Implementation of Analog and Digital Circuits using MatLab and Interfacing Keithley 2450 with LabVIEW

Short Review on Control System:

- PI Controllers:

A proportional integral controller is a feedback control loop that calculates an error signal by taking the difference between the output of a system. A PI Controller could improve relative stability and eliminate steady state error at the same time, but the settling time is increased.

- PID Controllers:

PID (proportional integral derivative) controllers use a control loop feedback mechanism to control process variables and are the most accurate and stable controller.

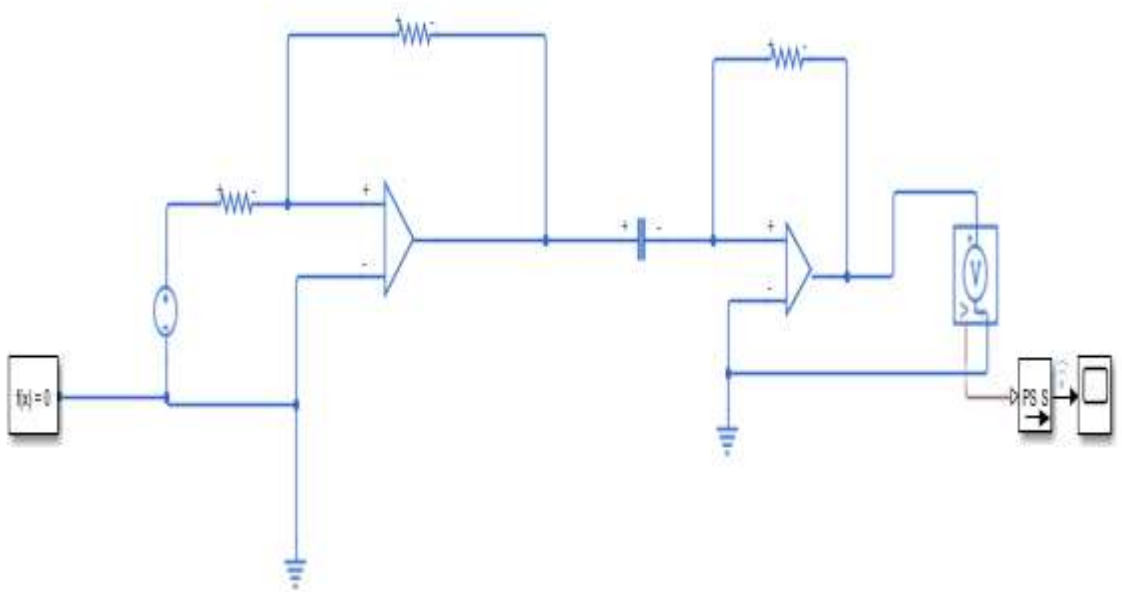
- PD Controllers:

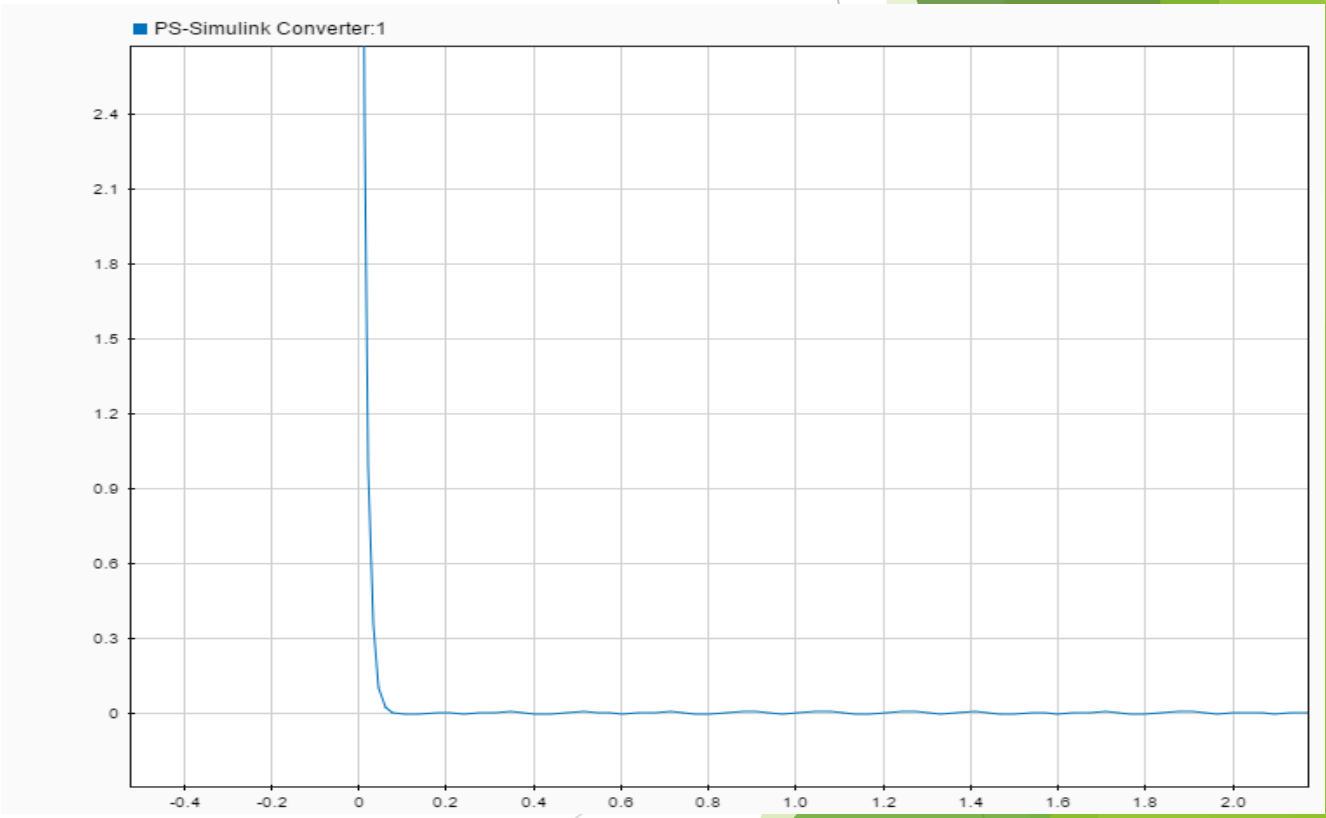
A proportional derivative controller (PD controller), as its name indicates, is a controller with a “proportional action” and a “derivative action”.

Implementation of PD,PID controllers using Matlab:
(Following outputs is taken for unit step function):

PD controller:

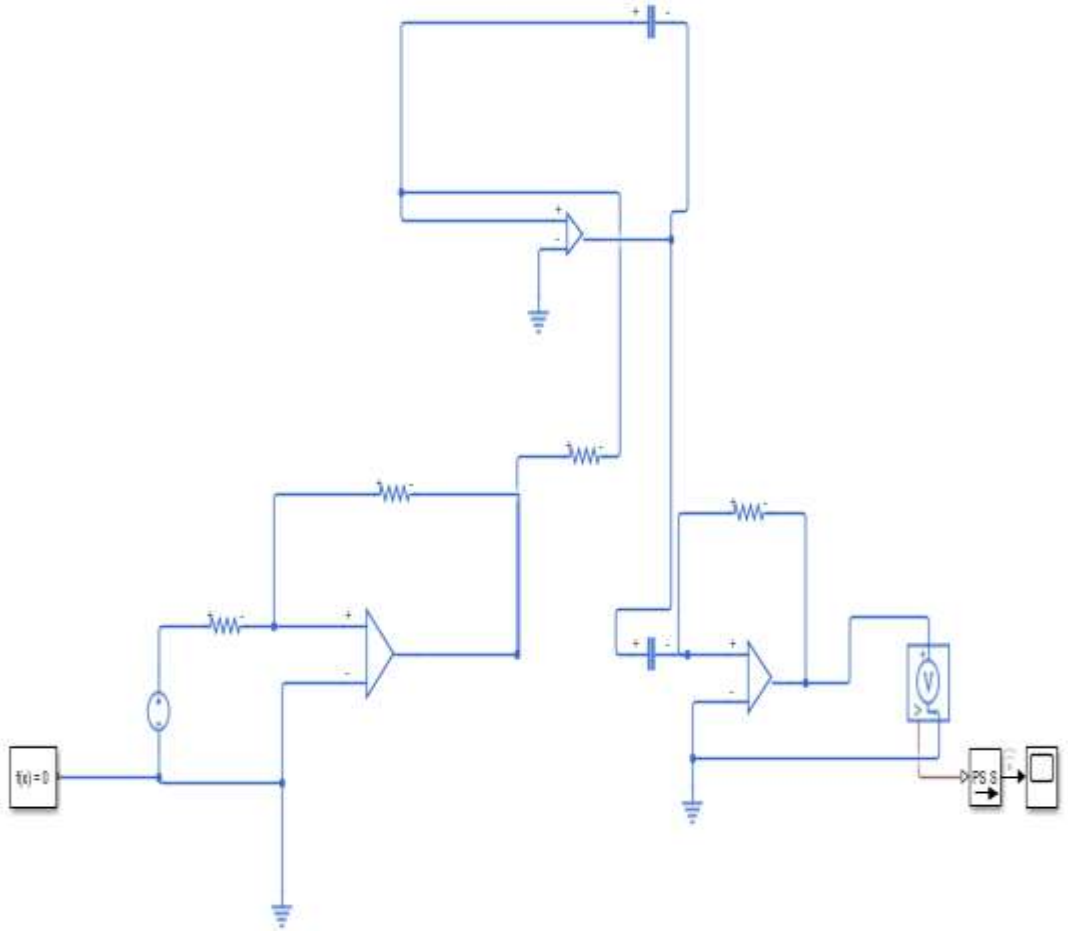
The output is shown below(in form of the graph)as shown by the scope:

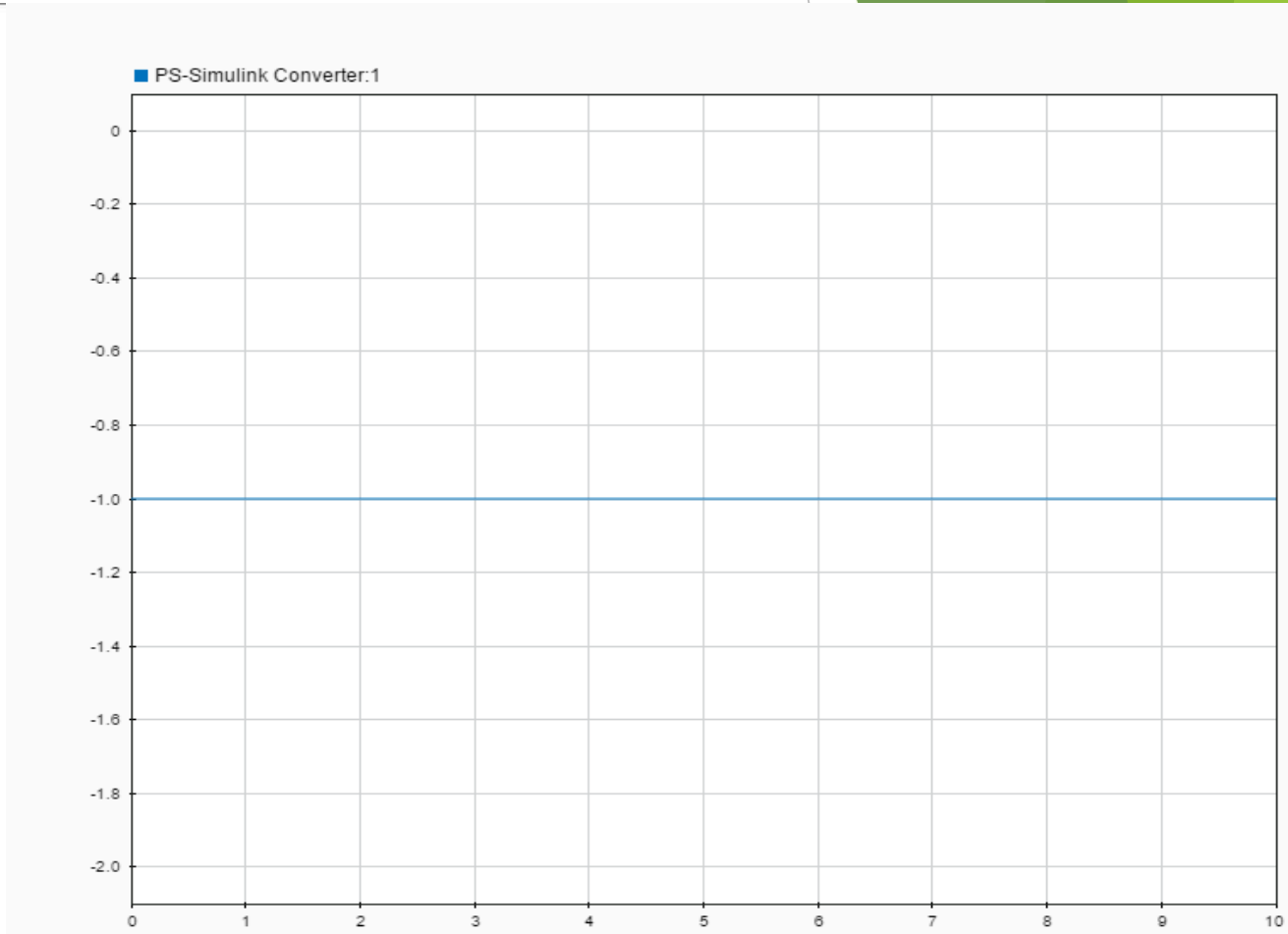




PID Controller:

Here, the output is amplified by a factor of -1(as integrator and differentiator cancel each other)



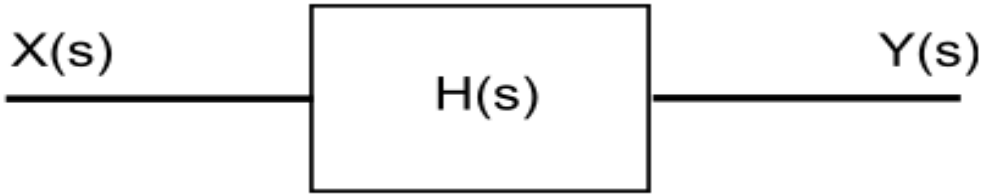


Transfer Function and Signal Flow Diagrams:

1. Transfer function is used in the frequency domain analysis of systems using transform methods such as the Laplace transform which means the amplitude of the output as a function of the frequency of the input signal.
2. To find the transfer function of overall system , the State Flow Graph technique(SFG) is used.
3. The very first thing to find out the transfer function is to convert the time domain system to Laplacian system function which are easy to solve.
4. Transfer function is found out using SFG for two types of systems:

- Open loop system:

The transfer function is H(s)(here),

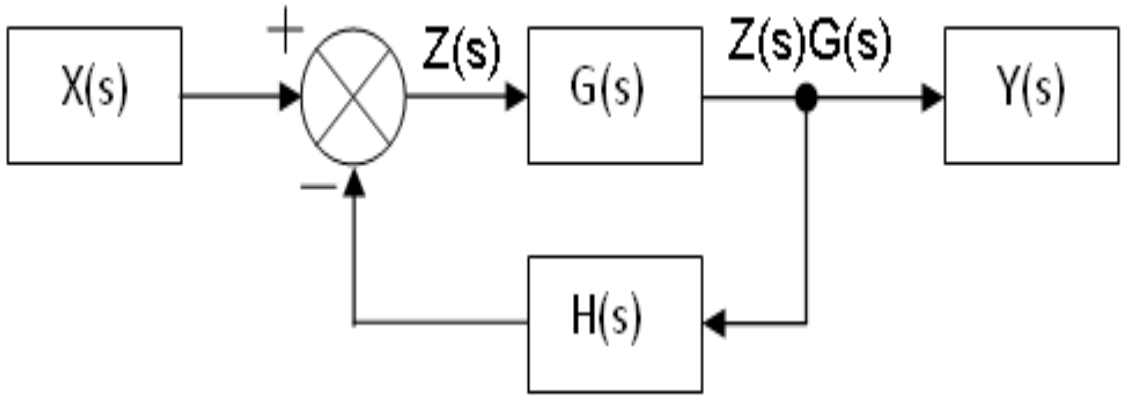


$$H(s) = \frac{Y(s)}{X(s)}$$

- Closed Loop System:

This is a set of mechanical or electronic devices that automatically regulates a process variable to a desired state or set point without human interaction. Closed loop control systems contrast with open loop control systems, which require manual input i.e. this system requires a feedback which is not used in case of open loop system.

An example of a closed-loop transfer function is shown below:



The summing node and the $G(s)$ and $H(s)$ blocks can all be combined into one block, which would have the following transfer function:

$$\frac{Y(s)}{X(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

Mason Gain Formula:

- This formula was used mainly to find the transfer function of closed loop system (An example is shown below):

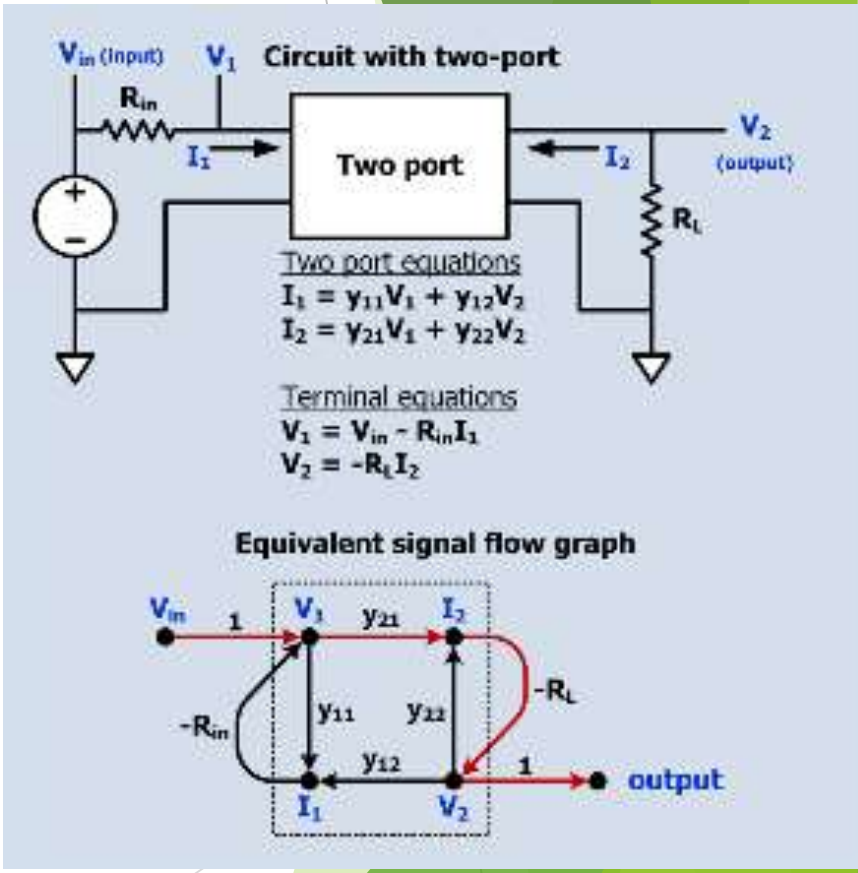
The gain formula is as follows:

$$G = \frac{y_{\text{out}}}{y_{\text{in}}} = \frac{\sum_{k=1}^{N} G_k \Delta_k}{\Delta}$$

$$\Delta = 1 - \sum L_i + \sum L_i L_j - \sum L_i L_j L_k + \cdots + (-1)^m \sum \cdots + \cdots$$

where:

- Δ = the determinant of the graph.
- y_{in} = input-node variable
- y_{out} = output-node variable
- G = complete gain between y_{in} and y_{out}
- N = total number of forward paths between y_{in} and y_{out}
- G_k = path gain of the kth forward path between y_{in} and y_{out}
- L_i = loop gain of each closed loop in the system
- L_iL_j = product of the loop gains of any two non-touching loops (no common nodes)
- $L_iL_jL_k$ = product of the loop gains of any three pairwise nontouching loops
- Δ_k = the cofactor value of Δ for the k^{th} forward path, with the loops touching the k^{th} forward path removed. *

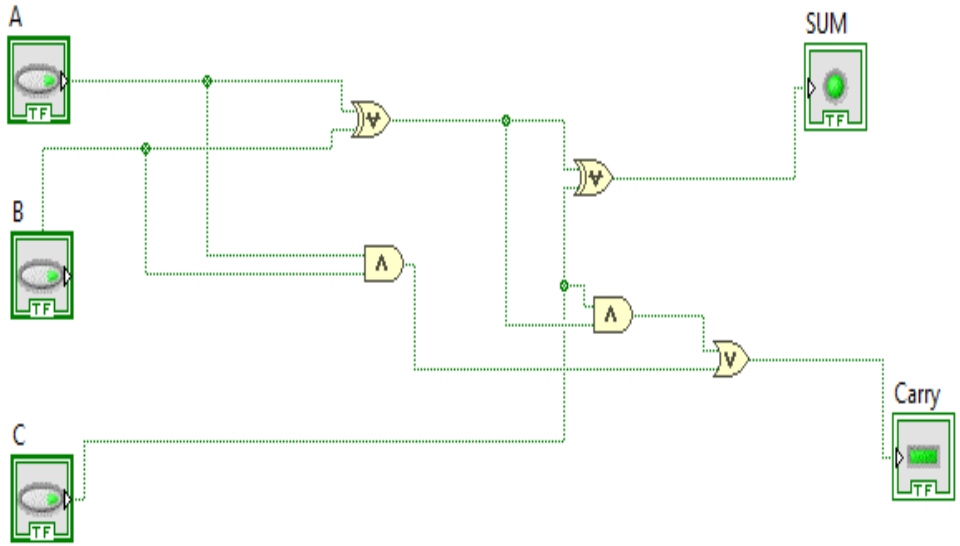


Logic Gates Implementation using LabVIEW:

Following elements were implemented using gates in LabVIEW:

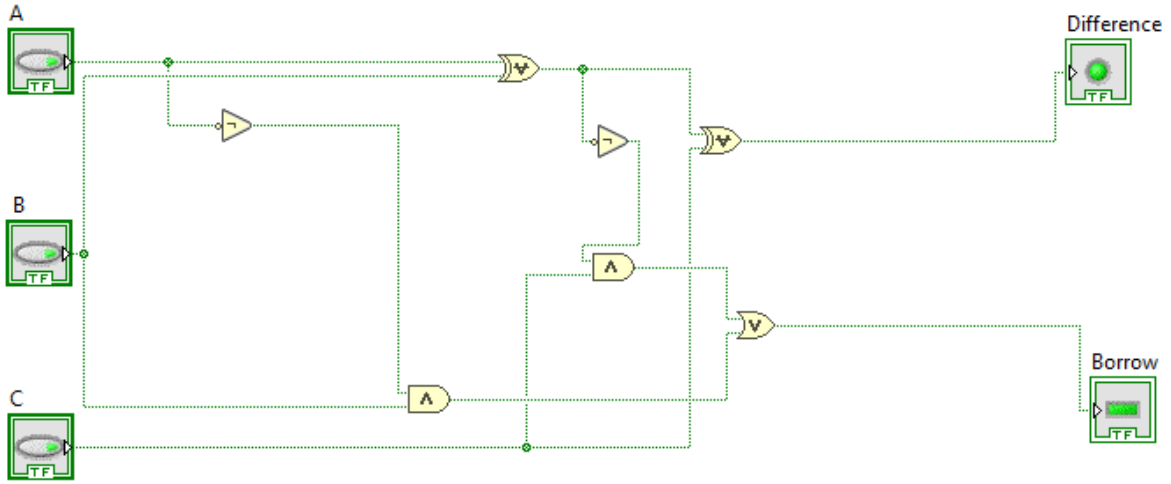
1. Full Adder:

It is implemented using XOR,AND,OR gates



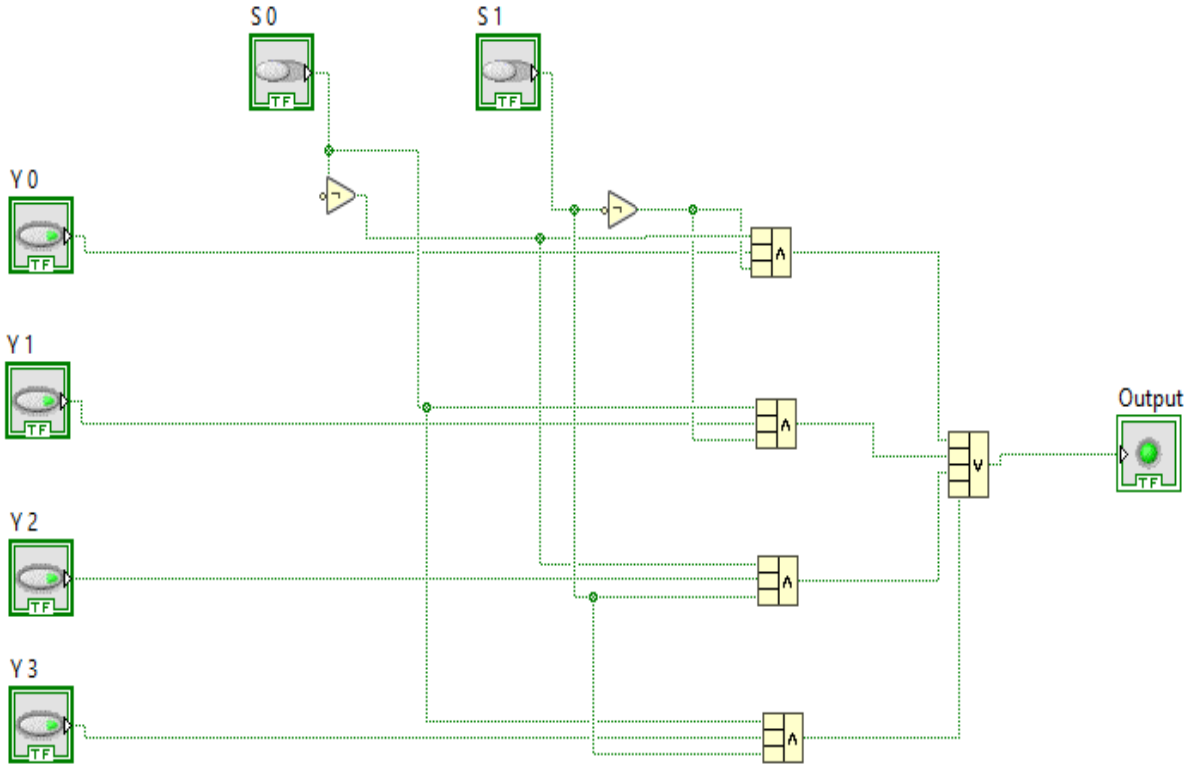
2. Full Subtractor:

It is implemented using XOR,AND,OR and NOT gates



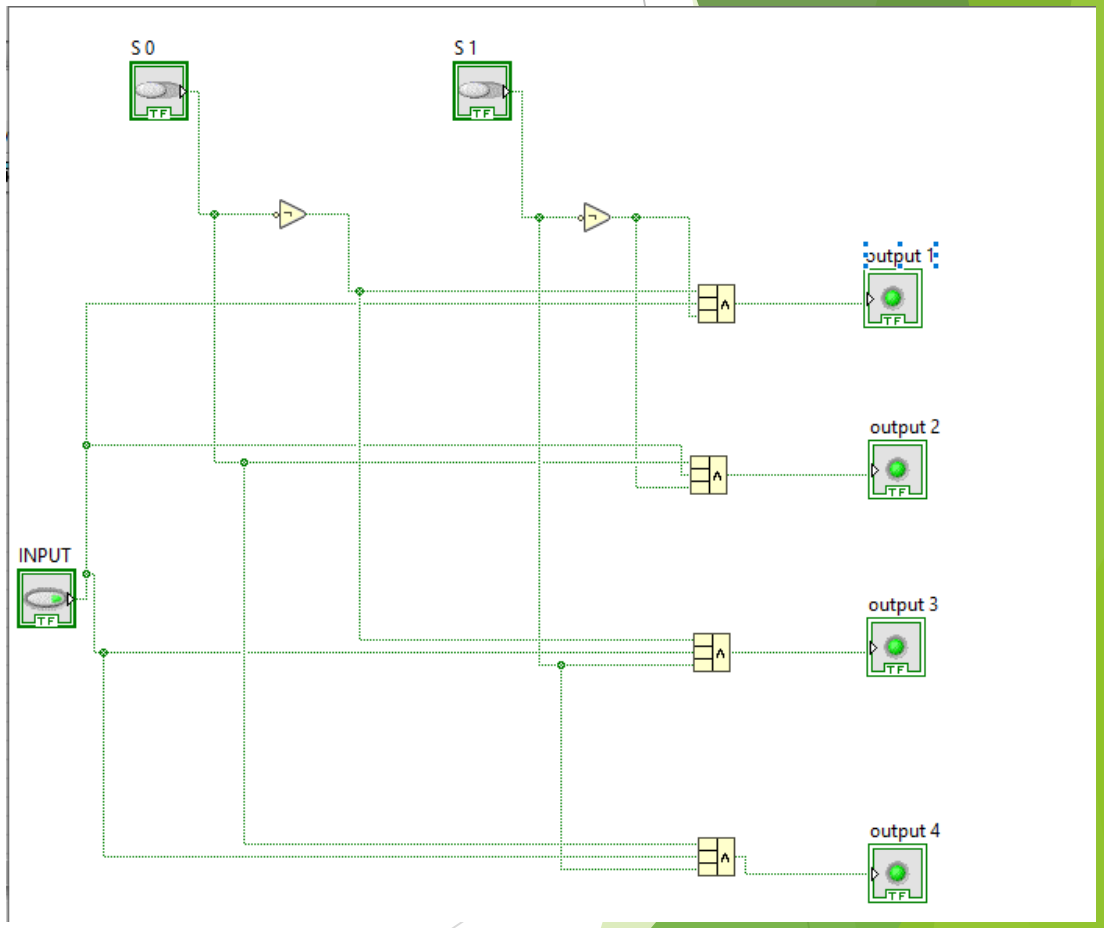
3. Multiplexer:

A 4:2 multiplexer is shown below



4. Demultiplexer:

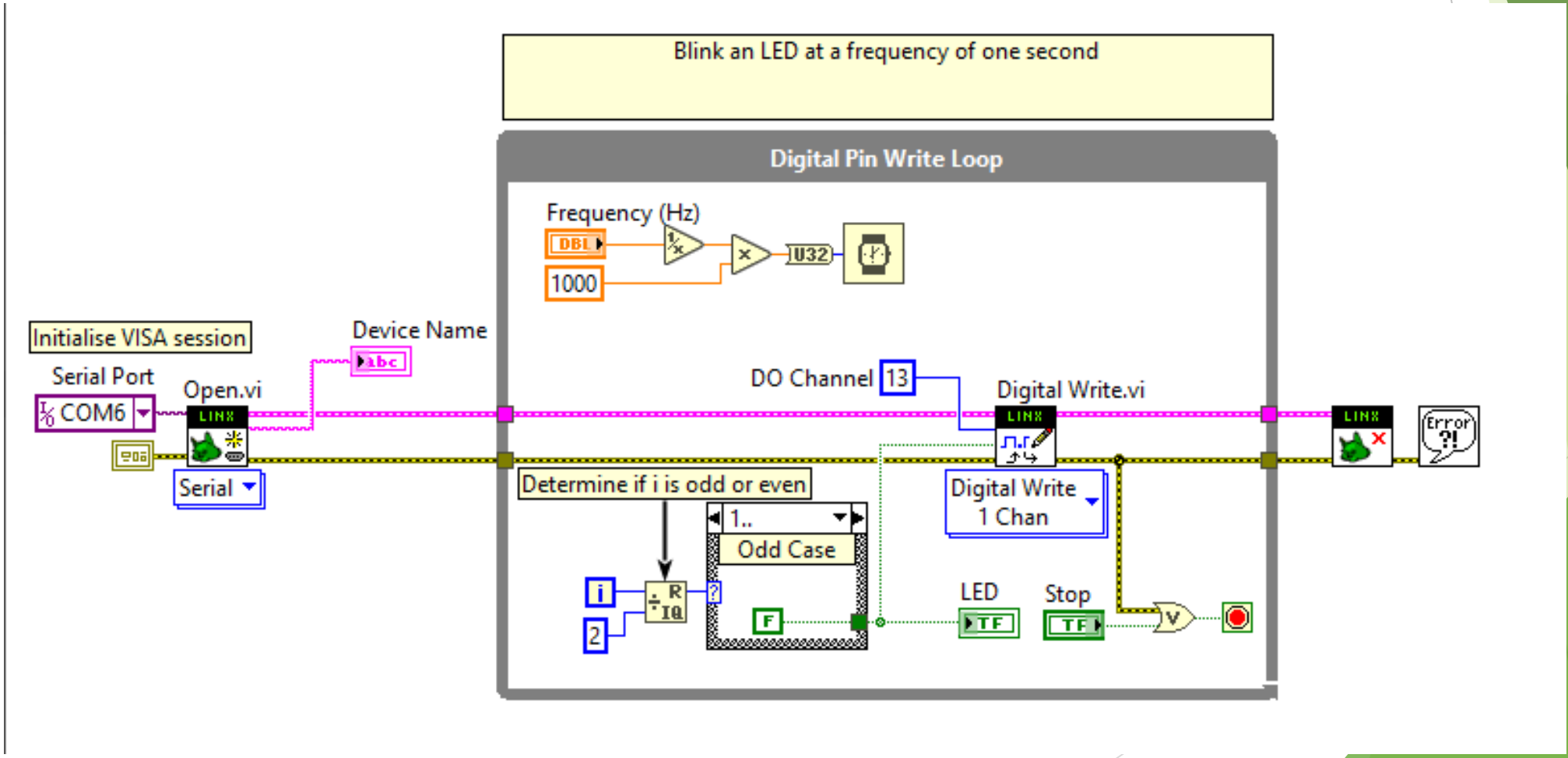
A 4:2 demultiplexer is shown below(implemented using AND and NOT gates).



LabVIEW Interfacing with Arduino:

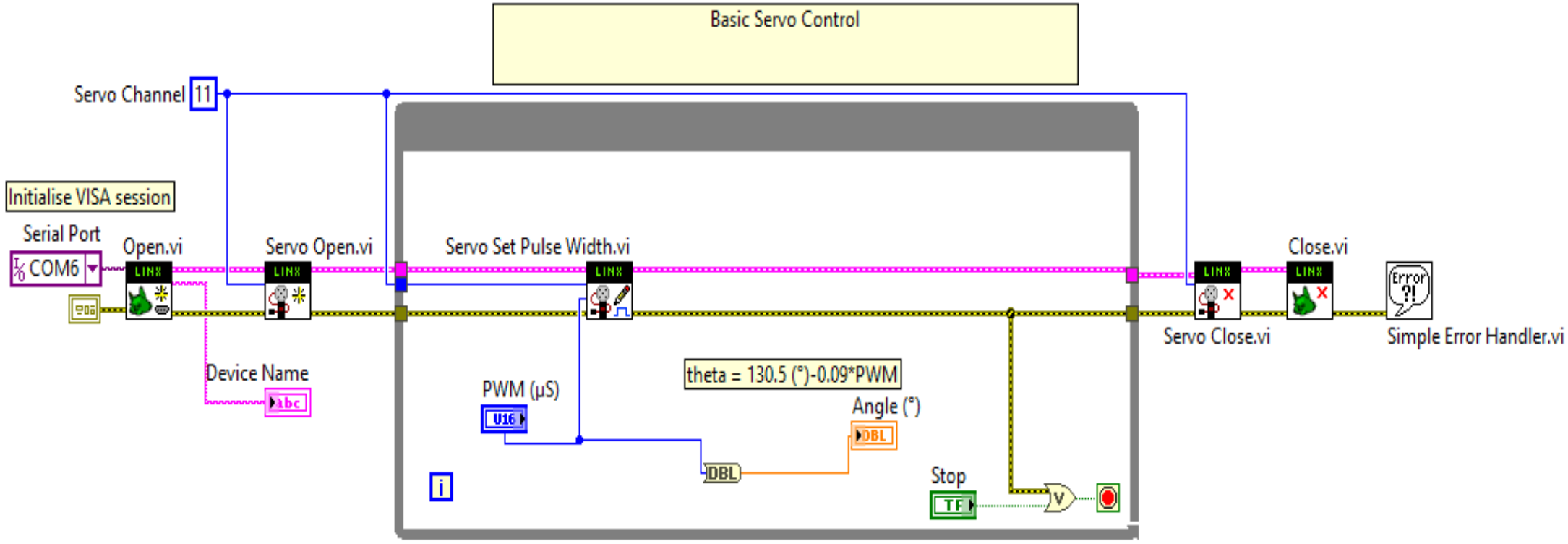
- LED Blinking Example:

Successful interfacing was done with the help of LINX library in LabVIEW to run the program in Arduino . Various examples are also given in this library for further circuit designing. The following Block Diagram is for the blinking of LED at various frequency .



Servo Control Example:

Here the Servo Motor is controlled/rotated using the Pulse Width Modulation(PWM) and the formula for the rotating angle is given by, theta = 130.5 (°)-0.09*PWM where theta is the angle of rotation. Here also it is implemented using the LINX library in LabVIEW to run the program in Arduino. Below is the block diagram.



Traffic Control System using Arduino:

The following block diagram is for the four-way intersection Traffic lights. Various delay timings have been used in the circuit and Boolean expressions are also present here. Below is not the complete block diagram, but it can be made by repeated use of the shown diagram.