INDEX

S.NO	DATE	EXPERIMENT TITLE	MARKS	SIGN
1.		Field Effect Transistors		
2.		Single Electron Transistors		
3a.		Resonant Tunneling Devices		
3b.		Quantum Tunneling Devices		

Ex. No: 1 **Field Effect Transistors**
Date:

Aim:

To simulate Field Effect Transistor (FET) characteristic using MATLAB.

Software used:

MathWorks Version (R2024b) of the MATLAB and Simulink product.

Procedure:

1. Open MATLAB.
2. Open new M-file.
3. Type the program.
4. Save in current directory.
5. Compile and Run the program.
6. For the output see command window / Figure window.

Theory:

A Field Effect Transistor (FET) is a three-terminal semiconductor device. Its operation is based on a controlled input voltage. By appearance JFET and bipolar transistors are very similar. However, BJT is a current controlled device and JFET is controlled by input voltage. Most commonly two types of FETs are available.

- **Junction Field Effect Transistor (JFET)**
- **Metal Oxide Semiconductor FET (IGFET)**

Junction Field Effect Transistor

The functioning of Junction Field Effect Transistor depends upon the flow of majority carriers (electrons or holes) only. Basically, JFETs consist of an N type or P type silicon bar containing PN junctions at the sides. Following are some important points to remember about FET –

Gate − By using diffusion or alloying technique, both sides of N type bar are heavily doped to create PN junction. These doped regions are called gate (G).

Source − It is the entry point for majority carriers through which they enter into the semiconductor bar.

Drain − It is the exit point for majority carriers through which they leave the semiconductor bar.

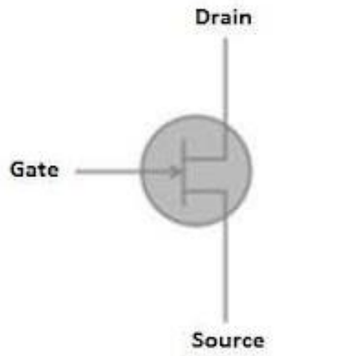
Channel − It is the area of N type material through which majority carriers pass from the source to drain. There are two types of JFETs commonly used in the field semiconductor devices: N-Channel JFET and P-Channel JFET.

N-Channel JFET

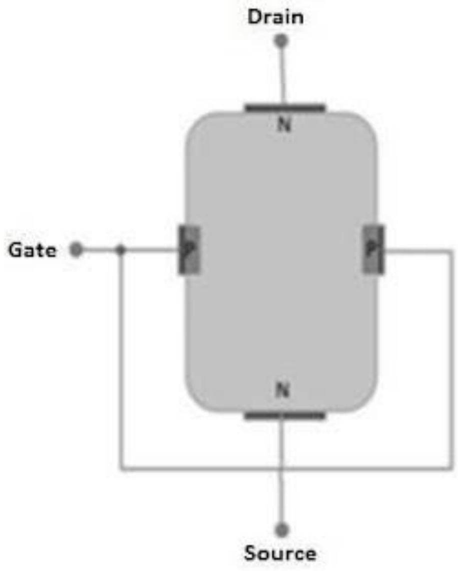
It has a thin layer of N type material formed on P type substrate. Following figure shows the crystal structure and schematic symbol of an N-channel JFET. Then the gate is formed on top of the N channel with P type material. At the end of the channel and the gate, lead wires are attached and the substrate has no connection.

When a DC voltage source is connected to the source and the drain leads of a JFET, maximum current will flow through the channel. The same amount of current will flow from the source and the drain terminals. The amount of channel current flow will be determined by the value of V DD and the internal resistance of the channel.

A typical value of source-drain resistance of a JFET is quite a few hundred ohms. It is clear that even when the gate is open full current conduction will take place in the channel. Essentially, the amount of bias voltage applied at ID, controls the flow of current carriers passing through the channel of a JFET. With a small change in gate voltage, JFET can be controlled anywhere between full conduction and cutoff state.



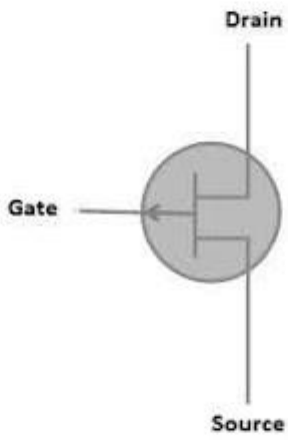
Symbol of N-Channel JFET



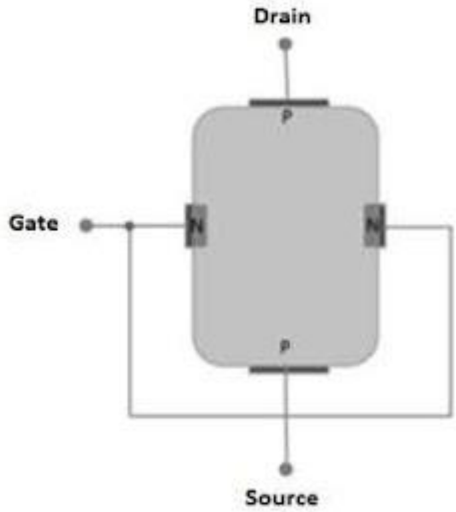
N-Channel JFET

P-Channel JFETs

It has a thin layer of P type material formed on N type substrate. The following figure shows the crystal structure and schematic symbol of an N-channel JFET. The gate is formed on top of the P channel with N type material. At the end of the channel and the gate, lead wires are attached. Rest of the construction details are similar to that of N- channel JFET.



Symbol of P-Channel JFET



P-Channel JFET

Normally for general operation, the gate terminal is made positive with respect to the source terminal. The size of the P-N junction depletion layer depends upon fluctuations in the values of reverse biased gate voltage. With a small change in gate voltage, JFET can be controlled anywhere between full conduction and cutoff state.

Output Characteristics of JFET

The output characteristics of JFET are drawn between drain current (ID) and drain source voltage (VDS) at constant gate source voltage (VGS) as shown in the following figure.

Initially, the drain current (ID) rises rapidly with drain source voltage (VDS) however suddenly becomes constant at a voltage known as pinch-off voltage (VP). Above pinch-off voltage, the channel width becomes so narrow that it allows very small drain current to pass through it. Therefore, drain current (ID) remains constant above pinch-off voltage.

Parameters of JFET

The main parameters of JFET are −

- AC drain resistance (Rd)
- Transconductance
- Amplification factor

AC drain resistance (R_d) − It is the ratio of change in the drain source voltage (ΔV_{DS}) to the change in drain current (ΔI_D) at constant gate-source voltage. It can be expressed as,

$R_d = (\Delta V_{DS})/(\Delta I_D)$ at Constant V_{GS}

Transconductance (g_{fs}) − It is the ratio of change in drain current (ΔI_D) to the change in gate source voltage (ΔV_{GS}) at constant drain-source voltage. It can be expressed as,

$g_{fs} = (\Delta I_D)/(\Delta V_{GS})$ at constant V_{DS}

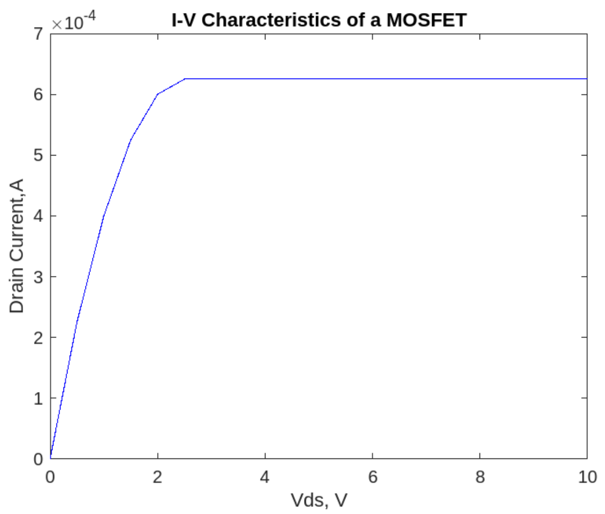
Amplification Factor (u) − It is the ratio of change in drain-source voltage (ΔV_{DS}) to the change in gate source voltage (ΔV_{GS}) constant drain current (ΔI_D). It can be expressed as,

$u = (\Delta V_{DS})/(\Delta V_{GS})$ at constant I_D

Program:

```
clear all;
%kn=un*cox = 100 microA/Volts
kn=1e-4;
%Let W/L= 2
W=360*(10^(-9));
L=180*(10^(-9));
beta=kn*W/L;
%Vth is the threshold voltage
Vth=1;
%Sweep drain to source voltge from 0 to 10V
vds=0:0.5:10;
%Ask the user to enter gate to source voltage
vgs=input('ENTER THE Vgs in volts');
%Estimate length of the array
m=length(vds);
for i=1:m
if vgs < Vth
current(1,i)=0;
elseif vds(i) >= (vgs - Vth)
current(1,i)=0.5* beta * (vgs - Vth)^2;
elseif vds(i) < (vgs - Vth)
current(1,i)= beta*((vgs-Vth)*vds(i) - 0.5*(vds(i)^2));
end
end
plot(vds,current(1,:),'b')
xlabel('Vds, V')
ylabel('Drain Current,A')
title('I-V Characteristics of a MOSFET')
```

Output Waveform:



Inference:

Result:

Ex. No: 2 **Single Electron Transistors**
Date:

Aim:

To simulate Single Electron Transistor (SET) characteristic using MATLAB.

Software used:

MathWorks Version (R2024b) of the MATLAB and Simulink product.

Procedure:

1. Open MATLAB.
2. Open new M-file.
3. Type the program.
4. Save in current directory.
5. Compile and Run the program.
6. For the output see command window / Figure window.

Theory:

Solid state quantum effect devices exploit the benefits of quantum effects. An essential common feature of all these devices is a small island confining the conductive charge in the form of electrons. This island is analogous to the channel of a MOSFET. These devices based on electron confinement are bifurcated into two categories: Single electron and Quantum dots. The composition and device dimensions of islands give the device its distinctive properties. Strategically controlling these parameters allows the device designer to flexibly employ quantum effects in a variety of styles to manage passage of electrons on to and off to the islands.

The Single Electron Transistors (SET) is a unique type of switching device that follows the phenomenon of regulated electron tunneling in order to amplify the current. A nano device can transfers a single electron at a time. The transfer of electrons is governed by Coulomb interaction and occurs on a minute conductive layer termed as island. SETs are considered to be the elements of the future. SETs will be used to produce highly dense and low powered integrated circuits, which will be able to detect the motion of individual electrons. During the discovery of single electronic tunneling and coulomb blockade mechanism, many researchers predicted that on shrinking the dimensions of quantum dots to nanometer range it is quite possible to manufacture applicable SETs. SET is a three-terminal Single Electron Device (SED) which offers low power consumption and high Operating speed. In addition, as the size reaches to nano, quantum mechanical effects are came into action, which makes SETs to work more efficiently.

Program:

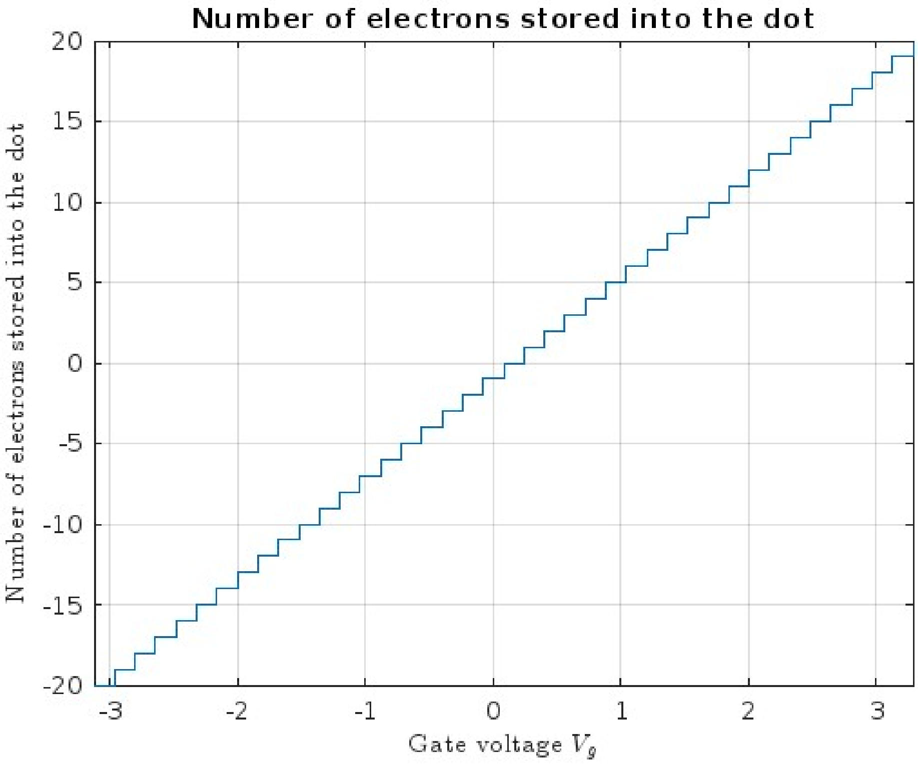
```matlab
% Single Electron Transistor (SET) Simulation
clear; clc; close all;

% Constants
e = 1.6e-19;       % Electron charge (C)
kB = 1.38e-23;     % Boltzmann constant (J/K)
T = 1;             % Temperature (K)
Cg = 1e-18;        % Gate capacitance (F)
Cd = 1e-18;        % Drain capacitance (F)
Cs = 1e-18;        % Source capacitance (F)
Ce = Cg + Cd + Cs; % Total capacitance (F)
Ec = e^2 / (2 * Ce); % Charging energy (J)
kBT = kB * T;       % Thermal energy (J)

% Voltage Ranges
Vg = linspace(-0.5, 0.5, 100); % Gate voltage range (V)
Vds = linspace(-10e-3, 10e-3, 100); % Drain-source voltage range (V)

% Preallocate Current Matrix
I = zeros(length(Vds), length(Vg));

% Function for Fermi-Dirac Distribution
fermi = @(E) 1 ./ (1 + exp(E / kBT));

% Simulation Loop
for i = 1:length(Vg)
    for j = 1:length(Vds)
        % Energy Difference for Electron Tunneling
        DeltaE = Ec * (2 * round((Cg * Vg(i) + Cd * Vds(j)) / (e * Ce)) - ...
                      (Cg * Vg(i) + Cd * Vds(j)) / e);

        % Current via Tunneling Probability
        I(j, i) = e * (fermi(DeltaE) - fermi(-DeltaE));
    end
end

% Plot Results: Coulomb Oscillations
figure;
imagesc(Vg, Vds * 1e3, I); % Plot Current as a Color Map
colorbar;
title('Current Characteristics of SET');
xlabel('Gate Voltage V_g (V)');
ylabel('Drain-Source Voltage V_{DS} (mV)');
set(gca, 'YDir', 'normal');

% Coulomb Oscillation Curve
figure;
plot(Vg, I(round(length(Vds) / 2), :), 'LineWidth', 2);
grid on;
title('Coulomb Oscillations');
xlabel('Gate Voltage V_g (V)');
ylabel('Current I (A)');

% in single electron transistor
```

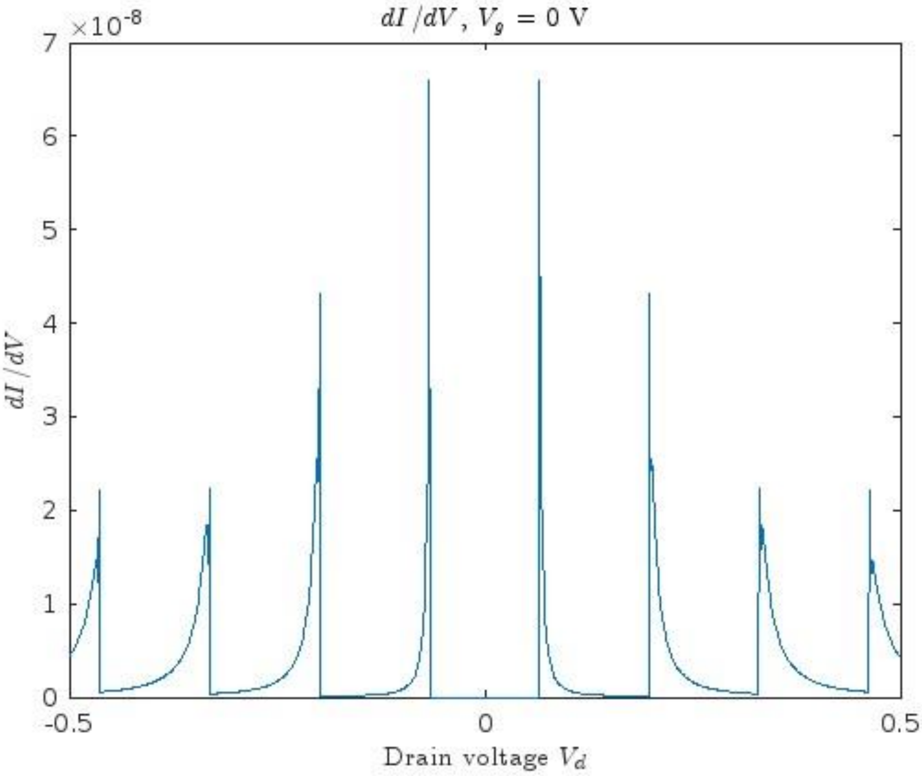
```
%1. first step, the following physical constant and device parameters are defined
as follows.
clc;
clear all;
close all;
% Definition of Physical constant
q=1.602e-19;    % electronic charge (C)
kb=1.381e-23;   % Boltzman constant (J/K)
% Definition of Device parameters
c1=1.0e-20;     % tunnel capacitor C1 (F)
c2=2.1e-19;     % tunnel capacitor C2 (F)
cg=1.0e-18;     % gate capacitor Cg (F)
ctotal=c1+c2+cg;    % total capacitance (F)
mega=1000000;   % definition of mega=106
r1=15*mega;     % tunnel resistance R1 (Ohm)
r2=250*mega;    % tunnel resistance R2 (Ohm)

%2. The values of external parameters (V, V_G, Q_0 and T) is given. Here,
%the V_G , Q_0 and T are kept a constant while the V is varied from Vmin to Vmax,
as follows:
Vg=0;   % gate voltage (V)
q0=0;   % background charge q0 is assumed to be zero
temp=10;    % temperature T (K)
vmin=-0.5;  % drain voltage minimum Vmin (V)
vmax=0.5;   % drain voltage maximum Vmax (V)
NV=1000;    % number of grid from Vmin to Vmax
dV=(vmax-vmin)/NV;  % drain voltage increment of each grid point
for iv=1:NV     % loop start for drain voltage
    Vd(iv)=vmin+iv*dV;   % drain voltage in each grid point
    % Note that loop end for drain voltage is located in the end of this program
source

    %3. Calculation of ?F, as follows:
    Nmin=-20;    % minimum number of N (charge number in dot)
    Nmax=20;     % maximum number of N (charge number in dot)
    for ne=1:Nmax-Nmin  % loop start for N
        n=Nmin+ne;   % N charge number in dot
        %Calculation of deltaE
        dE1p=q/ctotal*(0.5*q+(n*q-q0)-(c2+cg)*Vd(iv)+cg*Vg);
        dE1n=q/ctotal*(0.5*q-(n*q-q0)+(c2+cg)*Vd(iv)-cg*Vg);
        dE2p=q/ctotal*(0.5*q-(n*q-q0)-c1*Vd(iv)-cg*Vg);
        dE2n=q/ctotal*(0.5*q+(n*q-q0)+c1*Vd(iv)+cg*Vg);
        % Noted that loop end for N is located after calculation of \Gamma

        %4. the values of ?F are identified and then used for the calculation of
\Gamma. If ?F is negative,
        % will be calculated by equations (26a) and (26b). However, if the ?F is
positive, \Gamma is set to
        % be closed to the zero (very small). Note that the value of \Gamm ais
always positive. These
        % identifications are done for four conditiond of ?F.

        if dE1p<0
            T1p(ne)=1/(r1*q*q)*(-dE1p)/(1-exp(dE1p/(kb*temp)));
            % ? positive in equation (26a)
        else
            T1p(ne)=1e-1; % ? positive is assumed to be very small
        end
        if dE1n<0
```

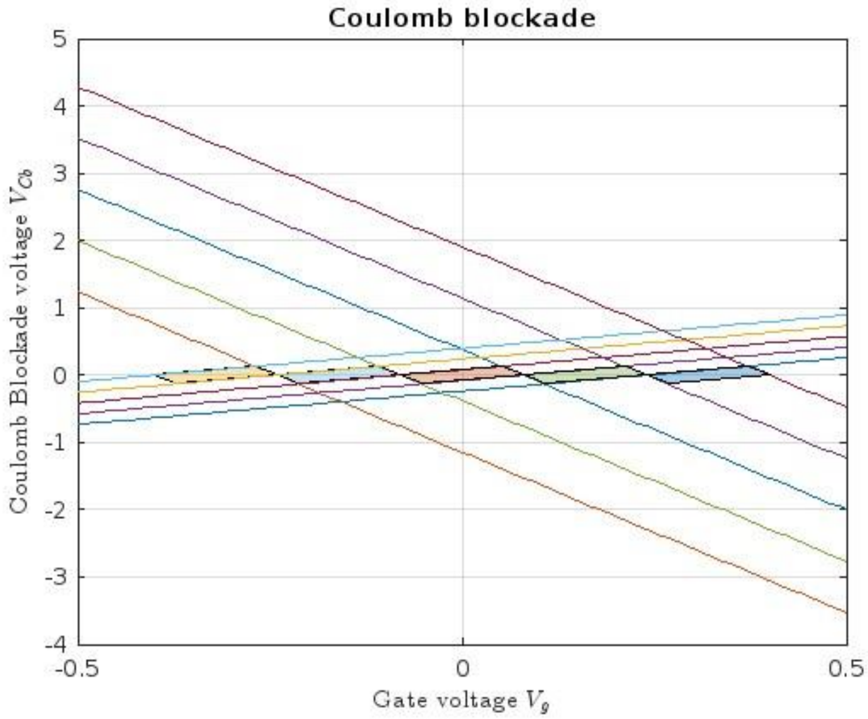
```
            T1n(ne)=1/(r1*q*q)*(-dE1n)/(1-exp(dE1n/(kb*temp)));
            % ? negative in equation (26a)
        else
            T1n(ne)=1e-1; % if negative is assumed to be very small
        end
        if dE2p<0
            T2p(ne)=1/(r2*q*q)*(-dE2p)/(1-exp(dE2p/(kb*temp)));
            %  positive in equation (26b)
        else
            T2p(ne)=1e-1; % if positive is assumed to be very small
        end

        if dE2n<0
            T2n(ne)=1/(r2*q*q)*(-dE2n)/(1-exp(dE2n/(kb*temp)));
            % ? negative in equation (26b)
        else
            T2n(ne)=1e-1; % if negative is assumed to be very small

        end
    end % loop end for N

    % 5. The p(n) of equation (28) is calculated. For this, normalization of
equation (31a) must
    % be satisfied. Here, the values of p(Nmin) and p(Nmax) is assumed to be 0.01.

    p(1)=0.001; % ?(Nmin) is assumed to be 0.01
    p(Nmax-Nmin)=0.001; % ?(Nmax) is assumed to be 0.01
    % 6. Normalization of p is done. Here, ? p(n) is calculated.
    sum=0; % sum=0 is initial value to calculate ?
    for ne=2:Nmax-Nmin
        p(ne)=p(ne-1)*(T2n(ne-1)+T1p(ne-1))/(T2p(ne)+T1n(ne));
        % calculation of ?(N) in equation (28)
        % The conditions below are used to avoid divergence of Matlab  calculation

        if p(ne)>1e250
            p(ne)=1e250;
        end
        if p(ne)<1e-250
            p(ne)=1e-250;
        end
        % ---------------------
        sum=sum+p(ne);
    end
    for ne=2:Nmax-Nmin
        p(ne)=p(ne)/sum; % Normalization in equation (31b)
    end

    %Finally, the current is computed as follows:

    sumI=0; % sumI=0 is initial condition for current calculation

    for ne=2:Nmax-Nmin
        sumI=sumI+p(ne)*(T2p(ne)-T2n(ne));
    end
    I(iv)=q*sumI; % I in equation (32b)
end % end of drain voltage loop
figure('Name','plot of I vs V_d','NumberTitle','off');
plot(Vd,I); % plot of I vs V
xlabel ('Drain voltage $V_d$', 'Interpreter', 'latex');
```

```
ylabel ('Drain current $I_d$', 'Interpreter', 'latex');
title ("Drain current, V_g = " + Vg + " V");
for iv=1:NV-1
    dIdV(iv)=(I(iv+1)-I(iv))/dV; % calculation of dI/dV
end
figure('Name','plot of dI/dV vs V_d','NumberTitle','off');
plot(Vd(1,1:NV-1),dIdV);
xlabel ('Drain voltage $V_d$', 'Interpreter', 'latex');
ylabel ('$dI/dV$', 'Interpreter', 'latex');
title ("$dI/dV$, $V_g$ = " + Vg + " V", 'Interpreter', 'latex');
%Coulomb Blockade Plot : Vd vs Vg
%deltaVd = e/cg;
deltaVcbp = q/(cg + c1); %usato per il fascio di rette con slope positivo da
plottare
deltaVcbn = q/c2; %usato per il fascio di rette con slope negativo da plottare
Vg_min = -0.5;
Vg = (Vg_min : 0.01 : -Vg_min);
for i = 1:length(Vg)
    Vcbp = ((q/2) + Vg *cg)/(cg+c1); %Vcb positive slope
    Vcbn = q/(2*c2) - (Vg*cg)/c2; %Vcb negative slope
end
deltax = q/cg;
deltay = (2*q)/ctotal;
numeroFasciRette = 4;
figure('Name','Coulomb Blockade','NumberTitle','off');
for i = -numeroFasciRette/2 : numeroFasciRette/2
    plot (Vg, Vcbp + i*deltaVcbp);
    hold on;
    plot (Vg, Vcbn + i*deltaVcbn);
    hold on;
    % Identificazione dei rombi
    pgon = polyshape([-q/(2*cg)+i*deltax ((cg+c1)*q/ctotal-q/2)/cg+i*deltax
q/(2*cg)+i*deltax  -(((cg+c1)*q/ctotal-q/2)/cg)+i*deltax],...
    [0  q/ctotal   0  -q/ctotal]);
    plot(pgon);
end
xlabel ('Gate voltage $V_g$', 'Interpreter', 'latex');
%xlim([min(Vg)  max(Vg)]);
ylabel ('Coulomb Blockade voltage $V_{Cb}$', 'Interpreter', 'latex');
%ylim([min(Vcbn+numeroFasciRette/2*deltaVcbn)
max(Vcbp+numeroFasciRette/2*deltaVcbp)]);
title("Coulomb blockade");
grid on;
hold off;
%N vs Vg
clear Vg
for n = Nmin:Nmax
    Vg(n+Nmax+1) = q/cg*(n+1/2);
end
figure('Name','Electrons in dot vs. Vg','NumberTitle','off');
stairs (Vg, Nmin:Nmax);
xlabel ('Gate voltage $V_g$', 'Interpreter', 'latex');
xlim([min(Vg)  max(Vg)]);
ylabel ('Number of electrons stored into the dot', 'Interpreter', 'latex');
ylim([Nmin  Nmax]);
title("Number of electrons stored into the dot");
grid on;
```

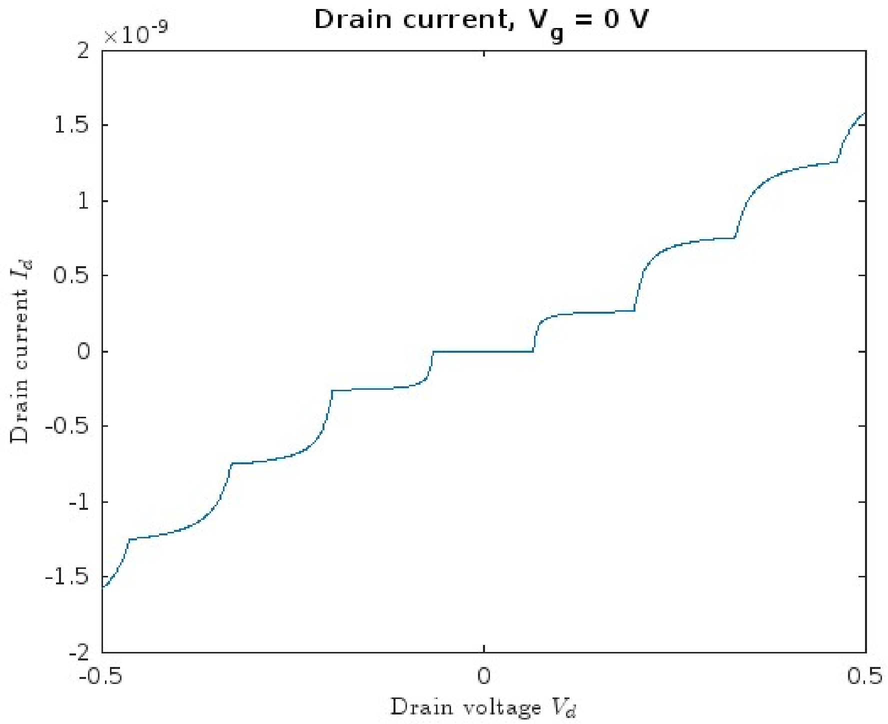
Output Waveform:







Coulomb blockade
Coulomb Blockade voltage V_{Cb}
Gate voltage V_g
5
4
3
2
1
0
-1
-2
-3
-4
-0.5
0
0.5



Drain current, V_g = 0 V
$\times 10^{-9}$
Drain current I_d
Drain voltage V_d
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-0.5
0
0.5

Inference:

Result:

Ex. No: 3a **Resonant Tunneling Devices**

Date:

Aim:

To simulate Resonant Tunneling devices characteristic using MATLAB.

Software used:

MathWorks Version (R2024b) of the MATLAB and Simulink product.

Procedure:

1. Open MATLAB.
2. Open new M-file.
3. Type the program.
4. Save in current directory.
5. Compile and Run the program.
6. For the output see command window / Figure window.

Theory:

Tunnel diodes are one of the most significant solid-state electronic devices which have made their appearance in the last decade. Tunnel diode was invented in 1958 by Leo Esaki. Leo Esaki observed that if a semiconductor diode is heavily doped with impurities, it will exhibit negative resistance. Negative resistance means the current across the tunnel diode decreases when the voltage increases. In 1973 Leo Esaki received the Nobel Prize in physics for discovering the electron tunneling effect used in these diodes.

A tunnel diode is also known as Esaki diode which is named after Leo Esaki for his work on the tunneling effect. The operation of tunnel diode depends on the quantum mechanics principle known as "Tunneling". In electronics, tunneling means a direct flow of electrons across the small depletion region from n-side conduction band into the p-side valence band.

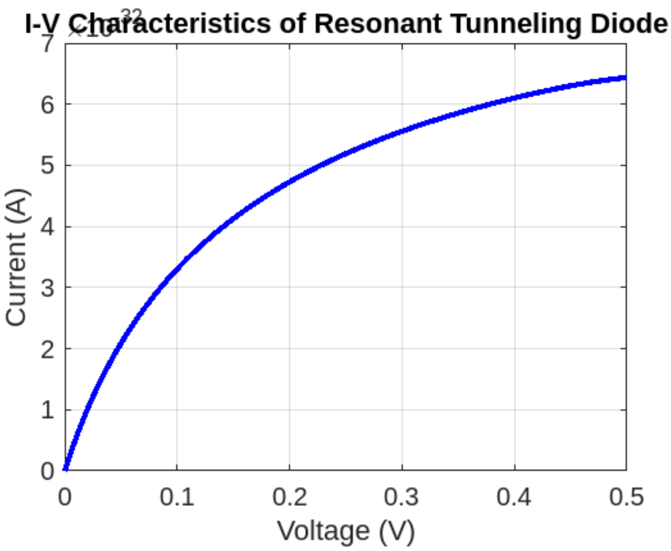
The germanium material is commonly used to make the tunnel diodes. They are also made from other types of materials such as gallium arsenide, gallium antimonide, and silicon.

Program:

```
% Resonant Tunneling Diode (RTD) Simulation
clc;
clear;
% Constants
q = 1.6e-19; % Elementary charge (C)
hbar = 1.054e-34; % Reduced Planck's constant (J.s)
m = 9.11e-31; % Electron mass (kg)
meff = 0.067 * m; % Effective mass of electrons in GaAs (kg)
```

```
kB = 1.38e-23; % Boltzmann constant (J/K)
T = 300; % Temperature (K)
% Parameters
V = linspace(0, 0.5, 500); % Voltage range (V)
E0 = 0.2; % Resonant energy level (eV)
Gamma = 0.01; % Resonant width (eV)
Area = 1e-12; % Device area (m^2)
J0 = 1e3; % Scaling factor for current density (A/m^2)
% Convert eV to Joules
E0 = E0 * q;
Gamma = Gamma * q;
% Fermi distribution function
f = @(E, Ef) 1 ./ (1 + exp((E - Ef) / (kB * T)));
% Calculate current for each voltage
I = zeros(size(V));
for i = 1:length(V)
Vbias = V(i);
% Energy integral for tunneling current
E = linspace(-0.5, 0.5, 1000) * q; % Energy range around the resonance (J)
T_E = (Gamma^2) ./ ((E - E0).^2 + Gamma^2); % Transmission coefficient
F = f(E, 0) - f(E, -q * Vbias); % Fermi difference
I(i) = J0 * Area * trapz(E, T_E .* F); % Integrate over energy
end
% Plot the results
figure;
plot(V, I, 'b', 'LineWidth', 2);
grid on;
title('I-V Characteristics of Resonant Tunneling Diode');
xlabel('Voltage (V)');
ylabel('Current (A)');
```

Output Waveform:



Inference:

Result:

Tunneling Devices was designed using MATLAB and the output waveform was viewed using Figure window.

Ex. No: 3b **Quantum Tunneling Devices**
Date:

Aim:

To simulate Quantum Tunneling devices characteristic using MATLAB.

Software used:

MathWorks Version (R2024b) of the MATLAB and Simulink product.

Procedure:

1. Open MATLAB.
2. Open new M-file.
3. Type the program.
4. Save in current directory.
5. Compile and Run the program.
6. For the output see command window / Figure window.

Theory:

Tunneling devices exploit the quantum mechanical phenomenon of tunneling, where particles can pass through potential barriers that would be insurmountable according to classical physics. Tunneling is a direct consequence of the wave nature of particles and occurs when a particle's wavefunction extends beyond a barrier, allowing it to appear on the other side with a reduced probability.

When a particle encounters a potential barrier higher than its energy, it cannot pass classically. In quantum mechanics, the wavefunction describing the particle decays exponentially within the barrier but does not vanish entirely, giving a finite probability of transmission.

Tunneling Devices :

1. Resonant Tunneling Diodes (RTDs):

- Use double-barrier quantum wells.
- Electrons tunnel resonantly through discrete energy levels in the quantum well.
- Exhibit negative differential resistance (NDR): current decreases with increasing voltage in certain regions.

2. Tunnel Diodes:

- Have heavily doped (p-n) junctions, creating a very narrow depletion region.
- Tunneling dominates at low forward bias, leading to NDR.
- Applications: oscillators, high-speed switches.

3. Single Electron Transistors (SETs):

- Utilize tunneling through small islands (quantum dots).
- Allow control of single electrons using the Coulomb blockade effect.
- Applications: nanoscale sensors, quantum computing.

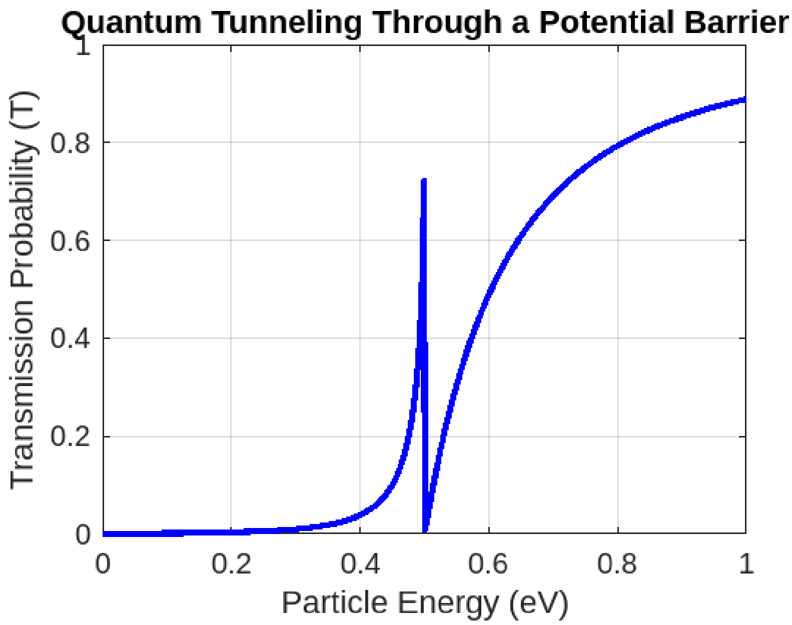
4. Josephson Junctions:

- Use tunneling of Cooper pairs (paired electrons) in superconductors.
- Exhibit quantum interference effects, used in quantum bits and precision measurements.

Program:

```
% Quantum Tunneling Simulation through a Potential Barrier
clc;
clear;

% Constants
hbar = 1.054e-34;  % Reduced Planck's constant (J·s)
m = 9.11e-31;      % Mass of an electron (kg)
q = 1.6e-19;       % Elementary charge (C)

% Barrier parameters
V0 = 0.5 * q;      % Barrier height in Joules (0.5 eV)
L = 1e-9;          % Barrier width in meters (1 nm)

% Energy range (0 to 2 * V0)
E = linspace(0, 2 * V0, 500); % Energy of the particle (Joules)

% Transmission coefficient calculation
T = zeros(size(E));
for i = 1:length(E)
    if E(i) < V0
        % E < V0: Quantum tunneling occurs
        kappa = sqrt(2 * m * (V0 - E(i))) / hbar; % Decay constant
        T(i) = exp(-2 * kappa * L);              % Transmission probability
    else
        % E >= V0: Classical-like transmission
        k = sqrt(2 * m * (E(i) - V0)) / hbar;    % Wave vector
        T(i) = 1 / (1 + (V0^2 / (4 * E(i) * (E(i) - V0)))); % T using continuity
conditions
    end
end

% Plotting the results
figure;
plot(E / q, T, 'b', 'LineWidth', 2); % Convert energy to eV for x-axis
grid on;
xlabel('Particle Energy (eV)');
ylabel('Transmission Probability (T)');
title('Quantum Tunneling Through a Potential Barrier');
```

Output Waveform:



Inference:

Result: