LAB 1-2

DELAY WITH INSTRUCTIONS

Name: LƯƠNG THÀNH VỸ (2151280)
TRẦN TRUNG TOÀN (2151265)

OBJECTIVE:

- Execute instructions to create delays using subroutines.
- Perform communication with shift registers.

REFERENCES:

- Lab manual chapter 1-2.

EXPERIMENT 1:

a) Use the following program:

```
.include "m324PAdef.inc"
.org   00
       ldi r16,0x01
       out    DDRA, r16
start:
       sbi    PORTA,PINA0
       cbi    PORTA, PINA0

       rjmp start
```

Connect PA0 to a measurement channel on the TEST STATION and measure pulse forms using an oscilloscope.

EXPERIMENT 2:

a) Write a subroutine Delay1ms and use it to write a program to generate a 1KHz square wave on PA0.

b) Use this subroutine to write subroutines Delay10ms, Delay100ms, and Delay1s.

c) Use the Delay1s subroutine to write a program to blink/turn off an LED connected to PA0.

EXPERIMENT 3:

a) Connect the necessary signals from an AVR port to the control signals of the shift register on header J13. Connect the output of the shift register to a LED bar.

b) Using the sample programs from the experiment guide, write a program to create a gradually lit LED effect from left to right, then gradually turn them off from left to right after every 500ms.

LAB REPORT

Group:

Class Group: *Subject:*

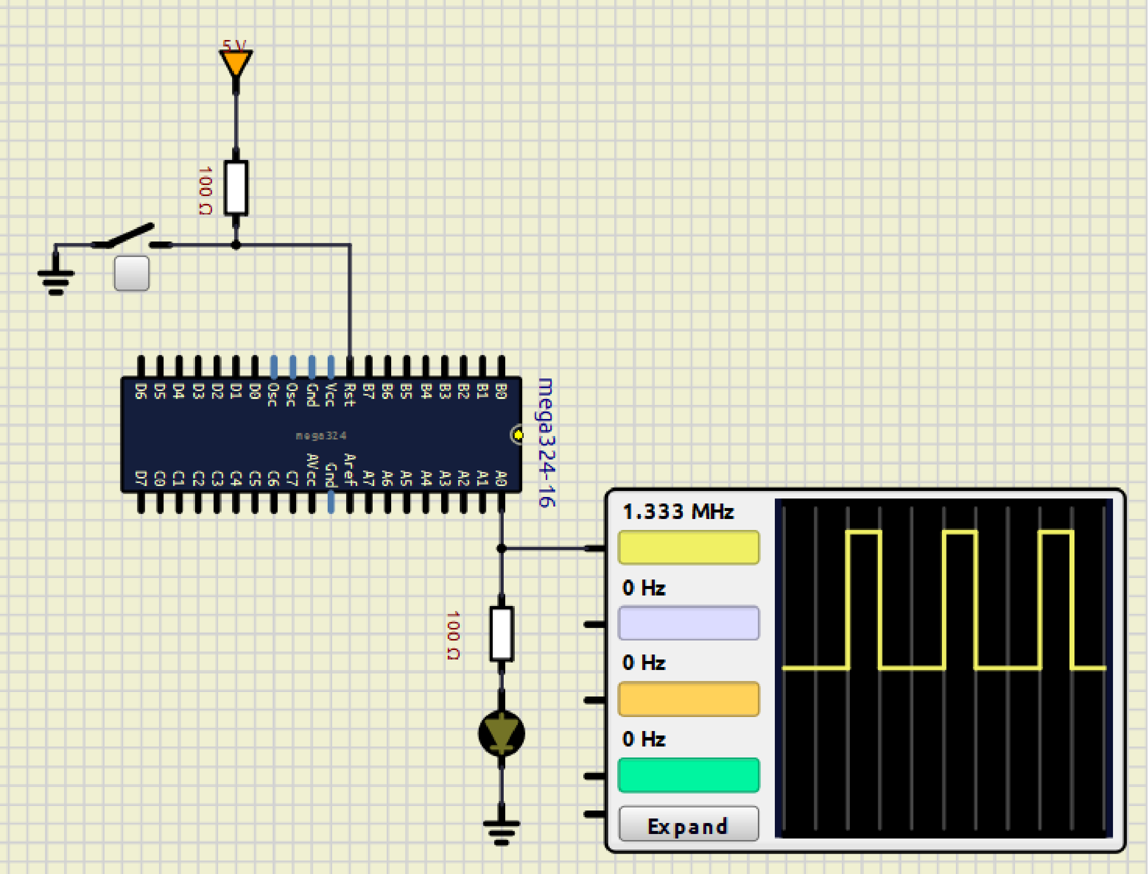
EXPERIMENT 1:

1. Answer the following questions:

   a. Capture a pulse waveform on PA0.

ANSWER:

We didn’t have access to an oscilloscope in the laboratory, so we captured the waveform using a simulation instead.


5V
100 Ω
mega324
mega324-16
D6 D5 D4 D3 D2 D1 D0 Osc Osc Gnd Vcc Rst B7 B6 B5 B4 B3 B2 B1 B0
D7 C0 C1 C2 C3 C4 C5 C6 C7 AVcc Gnd Aref A7 A6 A5 A4 A3 A2 A1 A0
100 Ω
1.333 MHz
0 Hz
0 Hz
0 Hz
Expand

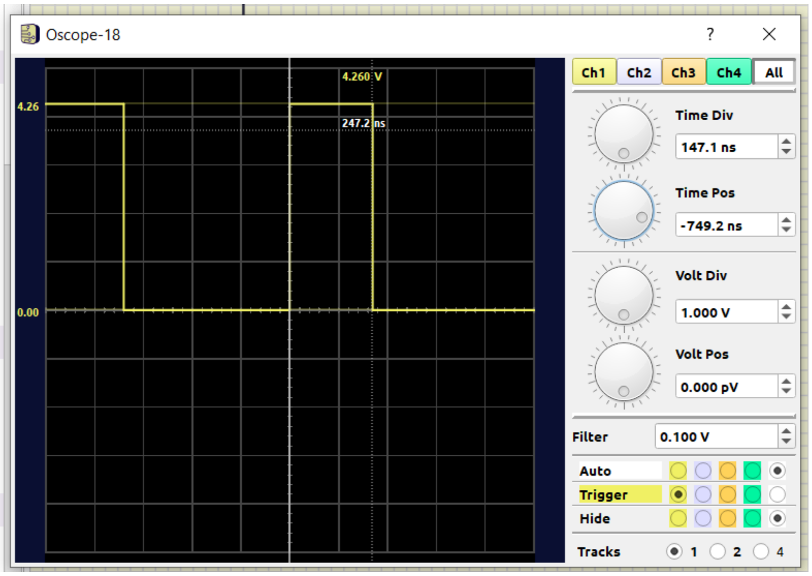
   b. What is the frequency, duration of the high signal, and duration of the low signal?

ANSWER:

```
.include "m324PAdef.inc"
.org    00
        ldi r16,0x01
        out     DDRA, r16
start:
        sbi     PORTA,PINA0
        cbi     PORTA, PINA0
        rjmp start
```

ANSWER:

The frquency observed is 1.333 MHz. And the HIGH duration of the signal is 247 ns, the LOW duration is 496 ns.



Time Div 147.1 ns

Time Pos -749.2 ns

Volt Div 1.000 V

Volt Pos 0.000 pV

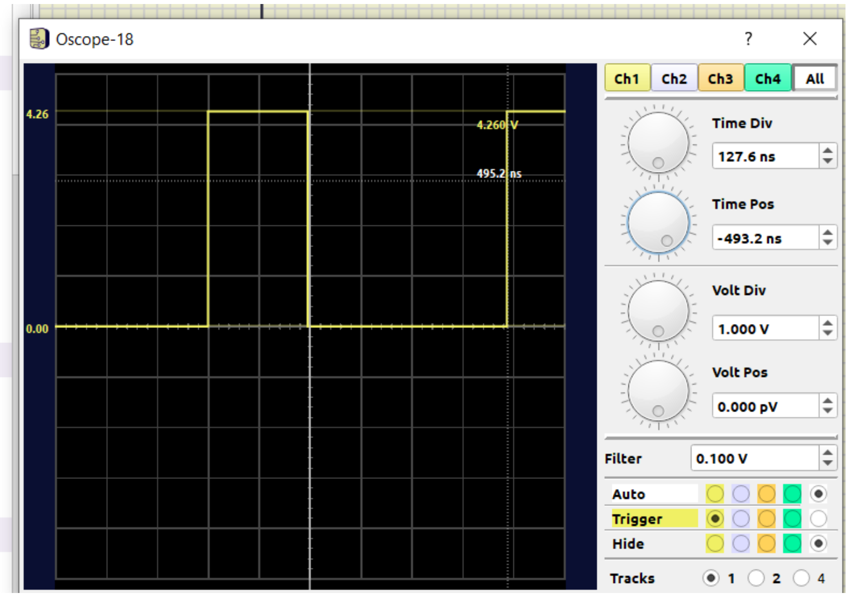
Filter 0.100 V

Auto ○ ○ ○ ○ ●

Trigger ● ○ ○ ○ ○

Hide ○ ○ ○ ○ ●

Tracks ● 1 ○ 2 ○ 4



Time Div 127.6 ns

Time Pos -493.2 ns

Volt Div 1.000 V

Volt Pos 0.000 pV

Filter 0.100 V

Auto ○ ○ ○ ○ ●

Trigger ● ○ ○ ○ ○

Hide ○ ○ ○ ○ ●

Tracks ● 1 ○ 2 ○ 4

c. Explain the measured results.

ANSWER:

The code creates a loop that toggles pin PA0 on and off using two consecutive instructions, generating a PWM waveform. Each loop iteration consists of three instructions: SBI, CBI, and RJMP, each requiring 2 cycles to execute. After setting the signal HIGH with SBI, the signal goes LOW 2 cycles later with CBI. However, it takes 4 cycles for the signal to go HIGH again, due to the RJMP and SBI instructions within the loop. This results in a LOW duration that is twice as long as the HIGH duration, creating a 33% duty cycle PWM.

The total period of the waveform is 6 cycles. Operating at 8 MHz, each cycle takes 0.125 µs, giving a period $T\ =\ 6 \times 0.125 = 0.75\ us$. Thus, the frequency is:

$$f = \frac{1}{T} = \frac{1}{0.75 \times 10^{-6}}\ =\ 1\ 333\ 333.33\ Hz\ =\ 1.333\ MHz.$$

The HIGH duration of the signal is $33\% \times 0.75 = 0.2475\ us$, and the LOW duration is $0.75 - 0.2475 = 0.5025$ us. These result matching the waveform observed on the oscilloscope.

EXPERIMENT 2:

1. Answer the following questions:

   a. How to calculate the number of machine cycles needed to execute the Delay1ms subroutine. Present a simulation image.

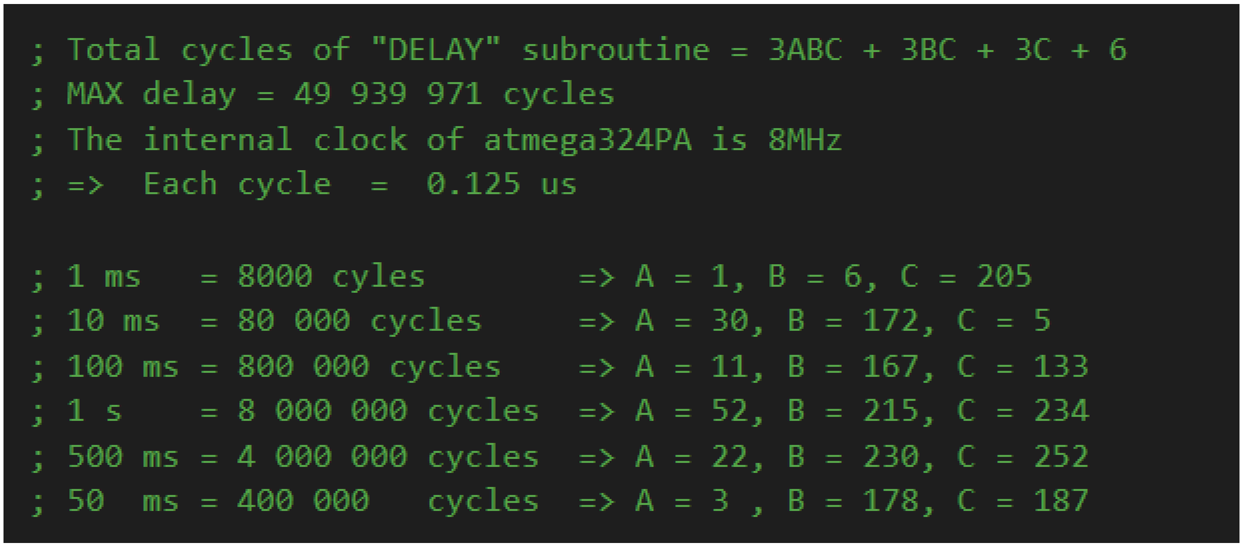
ANSWER:

At an operating frequency of 8 MHz, the AVR takes 0.125 µs per cycle. To achieve a delay of 1 ms, we need to create a subroutine that executes $\frac{1000}{0.125} = 8000$ cycles.

In our subroutine (shown in the source code below), we use 3 nested loops to generate a delay, with each loop iterating A, B, and C times, respectively. After writing the subroutine, we count the cycles executed by each instruction and loop iteration. From this, we derive a function for the delay in cycles as: 3ABC+3BC+3C+6

The PA0 pin experiences a 1 ms delay between each toggle, producing a square wave with a period of 2 ms. The frequency of this waveform is $\frac{1}{T} =\ 500\ Hz$.

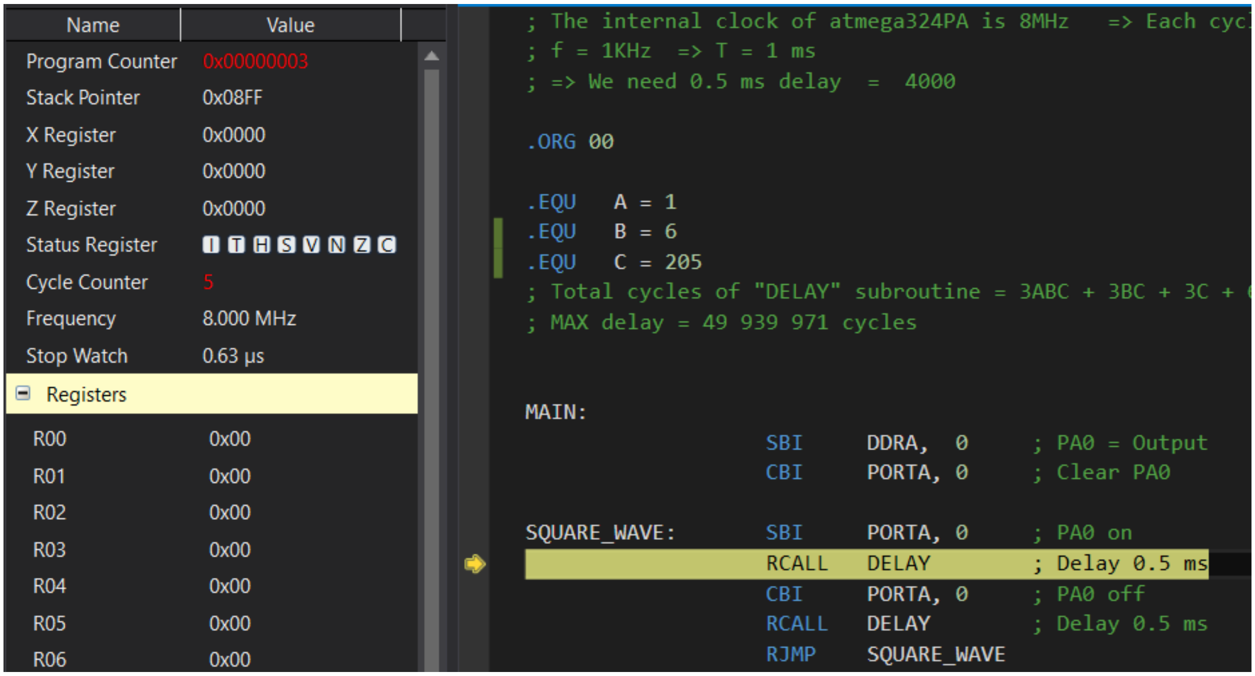
To achieve an 8000-cycle delay, we can assign random values to any two of A, B, or C and solve for the third. Alternatively, we can use a programming language like Python or C to iterate through all possible values of A, B, and C (from 0 to 255) and find the combination with the smallest difference from the target cycle count. Using this method, we have the delay subdroutine for these amount of delay as show below:



```
; Total cycles of "DELAY" subroutine = 3ABC + 3BC + 3C + 6
; MAX delay = 49 939 971 cycles
; The internal clock of atmega324PA is 8MHz
; =>  Each cycle  =  0.125 us

; 1 ms   = 8000 cyles          => A = 1, B = 6, C = 205
; 10 ms  = 80 000 cycles       => A = 30, B = 172, C = 5
; 100 ms = 800 000 cycles      => A = 11, B = 167, C = 133
; 1 s    = 8 000 000 cycles    => A = 52, B = 215, C = 234
; 500 ms = 4 000 000 cycles    => A = 22, B = 230, C = 252
; 50  ms = 400 000   cycles    => A = 3 , B = 178, C = 187
```

b. Image of a 1KHz square wave on PA0.

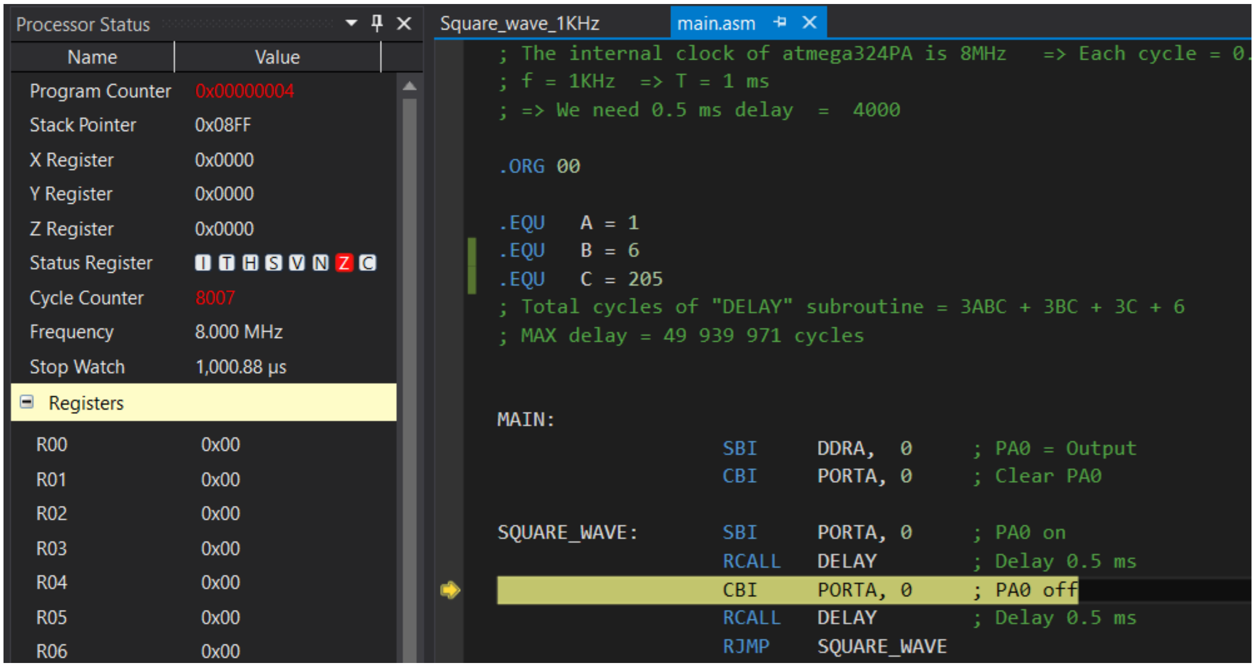
- **Before DELAY Suubroutine:**



Name	Value
Program Counter	0x00000003
Stack Pointer	0x08FF
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	5
Frequency	8.000 MHz
Stop Watch	0.63 µs
⊟ Registers	
R00	0x00
R01	0x00
R02	0x00
R03	0x00
R04	0x00
R05	0x00
R06	0x00

```
; The internal clock of atmega324PA is 8MHz   => Each cyc
; f = 1KHz   => T = 1 ms
; => We need 0.5 ms delay   =   4000

.ORG 00

.EQU    A = 1
.EQU    B = 6
.EQU    C = 205
; Total cycles of "DELAY" subroutine = 3ABC + 3BC + 3C + (
; MAX delay = 49 939 971 cycles


MAIN:
                SBI     DDRA,  0      ; PA0 = Output
                CBI     PORTA, 0      ; Clear PA0

SQUARE_WAVE:    SBI     PORTA, 0      ; PA0 on
                RCALL   DELAY         ; Delay 0.5 ms
                CBI     PORTA, 0      ; PA0 off
                RCALL   DELAY         ; Delay 0.5 ms
                RJMP    SQUARE_WAVE
```

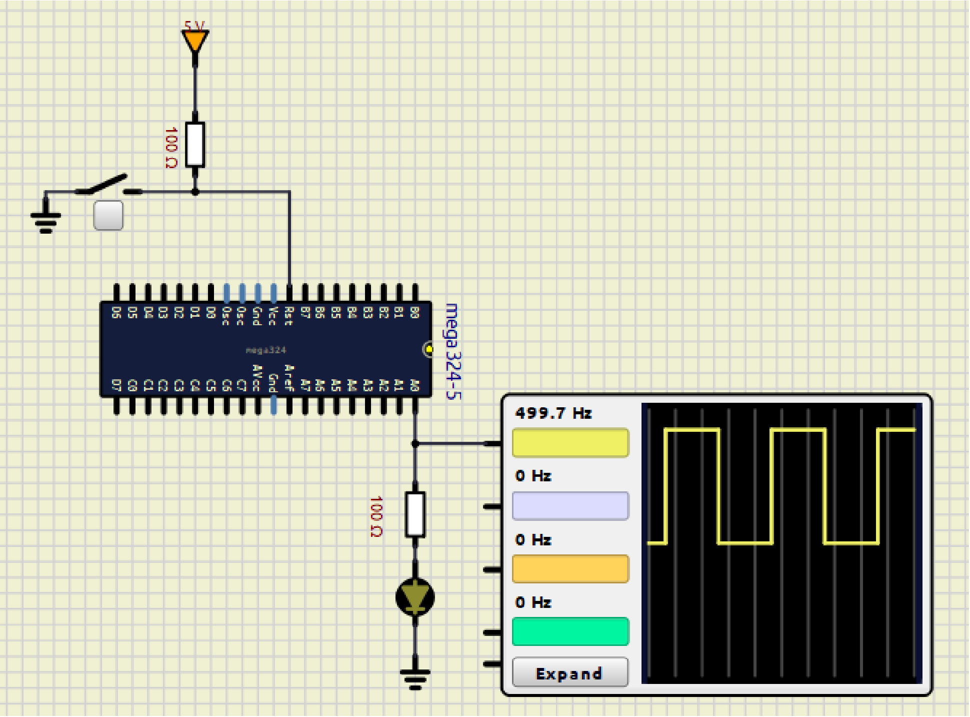
- **After DELAY subroutine:**



Name	Value
Program Counter	0x00000004
Stack Pointer	0x08FF
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	8007
Frequency	8.000 MHz
Stop Watch	1,000.88 µs
Registers	
R00	0x00
R01	0x00
R02	0x00
R03	0x00
R04	0x00
R05	0x00
R06	0x00

```
; The internal clock of atmega324PA is 8MHz   => Each cycle = 0.
; f = 1KHz  => T = 1 ms
; => We need 0.5 ms delay  =  4000

.ORG 00

.EQU   A = 1
.EQU   B = 6
.EQU   C = 205
; Total cycles of "DELAY" subroutine = 3ABC + 3BC + 3C + 6
; MAX delay = 49 939 971 cycles


MAIN:
                  SBI     DDRA,  0      ; PA0 = Output
                  CBI     PORTA, 0      ; Clear PA0

SQUARE_WAVE:      SBI     PORTA, 0      ; PA0 on
                  RCALL   DELAY         ; Delay 0.5 ms
                  CBI     PORTA, 0      ; PA0 off
                  RCALL   DELAY         ; Delay 0.5 ms
                  RJMP    SQUARE_WAVE
```

- **Waveform on oscilloscope:**



c. What is the error?

ANSWER:

The observed frequency is 499.7 Hz, indicating an error of 0.3 Hz, or $(1-\frac{499.7}{500})\times 100\% =$ 0.06 % , which is highly precise . This slight imperfection may be due to the RJMP instruction within the loop generating the waveform.

2. Source code for 2.c with comments.

ANSWER:

```
; The internal clock of atmega324PA is 8MHz   => Each cycle = 0.125 us
; f = 1KHz  => T = 1 ms
; => We need 1 ms delay  =  8000
; 1 ms   = 8000 cyles        => A = 1, B = 6, C = 205

.ORG 00
.EQU   A = 1
.EQU   B = 6
.EQU   C = 205
; Total cycles of "DELAY" subroutine = 3ABC + 3BC + 3C + 6

MAIN:
              SBI     DDRA,  0     ; PA0 = Output
              CBI     PORTA, 0     ; Clear PA0

SQUARE_WAVE:  SBI     PORTA, 0     ; PA0 on
              RCALL   DELAY        ; Delay 1 ms
              CBI     PORTA, 0     ; PA0 off
              RCALL   DELAY        ; Delay 1 ms
              RJMP    SQUARE_WAVE


DELAY:                    ; # of Cycle of Instr
              LDI   R22, C      ; +1
L2:           LDI   R21, B      ; +1              }
L1:           LDI   R20, A      ; +1        }
L0:           DEC   R20         ; +1  } L0 = 3A
              BRNE  L0          ; +2  }
                                ; -1        } L1 = B* (L0 + 4 -1) = 3AB + 3B
              DEC   R21         ; +1        }
              BRNE  L1          ; +2        }
                                ; -1              } L2 = C*(L1+4-1) = 3ABC + 3BC + 3C
              DEC   R22         ; +1              }
              BRNE  L2          ; +2              }
                                ; -1                      } +3 and +3 for first 3 LDI
              RET               ; +4                      }

; ====>  Total cycle of "DELAY" = 3ABC + 3BC + 3C + 6
```

RESULT ON EXPERIMENT KIT:

LAB REPORT

Group:

Class Group: *Subject:*

In this Experiment, we modify the DELAY subroutine to delay 1s, which create the sqaure. This cause the LED to turn on and off each after 1 second.

https://drive.google.com/file/d/10wZPJvxVltNhq05imKVNCjv54i7hUQFc/view?usp=sharing

EXPERIMENT 3:

1. Answer the following questions:

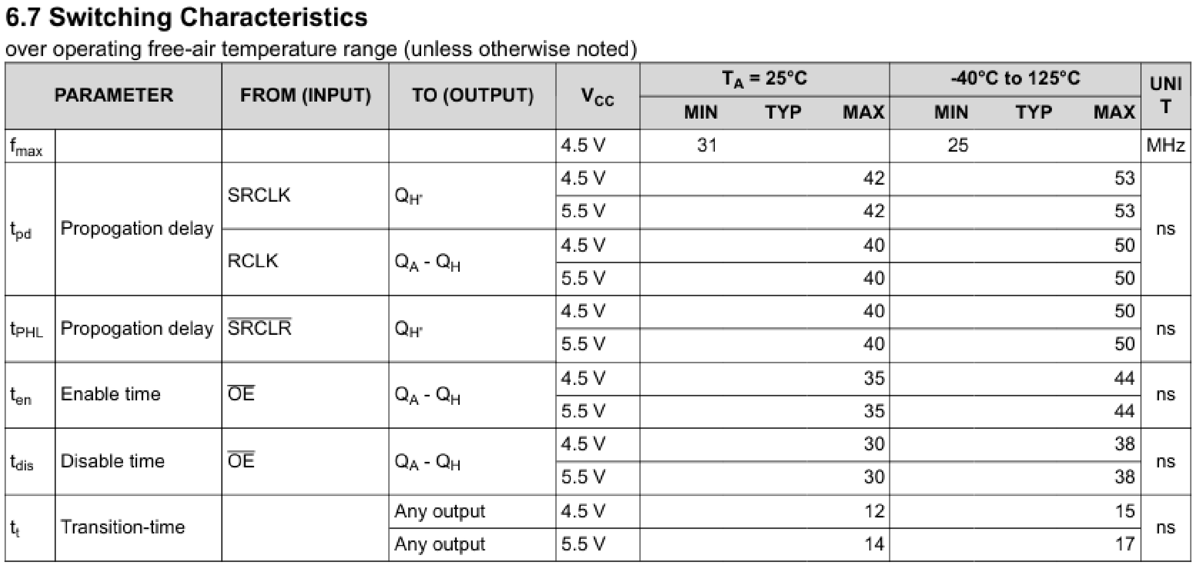
   a. Describe the connections on the experimental kit.

ANSWER:

On the experiment kit, there are 4 pins for the SRCLK, RCLK, OE, and DS signals to control the 74HC595 shift register. However, since the 74HC595 is a right shift register, we can only display LEDs lighting up from right to left, as the serial input bit is shifted in from the right. To make the LEDs light up from left to right, we need to reverse the connections between the shift register outputs and the LEDs. This means connecting the LSB output to the far-left LED and the MSB output to the far-right LED.

   b. According to the datasheet of 74HC595, what is the highest clock frequency it can operate at?

ANSWER:



6.7 Switching Characteristics

over operating free-air temperature range (unless otherwise noted)

PARAMETER		FROM (INPUT)	TO (OUTPUT)	V_{CC}	T_A = 25°C			-40°C to 125°C			UNI T
					MIN	TYP	MAX	MIN	TYP	MAX	
f_{max}				4.5 V	31			25			MHz
t_{pd}	Propogation delay	SRCLK	$Q_{H'}$	4.5 V			42			53	ns
				5.5 V			42			53	
		RCLK	Q_A - Q_H	4.5 V			40			50	
				5.5 V			40			50	
t_{PHL}	Propogation delay	$\overline{SRCLR}$	$Q_{H'}$	4.5 V			40			50	ns
				5.5 V			40			50	
t_{en}	Enable time	$\overline{OE}$	Q_A - Q_H	4.5 V			35			44	ns
				5.5 V			35			44	
t_{dis}	Disable time	$\overline{OE}$	Q_A - Q_H	4.5 V			30			38	ns
				5.5 V			30			38	
t_t	Transition-time		Any output	4.5 V			12			15	ns
			Any output	5.5 V			14			17	

The parameter f_{max} specified in the switch characteristic is 31 MHz at room temperature (T = 25^{o}C). This means that maximum operating frequency of 74HC595 IC at Normal condition is 32 MHz.

c. How do you expand the display to 16 LEDs?

ANSWER:

To display 16 LEDs, we'll need two 74HC595 shift registers. We can connect these two 8-bit shift registers by linking the Q7' (Serial Out) pin of the first register to the DS (Serial In) pin of the second register. Both registers will share the same RCLK, SRCLK clock signals, and OE signal, effectively creating a 16-bit shift register. We then control the RCLK, SRCLK, and OE signals as we would for a single 8-bit shift register, but the output now extends to 16 bits. This allows us to connect and control 16 LEDs.

However, on the experiment kit, there is only one LED bar and a single 74HC595 IC, so we displayed output on just 8 LEDs.

2. Source code with comments.

ANSWER:

```
; The internal clock of atmega324PA is 8MHz
; =>  Each cycle  =  0.125 us
; 500 ms = 4 000 000 cycles   => A = 22, B = 230, C = 252

; PA0 = SRCLK
; PA1 = RCLK
; PA2 = DS or SER (serial input)
; PA3 = SRCLR  (active LOW clear shift register)

.ORG 00

MAIN:
                       LDI     R16,    0x0F   ; Config P0 -> P3 as OUTPUT
                       OUT     DDRA,   R16

                       CBI     PORTA,  3      ; Pulse SRCLR pin to (clear shift
register)
                       SBI     PORTA,  3      ; SRCLR pin is ACTIVE LOW

; Turn on LED from Right to LEFT
CHASE_UP:     LDI     R17,    0x08   ; Counter for number of bit sent (8-bit)
              SBI     PORTA,  2      ; PA2 = DS = 1

LOOP_UP:      SBI     PORTA,  0      ; pulse PA0 = SRCLK
```

```
              CBI     PORTA,  0

              SBI     PORTA,  1      ; pulse PA1 = RCLK
              CBI     PORTA,  1

              RCALL   DELAY          ; Delay 500 ms
              DEC     R17            ; Check if we have sent 8-bit of data yet ?
              BRNE    LOOP_UP
              RJMP    CHASE_DOWN


; Turn off LED from Right to LEFT
CHASE_DOWN:   LDI     R17,    0x08   ; Counter for number of bit sent (8-bit)
              CBI     PORTA,  2      ; PA2 = DS = 0

LOOP_DOWN:
       SBI     PORTA,  0      ; pulse PA0 = SRCLK
       CBI     PORTA,  0

       SBI     PORTA,  1      ; pulse PA1 = RCLK
       CBI     PORTA,  1

       RCALL   DELAY               ; Delay 500 ms
       DEC     R17            ; Check if we have sent 8-bit of data yet ?
       BRNE    LOOP_DOWN
       RJMP    CHASE_UP



DELAY:                   ; # of Cycle of Instr
       LDI   R22, 22     ; +1
L2:    LDI   R21, 230    ; +1                 }
L1:    LDI   R20, 252    ; +1           }
L0:    DEC   R20         ; +1   } L0 = 3A
       BRNE  L0          ; +2   }
                         ; -1           } L1 = B* (L0 + 4 -1) = 3AB + 3B
       DEC   R21         ; +1           }
       BRNE  L1          ; +2           }
                         ; -1                 } L2 = C* (L1 + 4 -1) = 3ABC + 3BC + 3C
       DEC   R22         ; +1                 }
       BRNE  L2          ; +2                 }
                         ; -1                         } +3 and +3 for the first 3 LDI
       RET               ; +4                         }

; ====>  Total cycle of "DELAY" = 3ABC + 3BC + 3C + 6
```

RESULT ON EXPERIMENT KIT:

https://drive.google.com/file/d/1jlv9pvt4eus4BAlVC4lm9vk41Wagdr98/view?usp=sharing