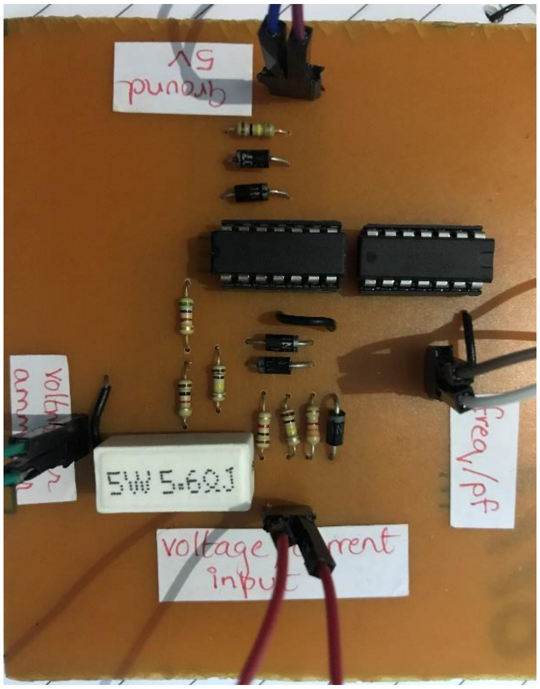
DIGITAL WATTMETER

ENA LAB PROJECT FALL 2019

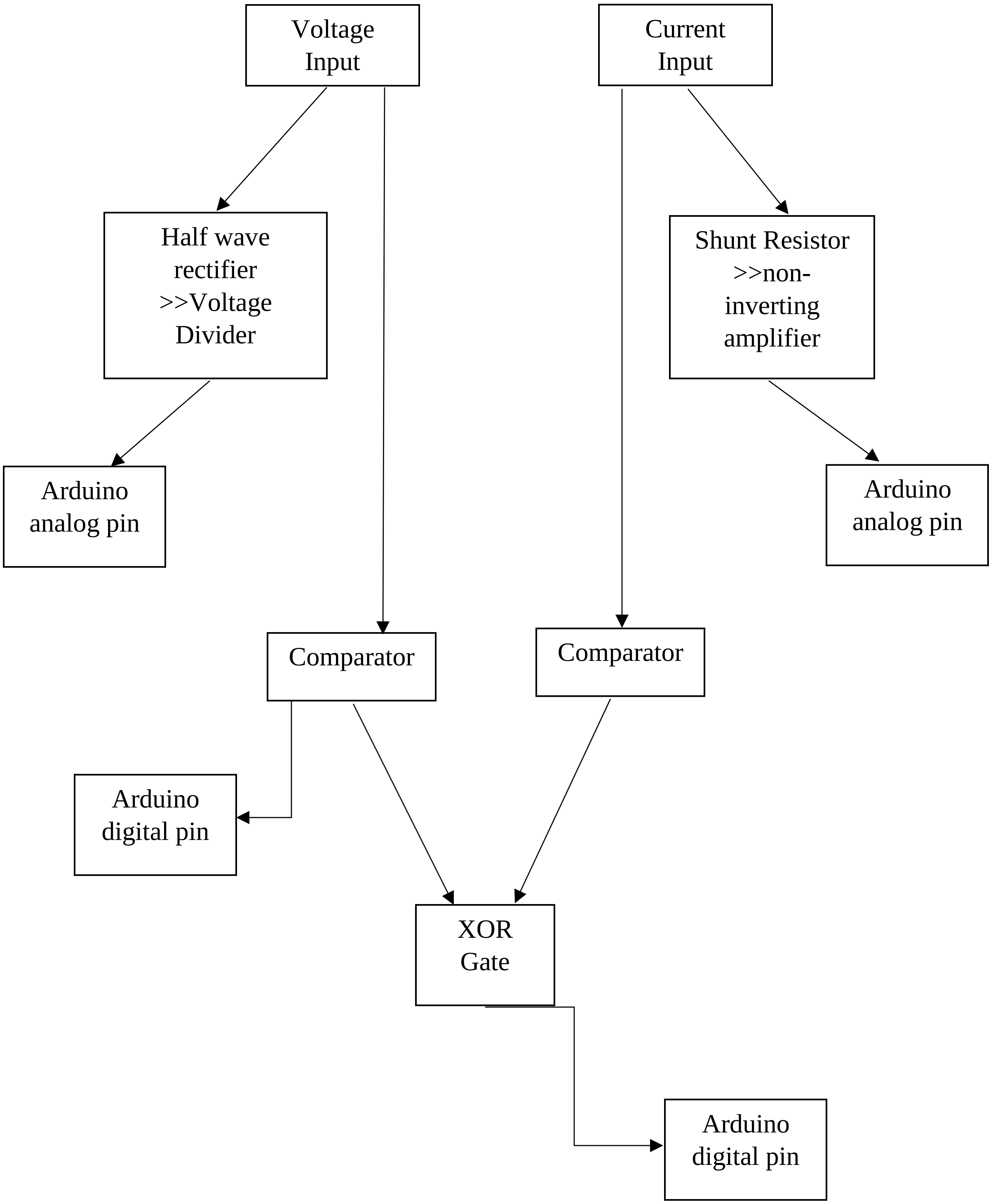
SUBMITTED BY AYESHA ISHAQ 18I-0772

Physical Description and operation:

This Digital Wattmeter is meant to measure AC voltage, current, frequency, power factor, apparent power, active and reactive power. The voltmeter is made using a simple half wave rectifier along with a voltage divider circuit to ensure only positive signal and maximum 5 volts at Arduino input. The ammeter consists of a shunt resistor across which voltage is measured and current is calculated. It incorporates a non-inverting amplifier to allow voltage to be amplified by a considerable amount for ease of measurement by Arduino. Next, frequency meter; this part of the circuit works by taking voltage input and converting it to a square wave using a Schmitt trigger comparator. The Arduino then reads the pulse time of the square wave to determine the frequency. Power factor calculation, is a step further in the same frequency meter circuit. Current input, that is, voltage across the shunt resistor is taken as input (as it is in phase with the current being measured) and converted to a square wave using a similar comparator. The output of these is then input to an XOR gate which outputs pulses corresponding to the phase shift. Using these we determine the angle of phase shift and power factor. Apparent power, active and reactive power can then easily be calculated using voltage, current and power factor.



Block Diagram



Calculations to select component values:

Voltage divider for max 20Vp:

5/20=1/4

Voltage divider of 1:3; values selected 1k ohm, 3.3k ohm.

Shunt: 5.6 ohm

Max of 100mA.

Max volts across shunt: 5.6x100m= 0.56V

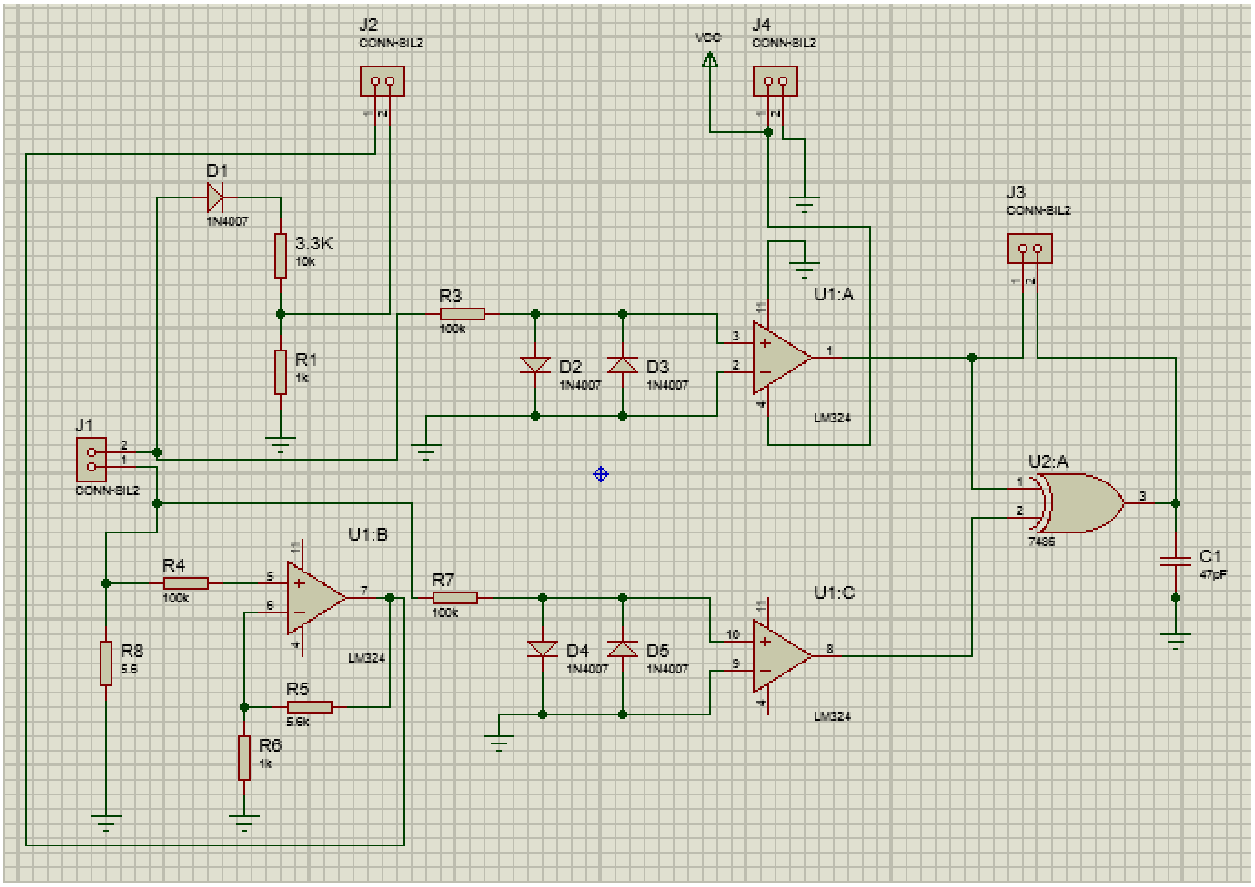
Amplifiying 0.56 to around 4 volts (to avoid saturation);

4/0.56= 7

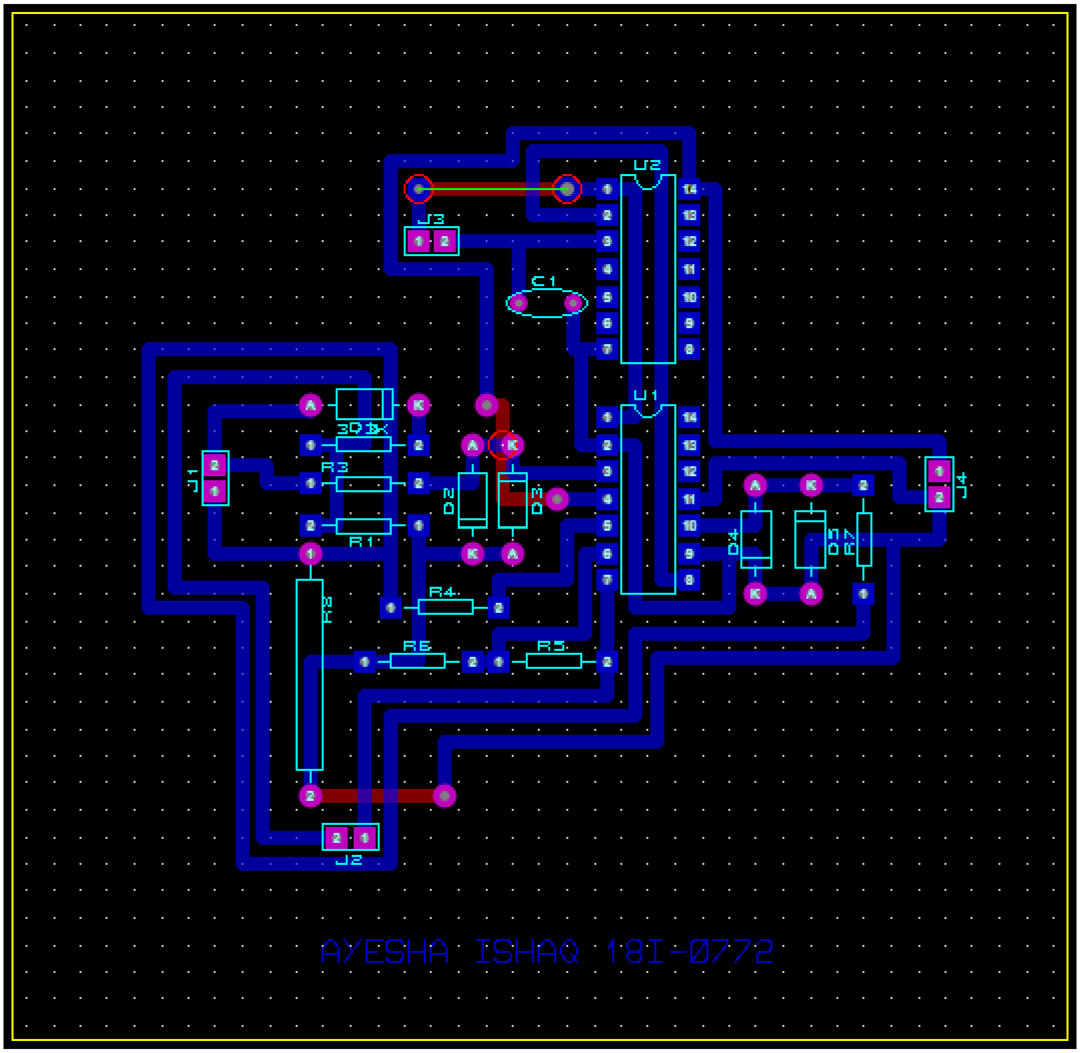
Feedback resistor selected: 5.6k ohm, input resistor 1k ohm;

Gain= (Rf/Ri +1) = (5.6/1) +1 = 6.6.

Schematic:



PCB Design:



Specifications:

Voltage range: 0-20Vp

Current range: 5-100mA

Components:

Resistors (3.3k, 1k 100k, 5.6k, 1M)

Capacitor (47pF)

Op-Amp (741, LM324)

XOR (7486)

Diodes (1N4007)

Arduino Code:

```
#include <LiquidCrystal.h>
float voltP=0, maxV=0, maxI=0, currP=0,
 voltRms=0, currRms=0;
float Rv1=3300, Rv2=1000, shunt=5.7,
Gain=6.6;
float ontime, offtime, freq, deltat,anR, anD, pf;
float appPow, actPow, reaPow;

LiquidCrystal LCD(4,6,10,11,12,13);

void setup(){
  LCD.begin(16,2);
  LCD.print("Digital");
  LCD.setCursor(0,1);
  LCD.print("Wattmeter");
  delay(1500);
   pinMode(7,INPUT);
  pinMode (2,INPUT);
}

void loop() {
  for(int i=0; i<1000; i++)
  {
    voltP=analogRead(A0);
    if(voltP>maxV)
    maxV=voltP;
  }
  for(int i=0; i<1000; i++)
  {
    currP=analogRead(A1);
    if(currP>maxI)
    maxI=currP;
  }
  maxV=maxV*5.0/1024;
  voltP=maxV*(Rv1+Rv2)/Rv1;
  voltP=voltP+0.7;
  voltRms=voltP*0.707;
  LCD.clear();
  LCD.print("Voltage");
  LCD.setCursor(0,1);
  LCD.print(voltRms); LCD.print("V");
  delay(1500);

  maxI=maxI*5.0/1024;
  currP=maxI/Gain;
```

```
currP=currP/shunt;
currRms=currP*0.707;
LCD.clear();
LCD.print("Current");
LCD.setCursor(0,1);
LCD.print(currRms*1000); LCD.print("mA");
delay(1500);

ontime=pulseIn(2,HIGH);
offtime=pulseIn(2,LOW);
freq=ontime+offtime;
freq=1000000/freq;
LCD.clear();
LCD.print("freq");
LCD.setCursor(0,1);
if(freq<1000)
{
LCD.print(freq); LCD.print("Hz");
}
else
{
LCD.print(freq/1000); LCD.print("KHz");
}
delay(1500);
LCD.clear();
ontime=pulseIn(7,HIGH);
deltat=ontime;
deltat=deltat/1000000;
LCD.print("time del");
LCD.setCursor(0,1);
LCD.print(deltat*1000);
LCD.print("ms");
delay(1500);
anD=deltat*freq*360;
anR=anD/57.128;
pf=cos(anR);
if(pf<0)
pf=-pf;
LCD.clear();
LCD.print("pf");
LCD.setCursor(0,1);
LCD.print(pf);
delay(1500);
```

```
  actPow=voltRms*currRms*pf;
  appPow=voltRms*currRms;
  reaPow=voltRms*currRms*sin(anR);
  if(appPow>0.01)
  {
  LCD.clear();
  LCD.print("Active Power");
  LCD.setCursor(0,1);
  LCD.print(actPow); LCD.print("W");
  delay(1500);
  LCD.clear();
  LCD.print("Reactive Power");
  LCD.setCursor(0,1);
  LCD.print(reaPow); LCD.print("VAR");
  delay(1500);
  LCD.clear();
  LCD.print("Apparent Power");
  LCD.setCursor(0,1);
  LCD.print(appPow); LCD.print("VA");
  delay(1500);
  }
  else
  {
  LCD.clear();
  LCD.print("Active Power");
  LCD.setCursor(0,1);
  LCD.print(actPow*1000); LCD.print("mW");
  delay(1500);
  LCD.clear();
  LCD.print("Reactive Power");
  LCD.setCursor(0,1);
  LCD.print(reaPow*1000);
LCD.print("mVAR");
  delay(1500);
  LCD.clear();
  LCD.print("Apparent Power");
  LCD.setCursor(0,1);
  LCD.print(appPow*1000);
LCD.print("mVA");
  delay(1500);
  }

}
```