 5

Code and a flowchart for the fifth task is presented. In order for our code to have interval of 0.5 seconds, CPU clock needs to be configured with 4MHz. On the debugging simulator it returned with 538073,00 micro seconds.

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;     1DT301, Computer Technology I
;     Date: 2016-09-05
;     Author:
;     Songho Lee
;     Sarpreet Singh
;
;
;     Lab number: 1
;     Title: Task 5
;
;     Hardware: STK600, CPU ATmega2560
;
;     Function: Ligntning LED corresponding with Ring counter with interval of 0.5 seconds
;
;     Input ports: None. This is handled by the loop inside the programme.
;
;     Output ports: On-board LEDs are connected to PORTB.
;
;     Subroutines: If applicable.
;     Included files: m2560def.inc
;
;     Other information:
;     Changes in program: (Description and date)
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

;Load pre-configured files for the ports, and memory adresses.
.include "m2560def.inc"

ldi r16, 0xFF  ;Set data direction registers.
out DDRB, r16  ;Set B port as output ports

;LED have 0 as on, 1 as off which is opposite to normal lamps
;Hereby we command second position of the lED to be on.


ldi r17,0b00000001
ldi r25,0b00000000
firstloop:

mov r16, r17
com r16
out portB, r16

LSL r17
ldi r18, 80;   r18 to be our N
;; Qestion: AVR does not jump in the loop when it was set as 250, but it worked with 100
ldi r23, 0;         r23 to be our k counter, 0
superdelay:
     ldi r19, 0;         r19 to be our i, which is 0
     outerdelay:
     ldi r24, 0;         r24 to bbe our j, which is 0

     innerdelay1:
```

```
        inc r24
        cp r18, r24
        brge innerdelay1

        inc r19
        cp r18, r19
        brge outerdelay

        inc r23
        cp r18, r23
        brge superdelay

cp r17, r25
breq equal

rjmp firstloop
equal: ldi r17,0b00000001
```

Since delay part of the task five requires complicated control sequences, hereby we divide the chart into two parts: part before the delay, and after the delay.

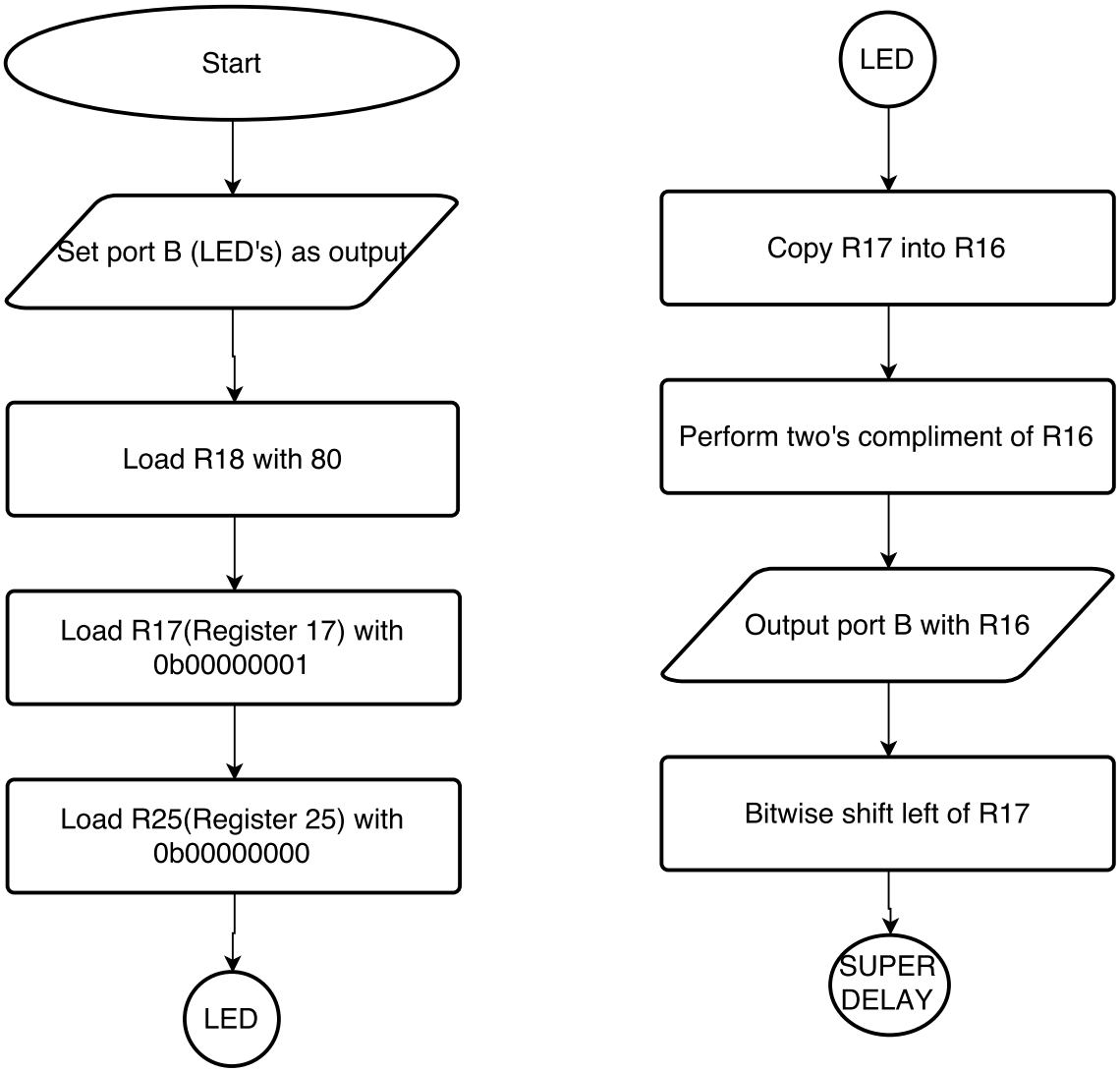

Figure 4: Flowchart of task 5 part 1

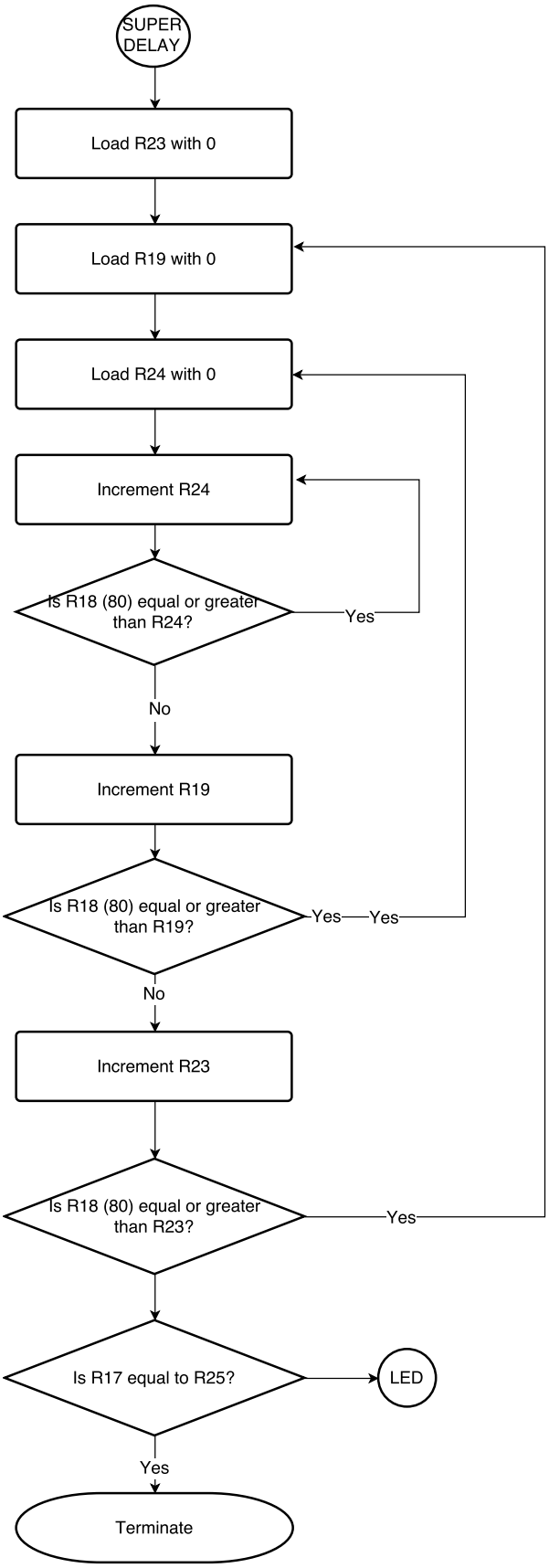

Figure 5: Flowchart of task 5 part 2

6 Task 6

Code and a flowchart for the sixth task is presented. In order for our code to have interval of 0.5 seconds, CPU clock needs to be configured with 4MHz. On the debugging simulator it returned with 538073,00 micro seconds.

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;     1DT301, Computer Technology I
;     Date: 2016-09-05
;     Author:
;     Songho Lee
;     Sarpreet Singh
;
;
;     Lab number: 1
;     Title: Task 6
;
;     Hardware: STK600, CPU ATmega2560
;
;     Function: It lights LED with Johnson Counter with intervals
;
;     Input ports: None. This is handled by the loop inside the programme.
;
;     Output ports: On-board LEDs are connected to PORTB.
;
;     Subroutines: If applicable.
;     Included files: m2560def.inc
;
;     Other information:
;     Changes in program: (Description and date)
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

;Load pre-configured files for the ports, and memory adresses.
.include "m2560def.inc"

;Initialize SP, Stack Pointer
ldi r20, HIGH(RAMEND)     ; R20 = high part of RAMEND address
out SPH,R20    ;SPH = high part of RAMEND address
ldi R20, low(RAMEND)      ; R20 = low part of RAMEND address
out SPL,R20

ldi r16, 0xFF  ;Set data direction registers.
out DDRB, r16  ;Set B port as output ports

;LED have 0 as on, 1 as off which is opposite to normal lamps
;Hereby we command second position of the lED to be on.

ldi r17,0b00000001
mov r21, r17   ;Registering 00000001 on both register 17, and 21.
ldi r25,0b11111111
ldi r22,0

mainloop:



firstloop:

;;THERE WAS DELAY HERE

```

```
ldi r18, 80;   r18 to be our N
ldi r23, 0;          r23 to be our k counter, 0
superdelay1:
     ldi r19, 0;          r19 to be our i, which is 0
     outerdelay1:
     ldi r24, 0;          r24 to bbe our j, which is 0

     innerdelay1:
     inc r24
     cp r18, r24
     brge innerdelay1

     inc r19
     cp r18, r19
     brge outerdelay1

     inc r23
     cp r18, r23
     brge superdelay1

;;DELAY ENDS

;These operations for the LED
mov r16, r17
com r16
out portB, r16

cpi r22,0xFF
breq shiftright

;COMPARISION
shiftleft:

LSL r17
add r17, r21

cpi r22,0xFF
brne begincompare

shiftright:
LSR r17

begincompare:
cp r17, r25
breq equal


rjmp firstloop

equal:
com r25
com r22

rjmp mainloop
```

Since delay part of the task six requires complicated control sequences, hereby we divide the chart into three parts.

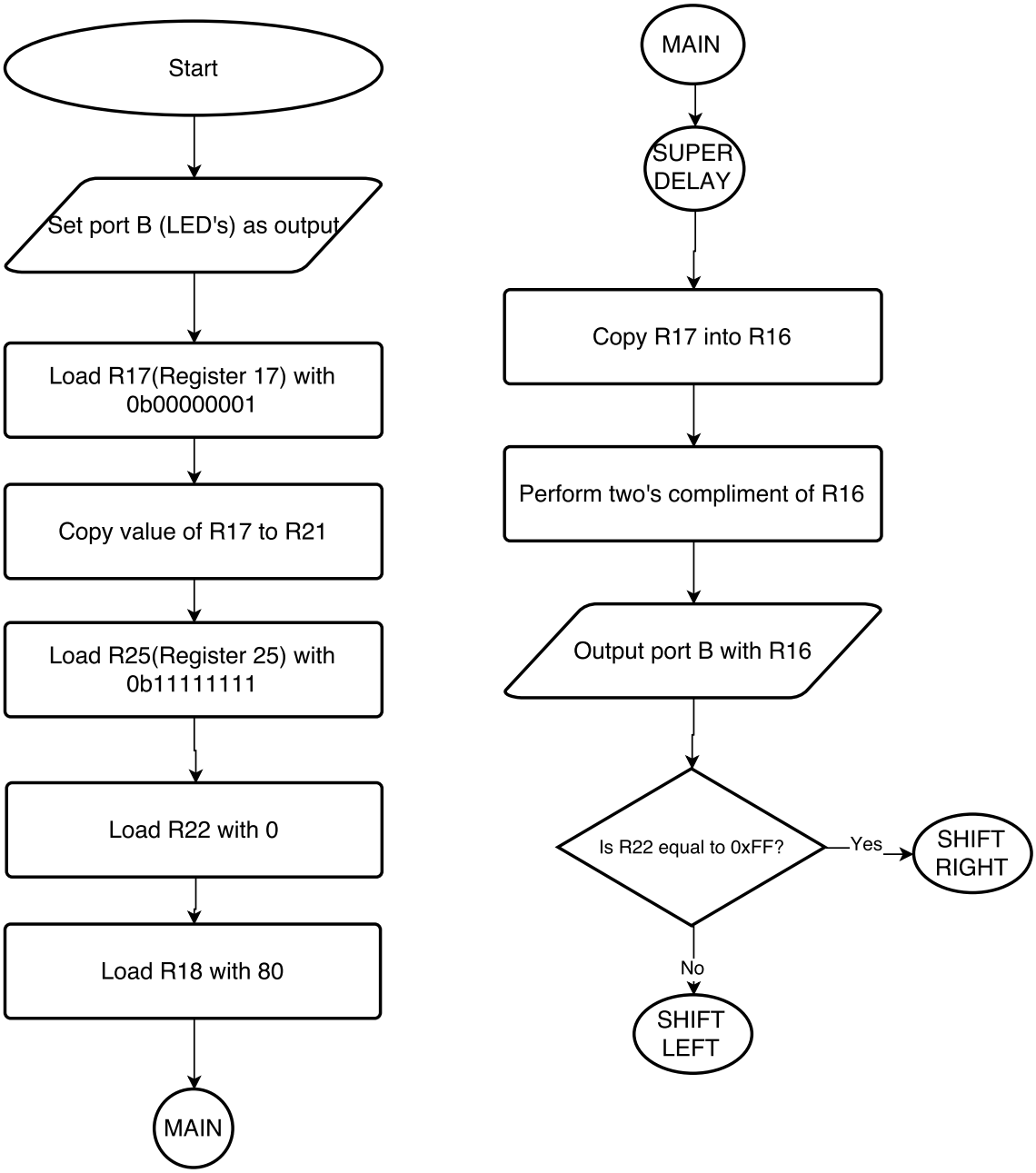

Figure 6: Flowchart of task 6 part 1

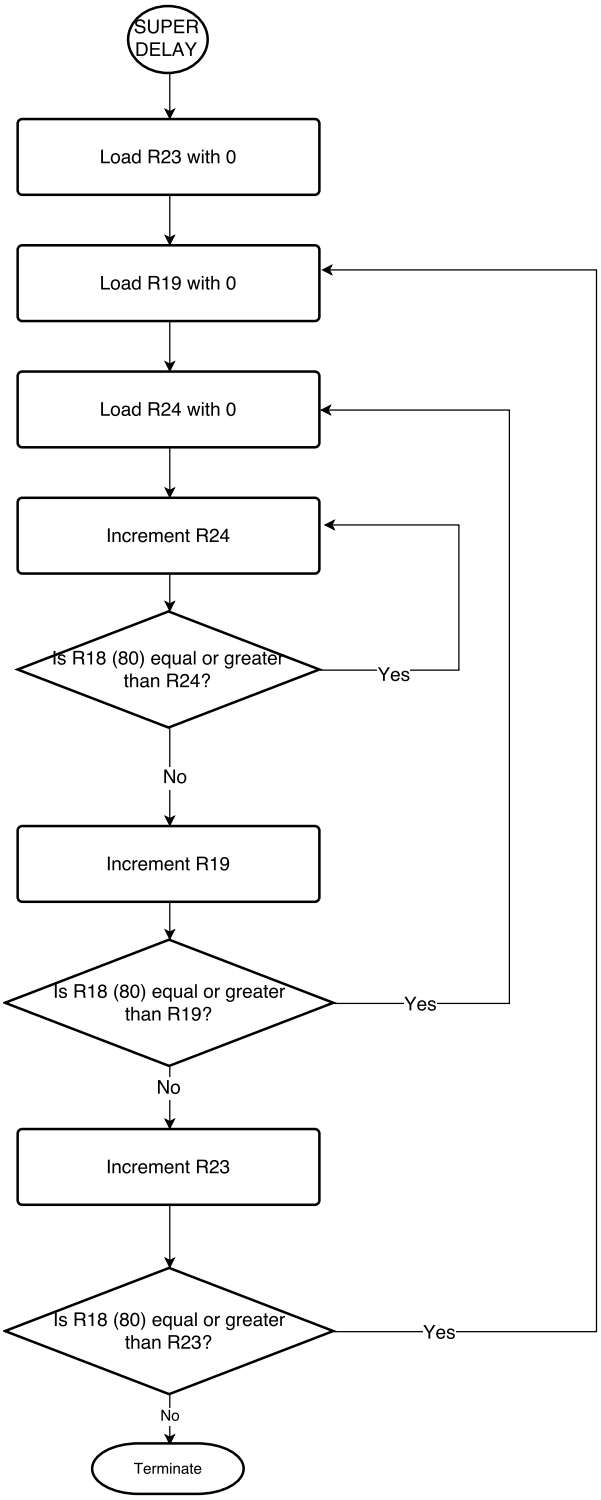

Figure 7: Flowchart of task 6 part 2

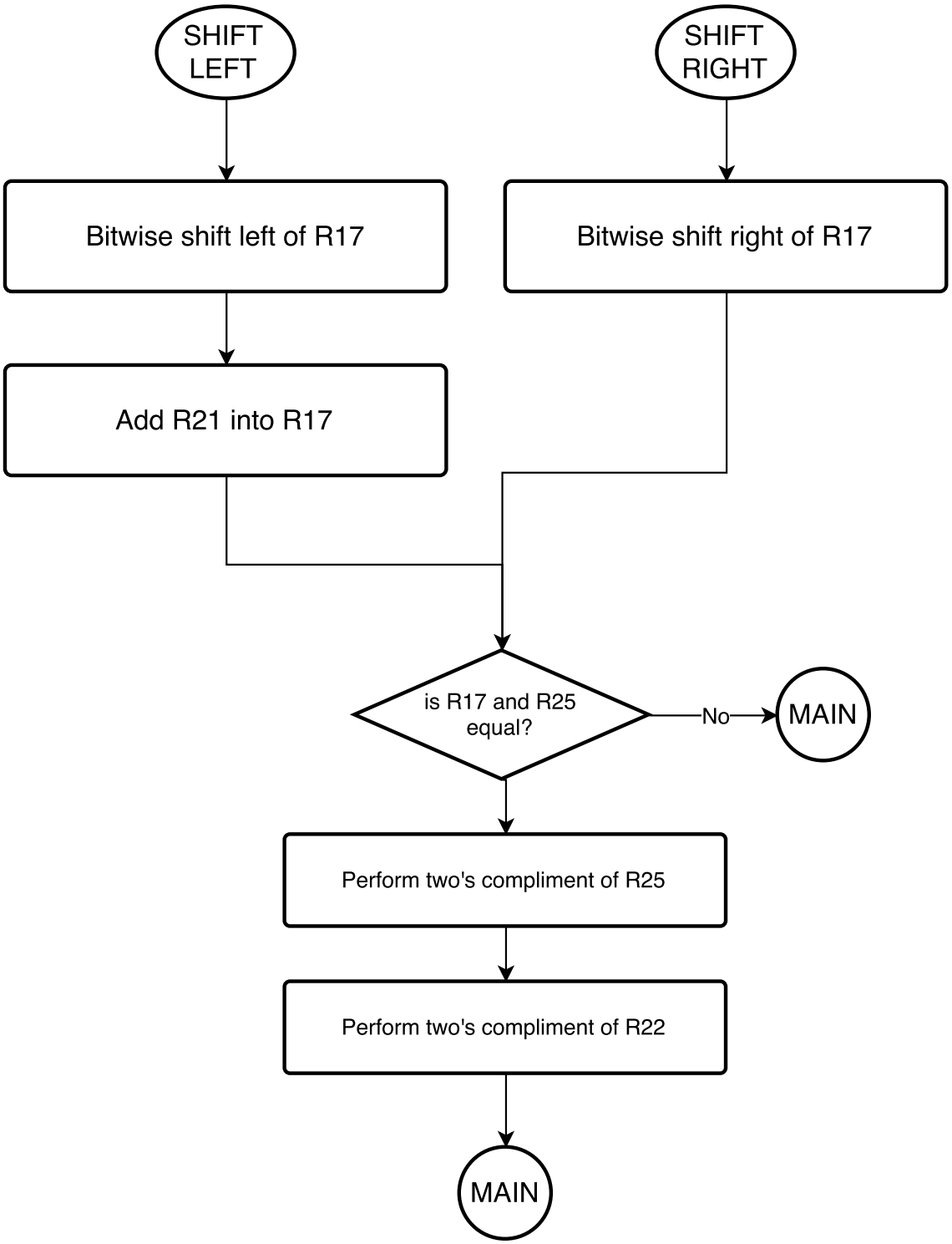

Figure 8: Flowchart of task 6 part 3