EMFLUX MOTORS PVT. LTD.
Embedded Internship Assessment Test

August 2024

PART A (All Questions are Compulsory)

1. Assuming the circuit starts at t = 0, plot a graph for Output Voltage vs Time t for the circuit as shown in Fig: 1.0, also mention the calculations involved. (Consider ideal Op Amp)

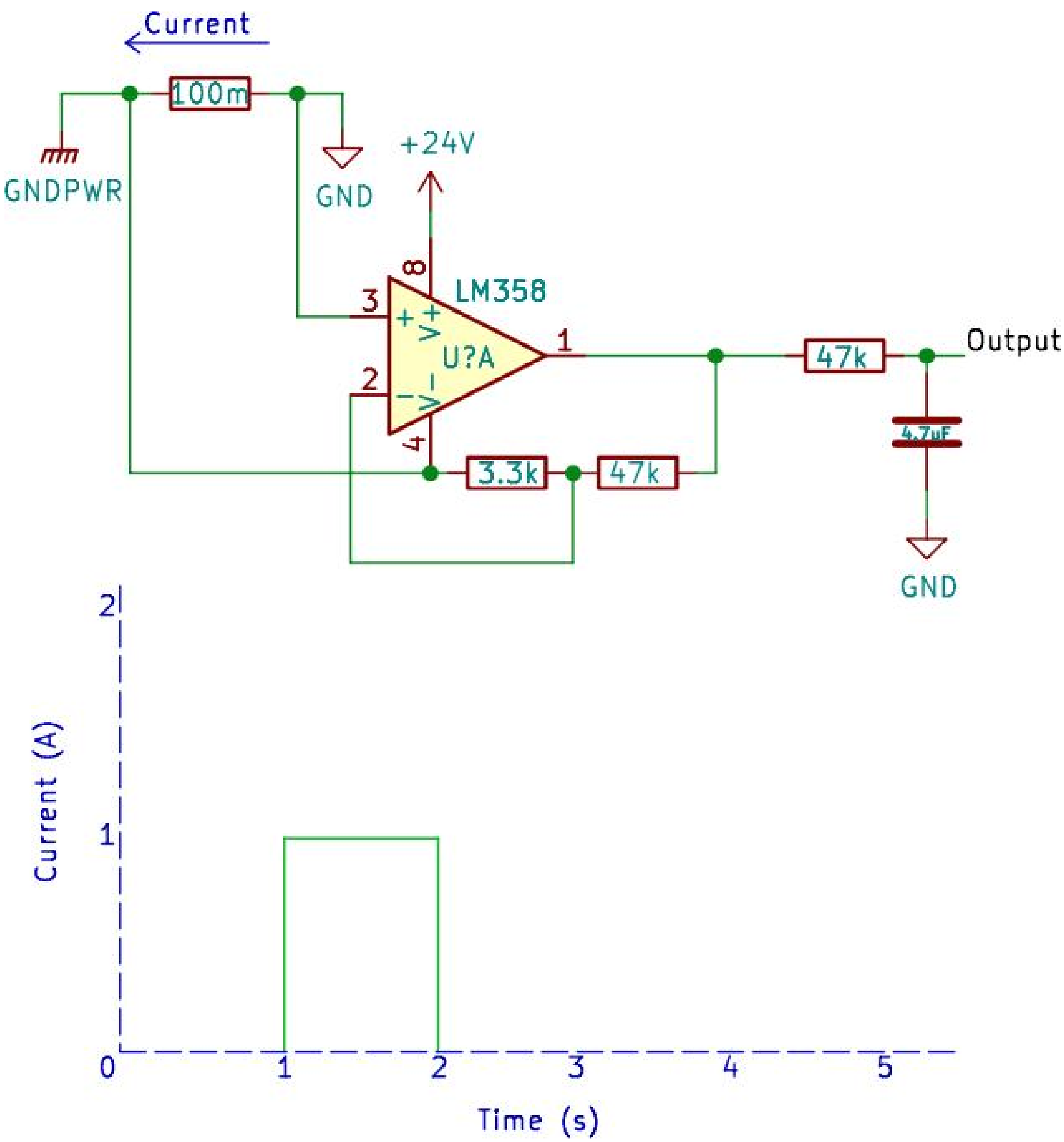


Fig: 1.0

Solution:-

Circuit Overview:-

- **Op-Amp:-** LM358 configured as a non-inverting amplifier.
- **Input:-** Current source with a pulse waveform (1A peak from 1 to 2 seconds).
- **Power Supply**:- +24V to the op-amp.
- **Resistors**:- Feedback resistors of 3.3kΩ and 47kΩ, indicating a gain of the amplifier.

Steps to Solve:-

❖ **Determine the Gain of the Op-Amp:-**

- The gain for a non-inverting amplifier is given by:-

$$\text{Gain} = 1 + \frac{Rf}{Sin}$$

Where:

- Rf = 47kΩ(feedback resistor)
- Sin= 3.3kΩ

Substituting the values:-

$$\text{Gain} = 1 + \frac{47k\Omega}{3.3k\Omega} = 1 + 14.24 \approx 15.24$$

❖ **Determine the Input Voltage**:-

- The input to the non-inverting terminal of the op-amp is the voltage across the 100Ω resistor.

- Using Ohm's Law:

$$Vin = I\times R = 1A\times 100\Omega = 100V$$

However, if the op-amp is ideal and powered by +24V, the output will be limited by the supply voltage.

❖ **Determine the Output Voltage**:-

- The theoretical output voltage would be:

$$Vout = Gain \times Vin = 15.24\times 100V = 1524V$$

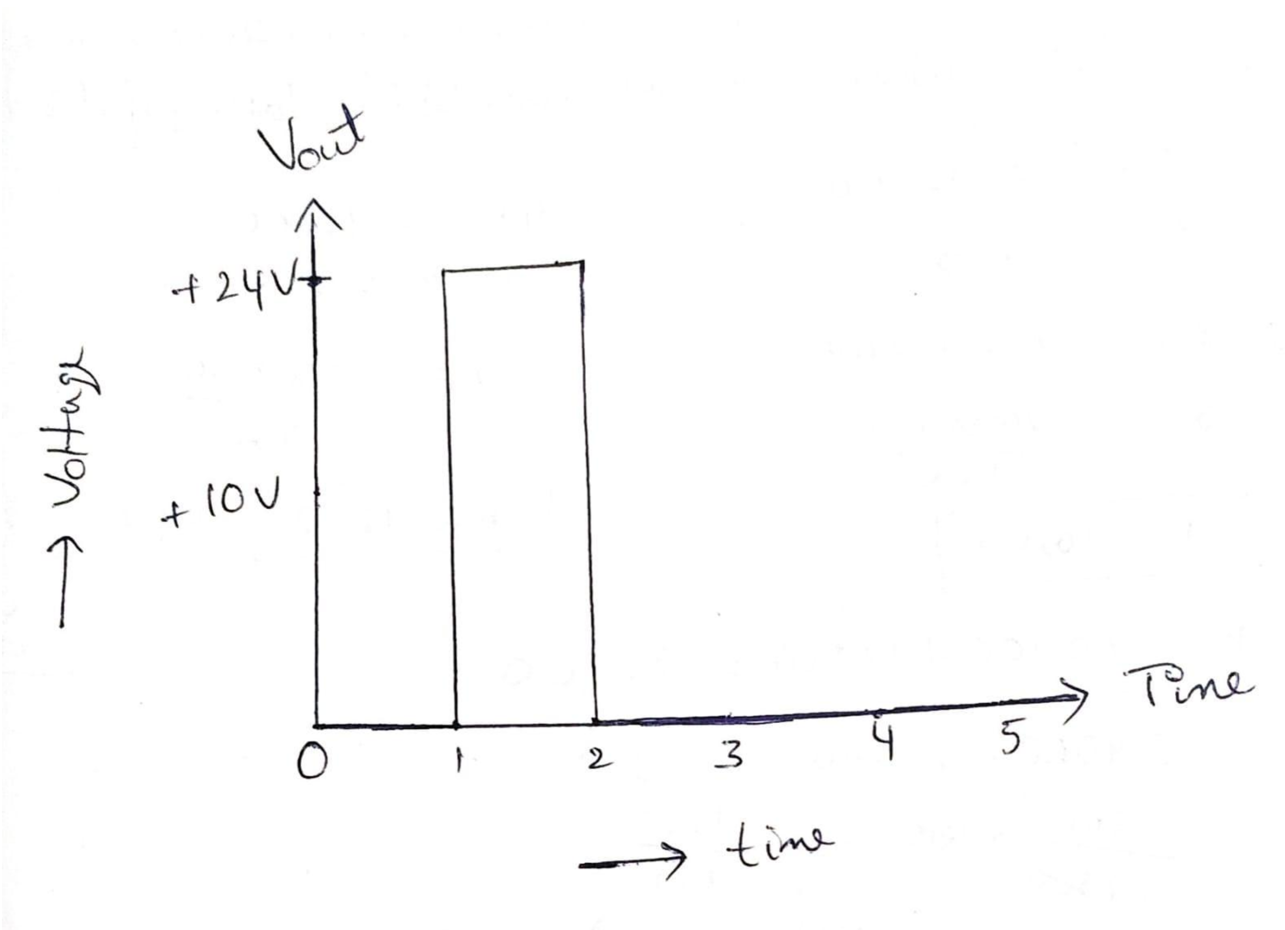
But since the op-amp is powered by +24V, the output will be clipped to +24V.

❖ **Plot the Graph**:-

- From t=0 to t=1 second, the input current is 0, so the output voltage will be 0V.
- From t=1 to t=2 seconds, the input current is 1A, so the output will rise to +24V due to the supply limit.
- After t=2 seconds, the input current drops back to 0A, and the output will return to 0V.

❖ Final Output Graph:-

- The output voltage remains 0V from 0 to 1 second.
- From 1 to 2 seconds, the output voltage is +24V.
- After 2 seconds, the output voltage drops back to 0V.



2. Consider an input voltage waveform (Vin) as given in the graph. Also consider Vbe to be 0.6 V and Zener Voltage of the diode as 10V. Plot Output Voltage vs Time. (Refer Fig: 2.0)

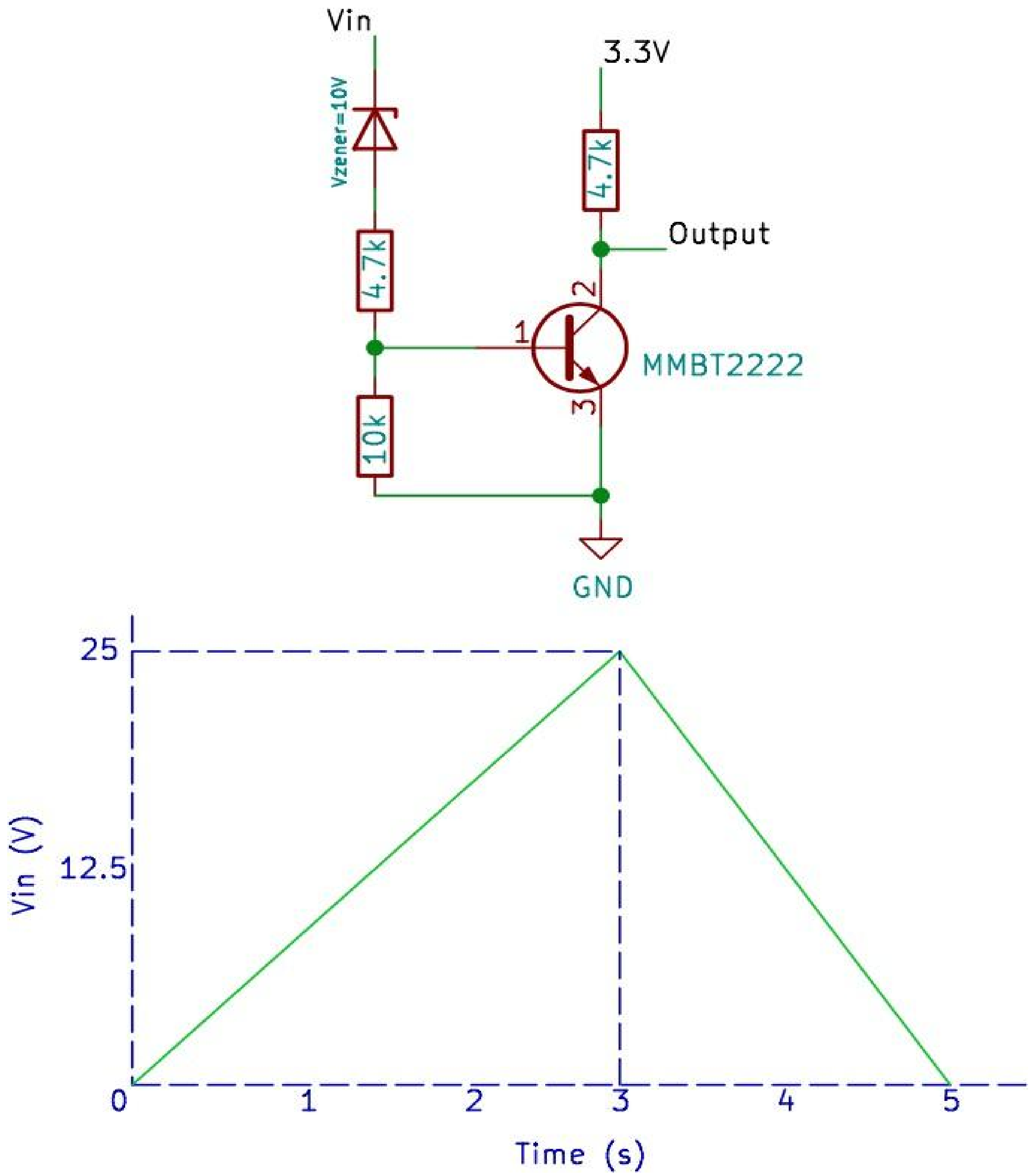


Fig: 2.0

Solution:-

Given:-

- Resistor Values:-

  - Base Resistor RB = 4.7kΩ
  - Collector Resistor RC = 4.7kΩ
  - Zener Resistor RZ = 10kΩ

- **Transistor**: NPN transistor (MMBT2222)
- **Zener Diode Voltage**: VZ = 10V
- **Input Voltage Waveform**: As shown in the graph (triangular waveform peaking at 25V over 5 seconds)
- VBE :- 0.6V
- **Power Supply**: 3.3V at the collector
- **Zener Voltage**: 10V

Steps to Solve:-

❖ **Zener Diode Behavior**:

- The Zener diode will clamp the base voltage to 10V when the input voltage exceeds 10V.

- If the input voltage Vin is less than 10V, the Zener diode will not conduct, and the base voltage will follow the input voltage.

❖ **Base Voltage VB**:-

- When Vin ≤ 10V, VB =Vin
- When Vin > 10V, VB, VB =10V (due to the Zener diode)

❖ **Emitter Voltage VE:-**

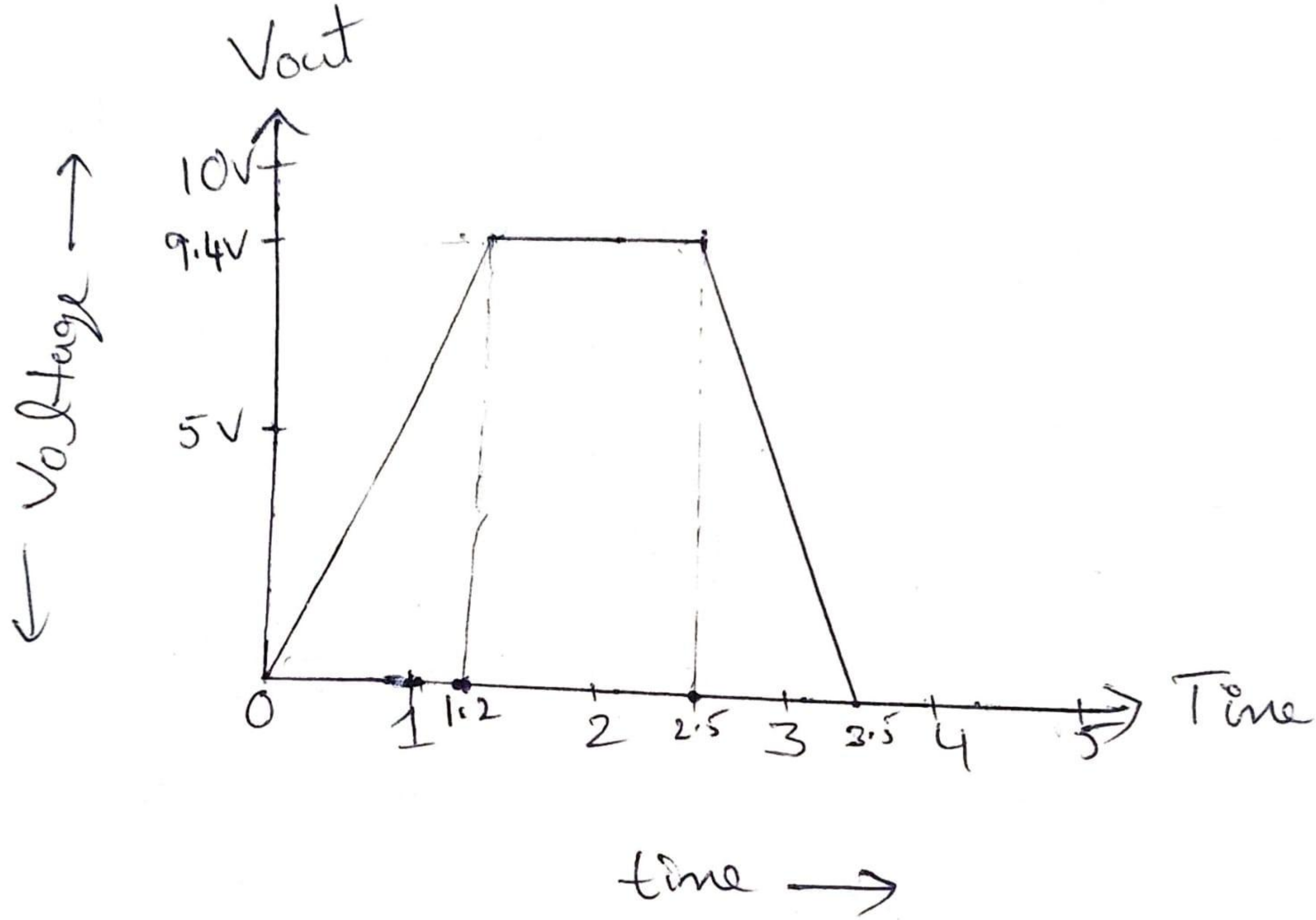
- VE = VB − VBE
- When Vin ≤ 10V, VE = Vin − 0.6V
- When Vin >10V, VE=10V − 0.6V = 9.4V.

❖ **Collector Current IC:-**

- The current through the collector resistor is determined by the voltage difference between the supply voltage (3.3V) and the collector voltage.
- When the transistor is conducting, the voltage at the collector is VC = 3.3V − IC × RC.
- The output voltage at the emitter is also affected by the collector current.

❖ **Output Voltage Plot:-**

- **0 to 1.2 seconds**: The output voltage rises linearly from 0V to approximately 9.4V.
- **1.2 to 2.5 seconds**: The output voltage stays constant at 9.4V (Zener clamping).
- **2.5 to 3.5 seconds**: The output voltage linearly drops from 9.4V to 0V as the input voltage decreases.



Finally, This results in a trapezoidal waveform for the output voltage.

3.Find the output voltage (Refer Fig: 3.0) (Consider β=100, Vbe=0.7, Vce=0.3 Volts)

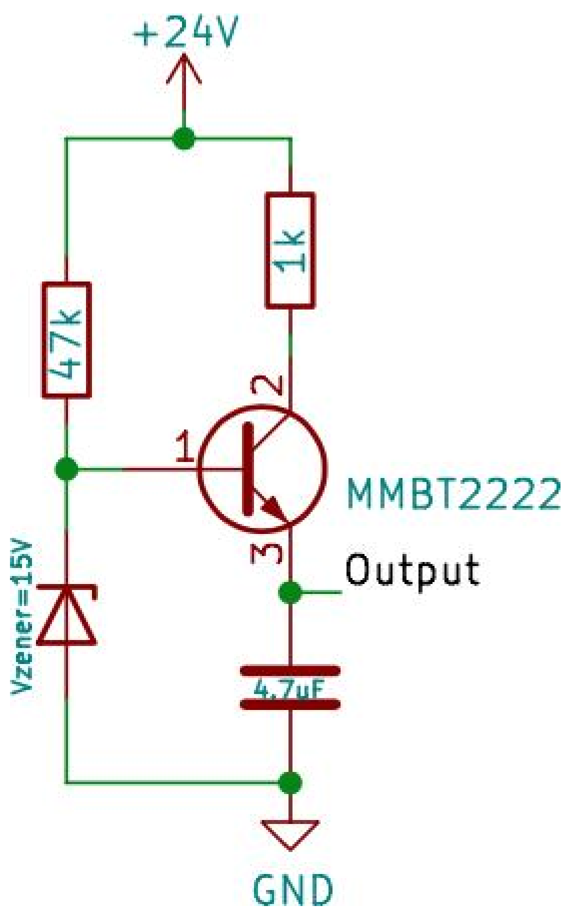


Fig: 3.0

Solution:-

Given:-

- **Transistor**: NPN transistor (MMBT2222).
- **Resistors**:
    - 1kΩ in the collector.
    - 47kΩ in the base.
- **Capacitor**: 4.7μF (4.7x10^-6 F) between the collector and emitter.
- **Zener Diode**: Zener voltage of 15V.
- **Power Supply**: +24V.

We are tasked with finding the output voltage at the emitter.

Steps to Solve:-

❖ **Base Voltage Calculation:-**

- The Zener diode clamps the base voltage VB to 15V.

❖ **Emitter Voltage Calculation**:-

- The emitter voltage VE is related to the base-emitter junction as:

$$VE = VB - VBE$$

- Given VBE=0.7V, the emitter voltage VE is:

$$VE=15V-0.7V=14.3V$$

❖ Capacitor Behavior:-

- The capacitor (4.7µF) in the collector-emitter path can affect the transient response of the circuit but does not impact the steady-state DC voltage across the emitter. Its primary role would be in filtering or timing functions, but it doesn't directly alter the steady-state emitter voltage.

❖ Steady-State Output Voltage at the Emitter:-

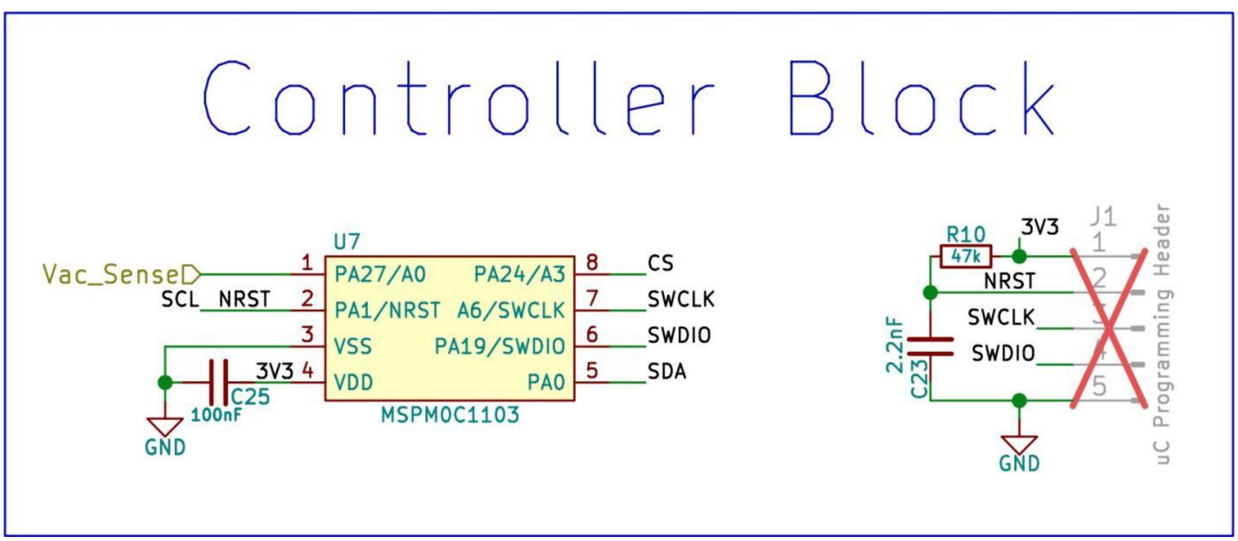
- In steady-state, the capacitor acts as an open circuit for DC analysis. Therefore, the output voltage at the emitter remains determined by the base-emitter junction and Zener diode clamping.
- Thus, the output voltage at the emitter is 14.3V.

❖ Final Answer:-

The output voltage at the emitter VE is **14.3V** in steady-state conditions.

PART B- EMBEDDED QUESTION

1. This is the microcontroller section of a BLDC ceiling fan which is operated using a fan regulator. Make an embedded program to control the fan assuming the following configuration:-
   - The internal ADC reference voltage is set to 2.5V.
   - CS is the current sense signal which corresponds to 69mV per Watt of power from the mains supply. This signal is used for closed loop running of the fan at different power levels.
   - I2C communication is used to communicate with the fan motor driver DRV10983.
   - Speed setting is done based on Vac_Sense signal where 265 Vac corresponds to 2.5V (100% ADC value). 30Vac corresponds to 0 speed and 200Vac corresponds to full speed. Between 30Vac and 200Vac, the fan speed increases linearly.
   - You can make assumptions for any missing information. Clearly mention all the assumptions.



Solution:-

- **Objective:-** Write an embedded program for a BLDC ceiling fan that uses a fan regulator to control speed. Use the provided configurations.

- **Configuration:-**

- The internal ADC reference voltage is set to 2.5V.

- **CS**:- Current sense signal corresponding to 69mV per Watt of power from the mains supply. This is used for closed-loop running of the fan at different power levels.
- **I2C Communication**: -To communicate with the fan motor driver DRV10983.
- **Speed Sensing**:- Based on Vac_Sense, where:

    - 265 Vac corresponds to 2.5V (100% ADC value).
    - 30 Vac corresponds to 0 speed.
    - 200 Vac corresponds to full speed.
    - Speed increases linearly between 30Vac and 200Vac.

- **Micro-controller:-**The microcontroller used is MSP430 (according to the block diagram).

- **Tasks**:-

- Create an I2C communication setup to interact with the DRV10983 motor driver.
- Implement ADC conversion for the Vac_Sense signal to control the fan speed.
- Implement closed-loop control using the CS signal for power regulation.

Embedded Code Outline:-

Here is an outline of the embedded C code that could be used for this application:-

C:-

```
#include <msp430.h>
// Constants
#define ADC_REF_VOLTAGE 2.5
#define MAX_VAC 265
#define MIN_VAC 30
#define FULL_SPEED_ADC_VALUE 2.5
#define ZERO_SPEED_ADC_VALUE 0

// I2C communication setup
#define I2C_SLAVE_ADDRESS 0x58 // Example I2C address for DRV10983

// Function Prototypes
void setup_ADC();
void setup_I2C();
float read_Vac_Sense();
void set_fan_speed(float speed);
void closed_loop_control(float power);

int main(void) {
```

```
   WDTCTL = WDTPW | WDTHOLD;  // Stop watchdog timer

   setup_ADC();
   setup_I2C();

   while (1) {
      // Read Vac_Sense signal (Voltage corresponding to mains input)
      float vac_sense_voltage = read_Vac_Sense();

      // Calculate the desired fan speed based on Vac_Sense
      float speed = (vac_sense_voltage - ZERO_SPEED_ADC_VALUE) / (FULL_SPEED_ADC_VALUE - ZERO_SPEED_ADC_VALUE);

      // Clamp the speed between 0 and 1
      if (speed > 1) speed = 1;
      if (speed < 0) speed = 0;

      // Set fan speed
      set_fan_speed(speed);

      // Read the current power consumption
      float power = /* Read from current sense circuit */;
```

```
        // Perform closed-loop control for power regulation
        closed_loop_control(power);
    }
}

void setup_ADC() {
    // Setup code for ADC to read Vac_Sense
    ADCCTL0 |= ADCSHT_2 | ADCON;        // ADC ON, sampling time
    ADCCTL1 |= ADCSHP;                  // ADC sample-and-hold pulse mode
    ADCCTL2 |= ADCRES;              // 10-bit resolution
    ADCMCTL0 |= ADCSREF_1 | ADCINCH_0;  // Reference = 2.5V, input = Vac_Sense channel
    ADCIE |= ADCIE0;                // Enable ADC interrupt
}

void setup_I2C() {
    // Setup I2C communication with DRV10983
    UCB0CTLW0 |= UCSWRST;           // Put eUSCI_B in reset
    UCB0CTLW0 |= UCMODE_3 | UCSYNC; // I2C mode, synchronous mode
    UCB0BRW = 10;                   // Set baud rate
    UCB0I2CSA = I2C_SLAVE_ADDRESS;  // Slave address
```

```
    UCB0CTLW0 &= ~UCSWRST;            // Release eUSCI_B
for operation
}

float read_Vac_Sense() {
    // Start ADC conversion
    ADCCTL0 |= ADCENC | ADCSC;

    // Wait for conversion to complete
    while (ADCCTL1 & ADCBUSY);

    // Read ADC result
    return ADCMEM0 * (ADC_REF_VOLTAGE / 1023.0);  //
Convert to voltage
}

void set_fan_speed(float speed) {
    // Send speed control data to DRV10983 via I2C
    unsigned char speed_command = (unsigned char)(speed *
255);  // Convert speed (0 to 1) to 8-bit value (0 to 255)
    UCB0TXBUF = speed_command;
}

void closed_loop_control(float power) {
    // Implement closed-loop control logic based on the CS
```

```
(current sense) signal
    float desired_power = /* Calculate based on desired speed */;

    if (power > desired_power) {
        // Reduce speed
        set_fan_speed(/* Reduced speed value */);
    } else if (power < desired_power) {
        // Increase speed
        set_fan_speed(/* Increased speed value */);
    }
}
```

❖ **Key Assumptions:-**

- The speed control is linear between 30 Vac and 200 Vac.
- The closed-loop control adjusts the fan speed based on the power consumption signal from the current sense (CS).
- The microcontroller uses ADC to read the Vac_Sense signal and I2C to communicate with the DRV10983 motor driver.

❖ **Additional Considerations:-**

- Detailed tuning of the closed-loop control algorithm may be needed depending on the actual power characteristics of the fan.
- Interrupt handling and more detailed error checking might be necessary for robust operation.

This code outline assumes a basic understanding of MSP430 programming, ADC setup, and I2C communication protocols.