Dynamic Computation of Load Power Factor through the Evaluation of Maclaurin *cos x* Function using 8051 Microcontroller Architecture

Golam Mostafa
Department of Electrical and Electronic Engineering, Ahsanullah University of Science and Technology (AUST)
e-mail: mostafagb@yahoo.com

ABSTRACT
Power factor is an index of the utilization efficiency of electric energy at the load side. Linear inductive loads of the industries operate at low power factor and thus reduce the utilization factor of electric energy. To counteract it, the industry owners install PFI plant, which measures the load power factor and then switches capacitor banks as needed to raise the power factor. This paper presents a method [Eqn. 2], which finds the phase shift (θ) between v-i waves by detecting their zero crossing points, transform it into equivalent counts (N) and then pass it into the Maclaurin cos x Series for dynamic computation of power factor by 8051 microcontroller. The method has been tested using 8051 microcontroller (Fig. 1[2]) and the results are found in good agreement with the theoretically computed results (Table 1).

KEYWORDS: Zero Crossing Points, PFI Plant, Power Factor, Maclaurin Series, Inductive Load.

1. INTRODUCTION

Linear inductive loads such as induction motors of the industries draw reactive energy from the utility supply. As a result, the transmission lines are forced to carry more currents than expected and incur power losses. To avoid this abnormal operation of the transmission lines, the utility authority asks the industry owners to improve their load power factor, which in turn reduces the transmission line losses.

Power factor is improved by installing PFI plant at the industry yard. The PFI plant measures the power factor and then switches over capacitor banks across the load lines. The local capacitors supply the demanded reactive energy of the load. The transmission line is relieved from supplying the reactive energy and enjoys its healthy operating condition.

There are various ways of measuring power factor [1], [3] with relative merits and demerits and have been devised for both linear and non-linear loads. Some of these methods have used 'low-cost but less-accurate' zero-crossing detection techniques and some have used 'expensive but highly-accurate' sampling techniques. The method presented in this paper is for linear load and works on the detection of the zero crossing points of the voltage and current waves, which are available for most of the industrial loads. Our method is simple, low-cost, accurate and dynamic owing to its motivation for computing the power factor through the evaluation of the Maclaurin *cos x* (Section-4, Eqn. 2) function using 8051 microcontroller architecture.

It is hard to find reports on the internet literature about the application of Maclaurin *cos x* Function for the determination of power factor. The author's motivation for using this function to measure power factor came from his experience of writing 'Low Level Program [Appendix-C]' for converting 'n-bit Binary number' into equivalent 'BCD number' using 'High Speed Horner Rule (Appendix-C)'. This is the BINary-to-BCD (BIN2BCD) conversion subroutine that inspired the author transforming the Maclaurin *cos x* function into Eqn. 3, which is in 40-bit binary domain and requires the use of BIN2BCD subroutine for getting transformed back into decimal domain.

2. WORKS DONE

In this paper, we have primarily focused on the real time verification of the correctness of the power factor computing expression $gpf\ (10^{12}pf) = 10^{12} - 49348 * N^2$ [Eqn. 2], which we have approximately synthesized from the Maclaurin *cos x* series for dynamic computation of the power factor. The works are:

- Use of RC network (Fig. 5) to create voltage and current waves with varying phase shifts by changing the circuit component C_1.
- Detection of the positive zero crossing points of the voltage and current waves using discriminators (Fig. 5).
- Derivation of a single power factor pulse width (PW) using the zero crossing points and SR flip-flop (Fig. 5). The PW-width is directly proportional to the phase shift (the power factor, *pf*) between the voltage and the current waves.
- Precision measurement of the PW-width in terms of counts (N) using 1-MHz crystal clock and 16-bit Timer-1 of the 8051 microcontroller (Fig. 4).
- Algebraic manipulation to transform the counts (N) into equivalent phase shift (θ) [Eqn. 1].
- Algebraic simplification of the Maclaurin *cos x* series to derive the power factor expression in the form of : $gpf\ (10^{12}pf) = 10^{12} - 49348\ N^2$ [Eqn. 2].

- Algebraic manipulation to transform the *gpf* expression from Decimal-domain into BINary-domain [Eqn. 3].
- Algebraic verification through value substitution that the *gpf* expression does provide expected value of power factor (*pf*), which is, for example: $\cos\theta = \cos 30^0 = 00.86$.
- Formulating Software Control Structure to implement Eqn. 3 using the instruction set of the 8051 microcontroller.
- Experimental verification, using 8051 Microcontroller Trainer Kit (CMCKIT) (Fig. 3) [2], of the software routines and subroutines that are involved for the evaluation of *gpf* expression.
- Complete listing of the Control Program in the form of Data Structure (Fig. 1, 7), Flow Chart (Fig. 2), Schematic Diagram (Fig. 6).

3. TRANSFORM COUNTS (N) into PHASE SHIFT

Refer to pulse width measurement circuit of Fig. 4, we state that the internal 16-bit Timer-1 (TH1, Tl1) of the 8051 microcontroller is configured to work as a binary up counter with a preset value of 0000H. An internal 1MHz clock derived from a 12MHz crystal oscillator drives the Timer-1, T1 [Fig. 4].

The T1 starts counting the 1μs pulses of 1MHz source, when the switch SW2 is closed at the arrival of rising edge of PW-pulse at INT1-pin. The T1 keeps counting for the whole duration of the PW-pulse and stops counting at the arrival of falling edge of the PW-pulse. The falling edge opens the switch, SW2. It also sets the IE1-bit (Fig. 4), which the MCU polls to understand that the T1 has finished counting. If desired, the arrival of the falling edge of the PW-pulse could also be detected through INT1 interrupt.

The MCU reads the counts (N) from the T1 and relates it with phase shift (θ) in the following ways:

- N counts were accumulated over Nx1μs = N μs.
- 10 ms (Half cycle) of the 50Hz line frequency is equivalent to π radians
- Therefore, N μs is equivalent to:

$$\left(\frac{\pi * N * 10^{-6}}{10 x 10^{-3}}\right) = \left(\frac{\pi N}{10^4}\right)^c \text{ Radians ... (Eqn. 1)}$$

4. POWER FACTOR EXPRESSION FROM MACLAURIN *cos x* FUNCTION

The Maclaurin series for the expansion of *cos x* function is given by:

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!}$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots\ldots\ldots$$

$$\cos x \cong 1 - \frac{x^2}{2!}$$

$$\Rightarrow pf = 1 - \frac{1}{2!}\left(\frac{\pi N}{10^4}\right)^2 \text{ ; } x \text{ is replaced by } \frac{\pi N}{10^4}\text{, Eqn. 1}$$

$$\Rightarrow 10^8\, pf = 10^8 - \frac{\pi^2 N^2}{2}$$

$$\Rightarrow 10^8\, pf = 10^8 - 4.9348 N^2$$

$$\Rightarrow 10^{12}\, pf = 10^{12} - 49348 N^2$$

$$\Rightarrow gpf = 10^{12} - 49348 N^2 \quad \text{....(Eqn. 2)}$$

5. TRANSFORMATION OF *gpf* EXPRESSION FROM DECIMAL DOMAIN TO BINARY DOMAIN

In order evaluate Eqn. 2 using a binary computer such as an 8051 microcontroller, we need to transform it into binary-domain and then formulate the algorithm. The transformation simply involves the replacement of the decimal factors with their binary (hex) equivalents. The count (N) is already in binary format being the content of the T1-register of the MCU. Now, the required binary-domain of the *gpf* is:

gpf = E8D4A51000H – C0C4H * N^2 ...... (Eqn. 3)

6. VERIFICATION OF *gpf* EXPRESSION BY SUBSTITUTIONS

Let us verify the correctness of the *gpf* expression of Eqn. 3 by substituting the value of N for a given simulated phase shift of 30^0. We need to abide by the following steps:

- Find binary value of N for the phase shift of 30^0.

  For an 180^0 phase shift between the voltage and current waves, the equivalent time shift is 10ms ($10*10^{-3}$ sec). Therefore, for the given phase shift of 30^0 between *v-i* waves, the time shift is approximately $1666*10^{-6}$ sec. This indicates that the PW-pulse's width is 1666 μs.

  The above result further indicates that the T1of MCU of Fig. 4 counted pulses for 1666 μs. Since, the period of a driving clocking pulse of T1 is 1μs, the T1 counted 1666 pulses. Therefore, the value of N is 1666 (0682H).
- Evaluate Eqn. 3 with the value of 0682H for N.

  ⇒ *gpf* = E8D4A51000H–C0C4H * 0682H*0682H

  ⇒ *gpf* = E8D4A51000H – C0C4H * 2A5A04H

  ⇒ *gpf* = E8D4A51000H – 1FE3EFEB10H

  ⇒ *gpf* = C8F0B524F0H

  ⇒ *gpf* = 863031862512d [d for decimal system]

  ⇒ 10^{12} * *pf* = 863031862512

  ⇒ *pf* = (863031862512) /10^{12}

  ⇒ *pf* = .863031862512

  ⇒ *pf* ≈ 00.86 = cos 30^0 ; verified

7. FORMULATING CONTROL STRUCTURE TO IMPLEMENT *gpf* EXPRESSION

In Section-6, we have validated our power factor computing formula, Eqn. 3 through substitution. We intend to test it through the hardware intelligence of the 8051 microcontroller. Therefore, we need to submit this formula to the 8051 in the form of a program, which can be easily coded into executable binary bits using the instruction set of the 8051.

- L1: Initialize everything as needed
- L2: Design subroutine (SUR) to perform Binary multiplication (BMULT).
- L3: Design SUR to perform multi-byte binary subtraction (MSUB).
- L4: Design SUR to perform 40-bit Binary-to-BCD conversion using high speed Horner Rule (BIN2BCD).
- L5: Design SUR to convert multi-byte BCD data into CC-codes data (BCD2CC).
- L6: Design SUR to transfer CC-coded data onto CC-type 7-segment display devices.

The above text codes may be coded as follows by the assembly syntax of the 8051 microcontroller.

```
• L1:   LCALL  INIT      ; initialize as needed
• L2:   LCALL  BMULT     ; to perform N*N → N
  L2A:  LCALL  BMULT     ; to perform N*C0C4H
• L3:   LCALL  MSUB      ; multi-byte subtraction
• L4:   LCALL  BIN2BCD   ; using Horner Rule
• L5:   LCALL  BCD2CC    ; data formatting
• L6:   LCALL  CCX7SD    ; data xfer to display
```

8. EXPERIMENTAL VERIFICATION OF THE CONTROL STRUCTURE OF SECTION-6

All the principles that we have so far accumulated for computing power factor based on zero crossing detection of the voltage and current waves have strong mathematical support. Now, we wish to test these principles using actual hardware of the 8051 microcontroller. We have chosen the CMCKIT [2] (Fig. 3) for this purpose. It is an 89S52 (a family member of 8051) microcontroller based Trainer and Writer Kit. The set up under the CMCKIT requires the following resources[2]:

- The CMCKIT
  It is equipped with breadboards for placing the 7-segment display devices and the circuitry of Fig. 5. The Kit contains an 89S52 MCU to drive peripheral devices through jumper connections.
- IBMPC with WINXP operating system,
- WINASM5: It is a window-based assembler for 8051 microcontrollers.
- USB↔RS232 Converter Cable: This cable allows the IBMPC to communicate with the CMCKIT over USB port.
- CMCKTVCOM3: This is a GUI interface that allows the IBMPC to download and burn codes into the program memory of the target 89S52 microcontroller.
- PFM.asm: The complete ASM program downloadable from [2], which when fused in the target MCU of the CMCKIT, measures the PW-width, performs all calculations and then puts the power factor on 7-segment display devices.

9. CONCLUSIONS

In this paper, we have wished to document our field works on the "applications of 89S52 MCU and BIN2BCD subroutine through the evaluation of Maclaurin *cos x* function for measuring power factor of linear loads"; for the readers who wish to acquire microcontroller technology through the reproduction of results of working projects. To meet this purpose, we have provided complete setup, procedures, circuits and program codes.

The results of Table 1 are accumulated through actual measurements by varying the component C1 of Fig. 5. The results are correct and could be validated by putting the measured value of N in Eqn. 2. The errors are found to be within ±3% tolerance.

However, as we see in Table 1, the %error increases as the phase shift between *v-i* increases but this happens when the power factor falls below 0.7. This is not a problem for us as we maintain the power factor between 0.85 – 0.95.

10. FUTURE SCOPE OF WORKS

The works presented in this paper is a self-contained project and might help the academicians, researchers and engineers to find more and more interesting projects on electrical power theory to implement using microcontroller hardware.

Additional circuits could be inserted to detect the leading/lagging status of the current wave with respect to voltage wave. The techniques of this works could be applied to design Power Factor Improvement Plant for smooth regulation of the load power factor.

REFERENCES

[1]. http://www.freepatentsonline.com/6756771.html; For Power factor correction method with zero crossing detection & adjustable stored ref voltage

[2]. www.krdcbd.com for CMCKIT – an 89S52 Microcontroller Trainer/Write Kit and PFM.asm

[3]. A.R. Al-Ali. M/M.Nagm; A PLC Based Power Factor Controller, IEEE, 2000; p1065.

[4]. G. Mostafa, "8086 Microprocessor Interfacing and System Design", Karighar R&D Center, 2009, p354, p407.

[5]. J.B. Peatman, "Microcomputer Based Design", McGraw-Hill Book Company, 1988, p402.

APPENDIX-A: Data Structure for the Control Program PFM.asm

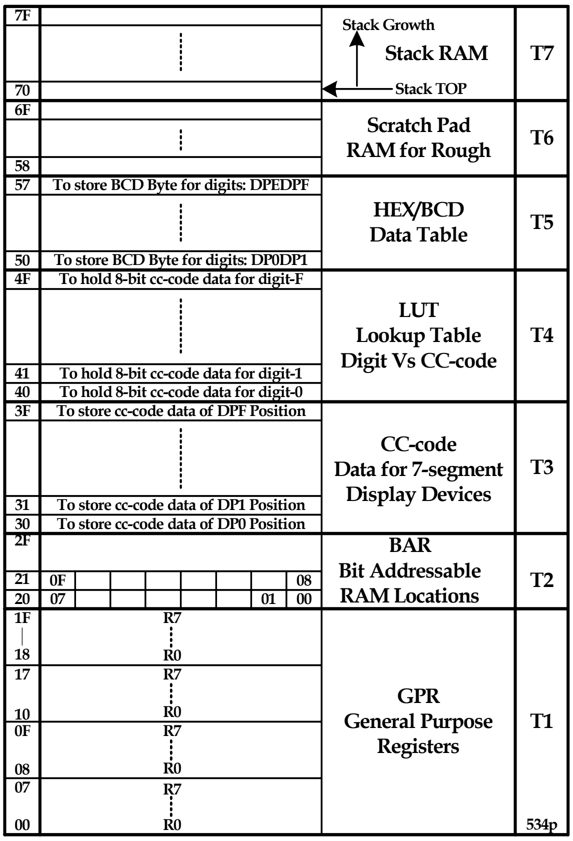


Fig. 1 Data Structure for Control Program PFM.asm

APPENDIX-B: Control Program Flow Chart for Program PFM.asm

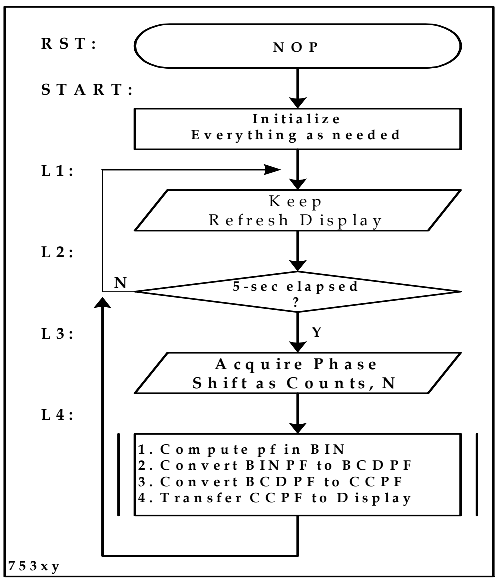


Fig. 2 Flow Chart for Control Program PFM.asm

APPENDIX-C: Horner Rule for Converting 40-bit (range: 0000000000h - FFFFFFFFFFh) Binary into BCD (range:00000000000000 – 01099511627755)

Introduction:

In 1711, Isaac Newton presented the polynomial $A(x) = a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\ldots\ldots\ldots+a_2x^2+a_1x^1+a_0x^0$, which later on in 1819, was modified by W.G.Horner for algebraic simplification and became to be known as Horner Rule. At the advent of digital computers, engineers have applied this rule for BINary-to-BCD and BCD-to-BINary conversions in connection with the design of instruments. But, unfortunately, very little information about the implementation mechanism of this rule has been recorded in the literature except [5]. The author had the opportunity to do extensive work on Horner Rule while designing Digital Weighing Machine in 1998[4].

Derivation of Horner Rule for 40-bit Input Binary

$BIN = b_{39}b_{38}b_{37},\ldots\ldots\ldots\ldots\ldots,b_2b_1b_0$

$BCD = b_{39}x2^{39}+b_{38}x2^{38}+b_{37}x2^{37}+\ldots.+b_2x2^2+b_1x2^1+b_0x2^0$

$BCD = (b_{39}x2^{38}+b_{38}x2^{37}+b_{37}x2^{36}+\ldots.+b_2x2^1)2+b_0$

……………………………………………………………

$BCD = (((\ldots(b_{39})2+b_{38})2+b_{37})2+\ldots\ldots\ldots+b_2)2+b_1)2+b_0$

$BCD=(((\ldots.(IPBCDx2+b_{39})+ b_{38})2+\ldots\ldots+b_2)2+b_1)2+b_0$

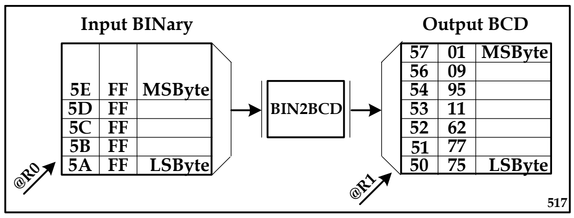
Where: IPBCD stands for Initial Partial BCD and is 00.

Control Structure for Computation:

for (x=39 to 0)

$IPBCDx2+b_x \rightarrow IPBCD$

8051 Data Structure for Computation



8051 Assembly Codes for BIN2BCD Subroutine:

```
;========================================
BIN2BCD:
       MOV    R2, #32H      ; bits in input BINary
LX1:   ;-- catch bit-39,38,37,…,2,1,0 ------------
       MOV    R3, #05H      ; input bytes
       MOV    R0, #59H      ; points Input Table
       ;-------
NXTB:  INC    R0
       MOV    A, @R0        ; getting 1st BIN Byte
       RLC    A
       MOV    @R0, A
       DJNZ   R3, NXTB      ; Rotate next byte
       ;--evaluate : IPBCDx2+bx (39-0) → IPBCD-----
       LCALL  EVBCDHR
       DJNZ   R2, LX2       ; catch next MS-bit
       RET
LX2:   MOV    R3, #05H
       MOV    R0, #59H
       LJMP   LX1
;---------------------------------------------
EVBCDHR:;Evaluate to BCD using Horner Rule
       MOV    R4, #07H      ; 7-byte to update
       MOV    R1, #4FH      ; points BCD Table
       ;------
UPDNB: INC    R1
       MOV    A, @R1
       ADDC   A, A          ; perform: IPBCDx2+bx
       DA     A             ; adjust to correct BCD
       MOV    @R1, A
       ;------
       DJNZ   R4, UPDNB     ; update next byte
       ;-------
       RET
;=============================================
```

Table 1

Theoretical Calculations of Electrical Quantities From Circuit of Fig. 3			Practical Measurements of Electrical Quantities Using CMCKIT		
A	B	C	D	E	F
Capacitor C1 (F)	Phase Shift [Eqn. 4] $\theta = \tan^{-1}(\frac{0.184}{C_1 * 10^6})$	pf $\cos\theta$	Power Factor pf	Counts (Hex) N	%Error (C-D) pf
$0.6x10^{-6}$	17.05^0	0.95	0.94	043B	-1%
$0.5x10^{-6}$	20.02^0	0.93	0.92	050D	-1.1%
$0.4x10^{-6}$	24.70^0	0.90	0.88	0616	-2.2%
$0.3x10^{-6}$	35.02^0	0.85	0.83	0756	-2.3%
$0.2x10^{-6}$	42.61^0	0.73	0.74	08FA	+2.7%

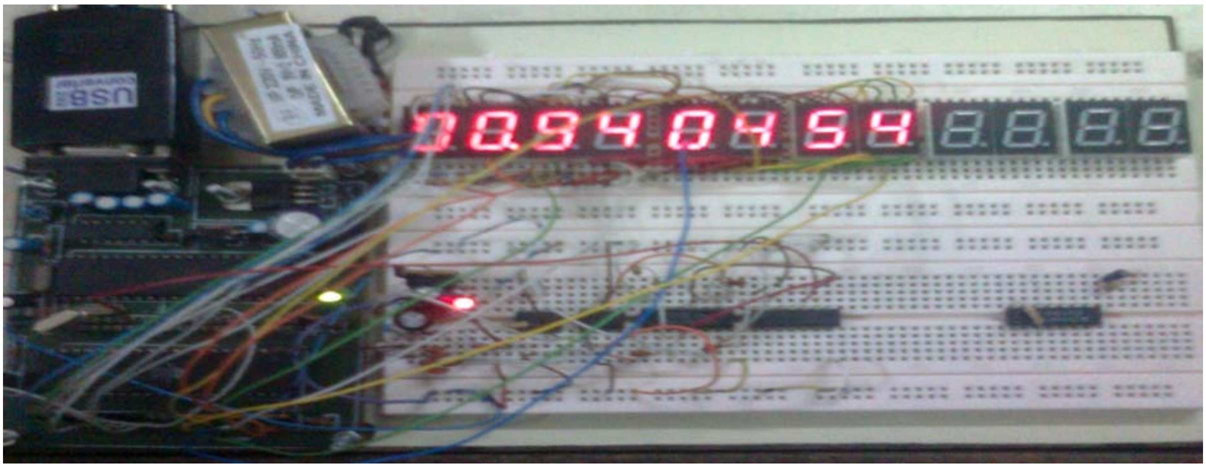


Fig. 3 Power Factor Meter Set up using CMCKIT – Atmel 89S52 CISC Microcontroller Trainer and Writer Kit

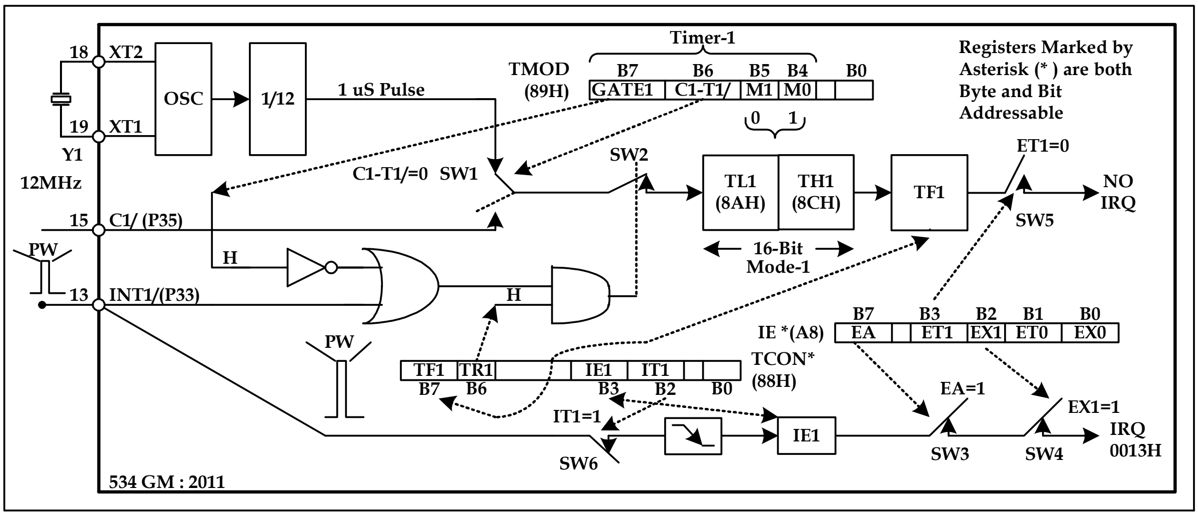


Fig. 4 Internal Structure of Timer-1 of the 89S52 Microcontroller

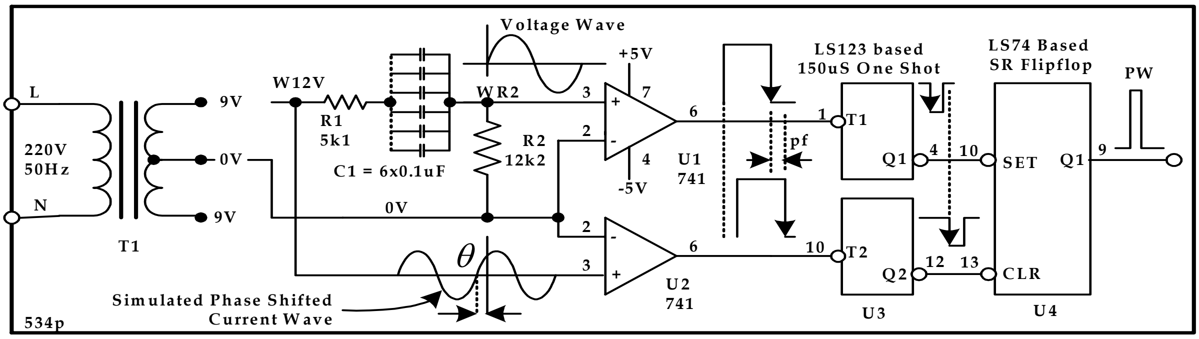


Fig. 5 Generating phase shifted voltage and current waves for creating artificial Power Factor Pulse (PW)

The phase shift θ between the simulated voltage and current waves of the above circuit is:

Impedance, $Z = 17.3*10^3 - j\frac{1}{2\pi*50*C_1} = 17.3*10^3 - j\frac{3.18*10^{-3}}{C_1}$

Phase shift, $|\theta| = \tan^{-1}(\frac{3.18*10^{-3}}{C_1}*\frac{1}{17.3*10^3}) = \tan^{-1}\frac{0.184}{C_1*10^6}$ [C_1 is in F]……………………(Eqn. 4)

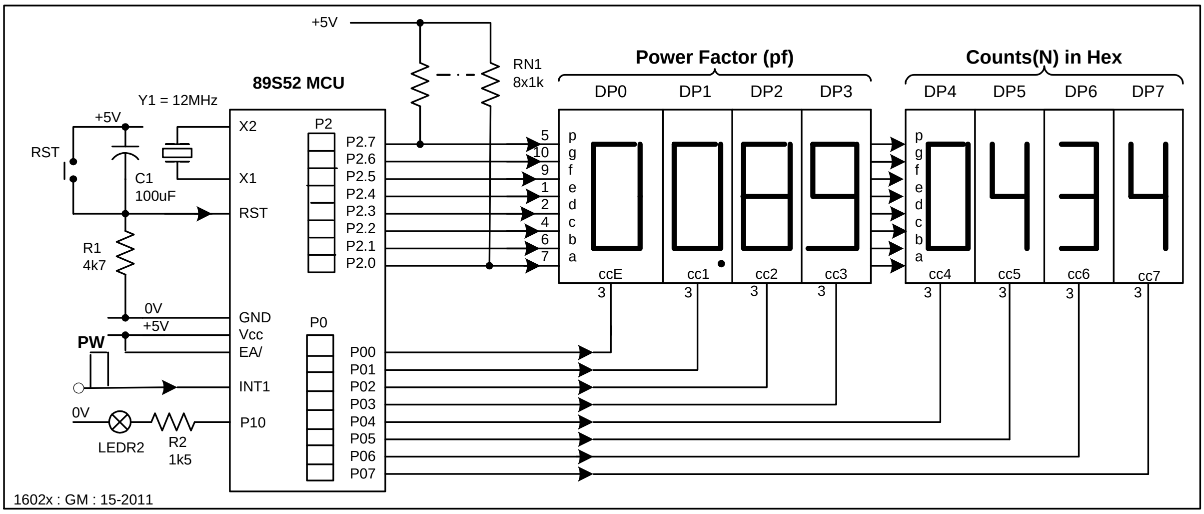


Fig. 6 Schematic Diagram for the 89S52 MCU based Power Factor Meter

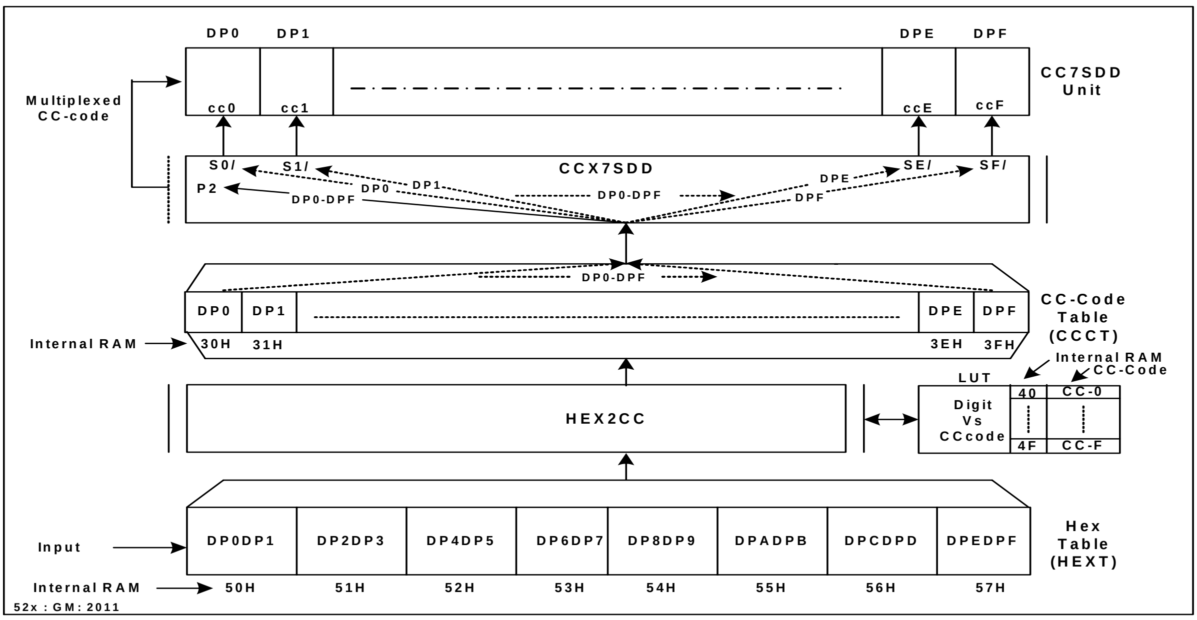


Fig. 7 Data Structure for the Data Display Mechanism of Power Factor Meter