EXP 8: Rise time budget

clc;

clear all;

close all;

t\_s = input('t\_source:');

t\_d = input('t\_detector:');

t\_f1 = input('intermodal:');

t\_f2 = input('intramodal:');

l = 11

t\_sys = sqrt((t\_s^2) + (t\_d^2) + (t\_f1\*l)^2 +(t\_f2\*l)^2)

NRZ = (0.7/t\_sys)\*10^3

RZ = (0.35/t\_sys)\*10^3

Exp : to plot the refractive index profile of step and graded index fiber

import numpy as np

import matplotlib.pyplot as plt

n1 = 1.5

delta = 0.01

a = 30

n2 = n1 \* np.sqrt(1 - 2 \* delta)

r = np.linspace(0, 50)

n\_step = np.where(r <= a, n1, n2)

plt.figure()

plt.plot(r, n\_step, 'b', label="Step-Index Profile")

plt.axvline(a, color='r', linestyle='--', label="Core Boundary")

plt.xlabel("Radial Distance (μm)")

plt.ylabel("Refractive Index")

plt.title("Step-Index Fiber profile ")

plt.legend()

plt.grid(True)

plt.show()

n1 = 1.5

delta = 0.01

a = 30

n2 = n1 \* np.sqrt(1 - 2 \* delta)

r\_range = np.linspace(0, a, 500)

m = (r\_range / a)

n\_triangular = n1 \* (1 - 2 \* delta \* m) \*\* 0.5

n\_parabolic = n1 \* (1 - 2 \* delta \* m \*\* 2 ) \*\* 0.5

n\_step = np.where(r\_range <= a, n1, n2)

plt.figure()

plt.plot(r\_range, n\_step, label="Step-Index (α = ∞)")

plt.plot(r\_range, n\_triangular, label="Triangular Profile (α = 1)")

plt.plot(r\_range, n\_parabolic, label="Parabolic Profile (α = 2)")

plt.axvline(a, color='m', linestyle='--', label="Core Boundary")

plt.xlabel("Radial Distance (mm)")

plt.ylabel("Refractive Index")

plt.title("Refractive Index Profiles")

plt.legend()

plt.grid(True)

plt.show()

EXP 7: Link Power Budget

pc=int(input('Enter the mean power launched from the LASER transmiter='))

pr=int(input('Enter the mean power required at the APD reciver='))

lsp=float(input('Enter the splice loss='))

af=float(input('Enter the cable fiber loss='))

sm=int(input('Enter the safety margin='))

lc=int(input('Enter the connector loss='))

dp=float(input('Enter the dispersion equalization penalty='))

pt=pc-pr

print('total allowed power loss=',pt)

L1=(pt-sm-2\*lc)/(af+lsp)

print('possible link length for 400mbps=',L1)

L2=(pt-sm-2\*lc-dp)/(af+lsp)

print('possible link length for 400mbps with dispersion equalization penalty of 1.5 =',L2)

r=L1-L2

print('reduction in length is ',r)