Design Example

In this design example we are going to design a Slot Machine

Assume Port D is wired as shown in the table below.

		Direction	DDRD bit(s)	Type / Initialization	PORTD
PD bits 3-0	switches	Input	000	Passive Input	1
PD bit 4	Win light	Output	1	Initially Off	0
PD bit 5	new Account button	Input	0	Active output of a DFF	0
PD bit 6	add Account button	Input	0	Active output of a DFF	0
PD bit 7	Lose light	Output	1	Initially Off	0

We begin by defining SRAM variable account and initializing Port D as defined in the table.

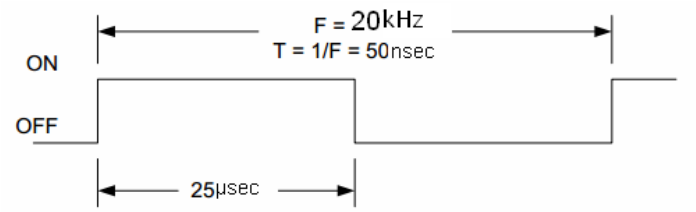
```
account     .BYTE 1

ldi r16, 0b10010000
out DDRD, r16
ldi r16, 0b00001111
out PORTD,r16
```

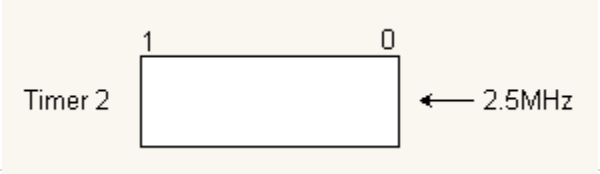
Now let's generate a clock to the two D flip-flops so we can read our new and add account button.

Given clock is 20 Mhz. I want to clock the DFF at a Frequency is 20Khz.

Alternative wording: I want to cycle and test if the button is pressed every 50 microseconds.

If I want to use timer 2, what divide frequency will I need to do that? What would you need to load into Timer 2 to generate this delay?



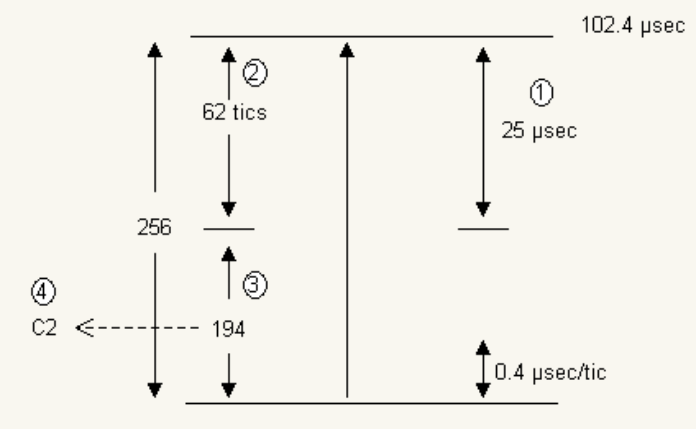


Calculate Max delay given the following information. You are using Timer 2, a clock frequency of f_{clk} = 20 MHz, and a clock divider of ÷8. Timer 2 is an 8-bit timer so the maximum number of tics is 2^8 = 256 tics. To convert to time we need to equate tics to time.

$f_{1/0} = f_{clk}/8 = 2.5\text{MHz}$ $\quad t_{1/0}=1/f_{1/0} = 0.4\ \mu\text{sec/tic}$

$t_{max} = 0.4\mu\text{sec/tic} \times 256\ \text{tics} = 102.4\ \mu\text{sec}$

So our timer with given conditions can generate a 25 microseconds delay, now let's look at what we need to preload our counter with to get a delay of 25 microseconds.



25 µsec ÷ 0.4 µsec/tics = 62.5 tics (Rounding down we leave it at 62). So we would need to preload timer 2 with a value of 256 – 62 = 194_{10} = 0xC2

We can now use polling or an interrupt service routine to generate our clock.

Now let's generate the subroutines to be called when the user sets a dollar amount into 4-switches and presses the new account or add account button.

```
New_account:

      ; When the button is pressed show the new amount
      in    r16, PIND
      ldi   r16, 0xF0
      and   r8, r16
      sts   account, r8
      call  WriteDisplay            //r8 output to 7-seg
      ret

Add_account:

      ; Most Likely this is what would be tested on
      in   r16, PIND
      cbr  r16, 0xF0
      lds  r17, Account
      add  r17, r16
      sts  account, r17
      mov  r8, r17
      call Write Display
```

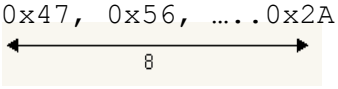
The Slot Machine Wheel

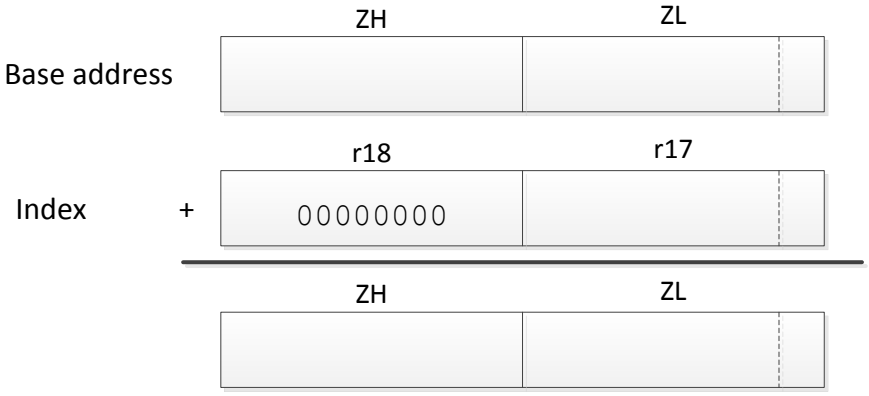
The strange characters to be generated by our slot machine are shown at left. In the table are the segments to be turned on/off to create the strange symbols.

db	g	f	e	d	c	b	a
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	0	1	0	1	0	1	0

Converting this table into bytes to be saved in Flash.

```
0x0123  wheel: .DB 0x47, 0x56, …..0x2A
```






```
Init:
      clr   r17
      clr   r18
loop: sbis   PIND, 4
      rjmp  loop
      inc   r17
      cbr   r17, 0xF8
      ldi   ZH, high(wheel<<1)
      ldi   ZL, low(wheel<<1)
      add   ZL, r17
      adc   ZH, r18
      lpm   r8, Z
      call  WriteDisplay
      rjmp  loop
      wheel: .DB 0x47, 0x56, …..0x2A
```