ANALYSIS AND IMPLEMENTATION
OF
MICROCOMPUTER CONTROLLED
PWM RECTIFIER

By

Mohammad Shahidul Islam

A Thesis
submitted to the Department of Electrical and Electronic Engineering,
BUET, Dhaka
in partial fulfillment of the requirements
for the degree
of
Master of Science in Electrical and Electronic Engineering

Department of Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology

July, 1996

Dedicated
to
my Parents

APPROVAL

The thesis "ANALYSIS AND IMPLEMENTATION OF MICROCOMPUTER CONTROLLED PWM RECTIFIER", submitted by Mohammad Shahidul Islam, Roll No. 921313P, Session 1990-91-92 to the Department of Electrical and Electronic Engineering , BUET has been accepted as satisfactory for partial fulfilment of the requirements for the degree of Master of Science in Electrical and Electronic Engineering. His thesis defense was held on July 3, 1996 and his performance in the defense was found to be satisfactory. The title of thesis was approved by CASR in meeting 118 dated 10.09.95 by resolution 13. The board of examiners was approved by CASR meetings 119 dated 23.10.95 by resolution 20 and 123 dated 17.06.96 by resolution 37.

BOARD OF EXAMINERS

1. **Dr. Mohammad Ali Choudhury**
Associate Professor
Department of Electrical & Electronic Engg.
BUET, Dhaka-1000,
Bangladesh.

Chairman
(Supervisor)
3-7-96

2. **Dr. A.B.M. Siddique Hossain**
Professor and Head
Department of Electrical & Electronic Engg.
BUET, Dhaka-1000,
Bangladesh.

Member
(Ex-officio)
3/7/96

3. **Dr. M. Rezwan Khan**
Associate Professor
Department of Electrical & Electronic Engg.
BUET, Dhaka-1000,
Bangladesh.

Member
3.7.96

4. **Dr. Kazi Khairul Islam**
Associate Professor
Department of Electrical & Electronic Engg.
BIT, Rajshahi,
Bangladesh.

Member
(External)
03.07.96

DECLARATION

I hereby declare that this work has been done by me and it has not been submitted elsewhere for the award of any degree or diploma.

Countersigned

machoudhury

(Dr. Mohammad Ali Choudhury)
Supervisor

S. Islam

(Mohammad Shahidul Islam)

CONTENTS

Page No.

Page No.

ACKNOWLEDGEMENT

Keen interest of Dr. Mohammad Ali Choudhury in the field of Power Electronics has influenced me to carry out a research work in this field. His constant guidance, supervision and suggestions at all stages have made it possible to complete this work. His encouragement during difficult times will always be remembered. I express my deep sense of gratitude to Dr. Mohammad Ali Choudhury.

I take this opportunity to express my heart felt gratitude and thanks to Dr. A.B.M. Siddique Hossain, Professor and Head of the Department of Electrical and Electronic Engineering, BUET, for his kind help and cooperation during this work.

I am indebted to my friends and colleagues Mr. Monzur Morshed and Mr. Nazmul Anwar for their sincere cooperation and valuable material support in this work.

I would like to express thanks to Mr. Kazi Mujibur Rahman, Assistant Professor, BIT, Chittagong for providing valuable suggestions. I would also like to offer thanks to my colleague Mr. Mizanur Rahman for his nice support and company during the work.

Finally, I wish to extend my heartiest thanks to my respected teachers and to all my well-wishers those who directly or indirectly helped me in this work.

ABSTRACT

Traditional phase controlled rectifiers are rugged, simple, easy to control and need very low maintenance. Demerits of these converters are low power factor and generation of low frequency harmonics at input / output side of the converters. Low frequency harmonics (supply frequency multiples) necessitates large sized filters at the input / output side of these converters. To compensate poor performance and enhance conversion efficiency, one of the suggested method is to adopt pulse width modulated (PWM) switching. Various types of sine PWM techniques exist for switching static converters. Microcomputer controlled PWM waveform generation has been the effort of researchers for a long time. As a result various off-line microcomputer control strategies have been developed for controlling PWM static converters. But true on-line implementation did not realize thus far due to the difficulty of on-line computation of switching points to reproduce the waveforms without interruption of the service of converters. The recently developed techniques for calculating switching points of PWM waves involve solution of single algebraic equation in each modulation pattern which may be used in the on-line computation and implementation of PWM waveform generation. Microcomputer control of PWM converters is usually achieved by storing switching points in EPROM as a look up table and outputting the waveform for a desired frequency of operation as required. In this research the main objective is the on-line computation of switching points of static converters according to the sine pulse width modulation scheme and produce the waveforms to switch a three phase rectifier. The switching point generation by microcomputer is implemented by 80386 DX-2 processor based microcomputer with the keyboard as input and pin 2-pin 7 of the parallel port of the computer as the output. The programming part uses high level Turbo C++ language. The output of the computer is used to control a six pulse three phase MOSFET rectifier through an interface developed during this thesis.

LIST OF FIGURES

LIST OF TABLES

LIST OF FLOWCHARTS

Chapter 1

1.1 Introduction

Performance improvements achievable by the pulse width modulation (PWM) techniques has increased the operation of static ac to dc converters in various applications. Implementation of these techniques has been facilitated by the advent of power transistors offering outstanding characteristics of high power handling capabilities, fast switching and the absence of commutation circuitry in converter topologies.

Phase controlled converters are simple, less expensive and high efficiency converters having efficiency in general above 95%. They are particularly attractive due to their inherent ruggedness, simplicity, low maintenance and ease of control afforded by the natural commutation. Demerits of these converters are low power factor and generation of low frequency harmonics. The harmonics generated tend to pollute the power network and cause interference with other equipment in close proximity. In dc drives, the ripple causes overheating and general derating of the motor.

To compensate the poor performance and improve the conversion efficiency, various methods have been proposed. Principal schemes for power improvement include:

1. Reactive compensation

2. Multiphase rectification

3. Sequential control and

4. PWM techniques etc.

1.1.1 Reactive Compensation [1,2]

Basically, this method is a filtering approach, utilizing shunt inductor-capacitor (LC) filters at the ac side and a smoothing reactor in series at the output dc stage. The LC filter located at the harmonic source is tuned to response at low order harmonics. The filter serves as a low impedance path for the harmonic current to flow, virtually eliminating their presence in the ac system. Although the scheme results in a definite improvement in the current waveform, it has the following disadvantages:

- A separate filter is required for every major harmonic component, or alternatively, continuous tuning is essential to eliminate troublesome harmonics
- Due to the magnitude of current and low harmonic components, large sized inductors and capacitors are required
- The filter causes voltage fluctuations, and
- It results in increased loss.

1.1.2 Multiphase Rectification [2,3]

Since harmonic currents are a function of converter pulses, improvement in power factor can be obtained by increasing the number of output pulses. In general, an *n* pulse converter generates harmonic components of the order of $nk \pm 1$, where k is any integer. For example, a six pulse converter contains harmonics of orders 5th, 7th, 11th, 13th, . . . , and correspondingly, a twelve pulse converter is characterized by 11th, 13th, 23rd, 25th, . . . , harmonics. This procedure, using a higher number of phases for low order harmonic cancellation is referred to as phase multiplication. Multiphase rectification is restricted to high power ac to dc converters such as high voltage dc transmission (HVDC) systems, where, the cost of additional power apparatus and complex circuitry are justified.

1.1.3 Sequential Control [4,5]

Improvement in power factor can be achieved by cascading two converter bridges and using sequential control. In this process, one bridge is maintained in full advance (rectifying) or full retard (inverting) and the other bridge is controlled. Since one converter operates as a diode bridge, the cascade connection simulates a semi converter in both the motoring as well as the regenerating modes, resulting in an improvement in displacement angle. The sequential control method is complicated, expensive and only partially effective and hence it has found limited use.

1.1.4 PWM Techniques

The line commutation method offers little flexibility in controlling thyristor switching due to the natural commutation of thyristors by line voltage and subsequent conduction of thyristors on different phases. Hence, the phase angle control technique is more or less restricted as the only control parameter in thyristor converters, where, the turn-on / turn-off are dictated by converter operation. In contrast , the forced commutation principle allows thyristor commutation at any desired instant by providing each thyristor with its own commutation circuit. In other words, the use of forced commutation increases the versatility of the converter and permits direct control of switching devices to improve performance.

Initially, control schemes incorporating forced commutation were based on a single pulse approach. Various single pulse control schemes have been reported in the literature [5,7]. When used in conjunction with the afore mentioned schemes, single pulse triggering scheme provides a general improvement in converter performance. It was soon realized that the maximum power factor and reduction in low order harmonics could be achieved by using multiple pulses per cycle. This opened the avenue for PWM techniques, where, thyristor switching are governed by certain modulation laws such that the ac waveform closely resembles a sinusoid.

PWM techniques have gained considerable attention in recent years due to the optimum performance attainable with a simple converter topology. The salient features of a PWM controlled ac/dc converter are,

- High operating power factor ,
- Unity displacement factor,
- Negligible lower order harmonics,
- Controllability and
- Reduced filter size.

Limitations of the process are :

- Complex control circuits,
- Auxiliary commutation circuits for SCR converters and
- Increased losses due to high switching frequency.

1.2 Review of PWM Techniques

Various types of PWM techniques exist and for converter switching the switching techniques used can be categorized as,

1. Natural Sampling,
2. Uniform Sampling,
3. Optimal Sampling, and
4. Adaptive control PWM.

1.2.1 Natural Sampling

In the natural sampling technique as shown in Fig. 1.1, the switching points are determined by the crossings of the modulating sine wave and an isosceles triangle carrier wave. This technique, most commonly referred to as Sinusoidal PWM [7-9], has the modulated waveform containing harmonics as a function of the ratio of carrier to modulating frequencies. The frequency ratio is defined as the ratio of the carrier frequency to modulating frequency. The higher the frequency ratio, the more distant the dominant harmonics are from the fundamental component. Another version of the carrier modulated PWM technique described above, uses a triangular wave and a constant dc reference waveform. In this case, variation in the dc level controls the pulse widths. Similar to the Sinusoidal PWM, the harmonic content in this method is a function of the triangular wave frequency [10]. High carrier frequency causes the dominant harmonics to appear in the upper frequency range.

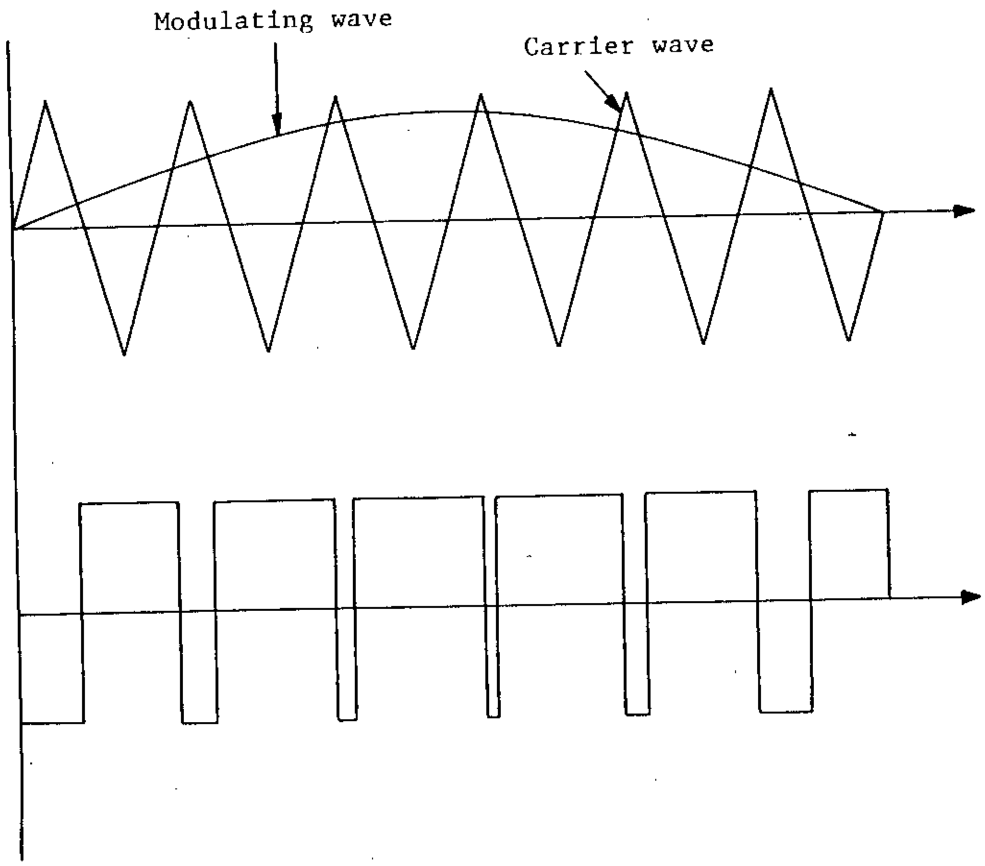


Figure 1.1: PWM strategies- Natural Sampling.

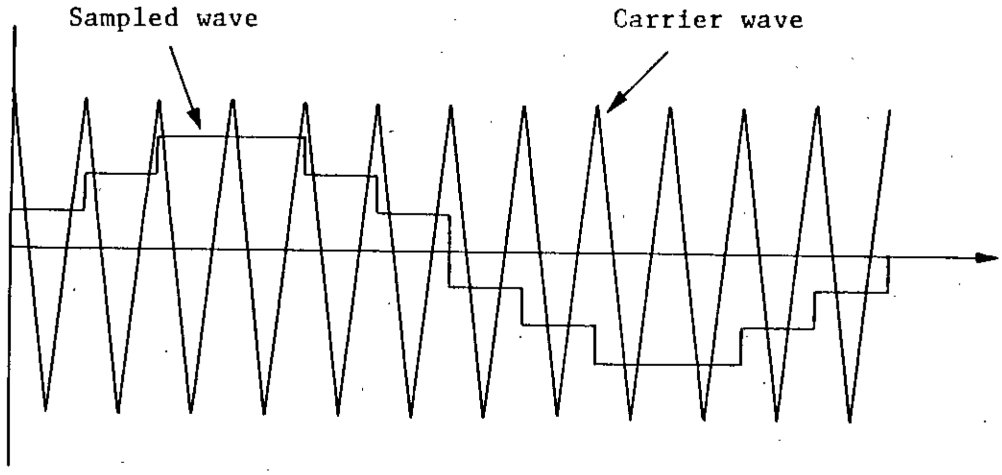


Figure 1.2: PWM strategies- Uniform Sampling.

2.2 Uniform Sampling

Uniform sampling PWM technique is based on a sample and hold principle having modulating wave represented by piecewise linear segments. The uniform sampling approach as shown in Fig. 1.2, is essentially a derivative of the natural sampling process, with the continuous sine wave replaced by an equivalent stepped version. It has the advantage that the sampling pulse can be made symmetrical about the trough of the carrier. This results in a significant reduction of low frequency harmonics and elimination of subharmonics at non-integer frequency ratios. The technique is particularly suitable for microprocessor implementation.

1.2.3 Optimal PWM

Optimal PWM strategies are based on the minimization of certain criteria [11,12] like elimination or minimization of particular harmonics, harmonic current distortion, peak current, etc. As opposed to the natural or uniform sampling methods, optimal PWM techniques require the PWM waveform to be defined a priori in terms of the desired switching instances and followed by their determination through numerical techniques. Optimal PWM techniques are computationally intensive and require dedicated microprocessors for successful implementation.

1.2.4 Adaptive PWM

In principle, adaptive PWM techniques utilize a "bang-bang" hysteresis control approach to minimize a desired signal (usually an error signal) within a certain limit. In this approach, the switching instances are intrinsically determined by the intersection of the error signal with the upper and lower hysteresis boundaries. For a sinusoidal reference signal, the modulated waveform has a fundamental frequency component equal to the reference frequency and the dominant harmonics appear at the ripple frequencies. Therefore, by adjusting the hysteresis limits, dominant harmonics can be laterally shifted towards the higher end of the frequency spectrum. The adaptive PWM technique has a dual advantage:

1. It allows a closed loop control process, and

2. The modulation process inherently eliminates lower order harmonics.

1.3 Computer Controlled DC Converters

1.3.1 Review

The advent of microcomputers since the beginning of the 1970's has brought a new dimension to power electronics and drive technology. In the beginning of the microcomputer era, microprocessors were essentially used for implementation of logic control functions. As the microcomputer improved in bit size, execution time, and hardware functional integration, its application gradually expanded to general control systems. In a power electronics system, the microcomputer functions can in general be categorized as follows:

- Control of feedback loop,
- Gate firing control of phase-controlled converters,
- PWM or square-wave generation of inverters,
- Optimal and adaptive control,
- Estimation of feedback signals,
- General sequencing control,
- Protection and fault overriding control,
- Signal monitoring and warning,

- Data acquisition,
- Diagnostics, and
- Miscellaneous computation and control.

Traditional phase controlled rectifiers are rugged, simple, easy to control and need very low maintenance. Demerits of these converters are low power factor and generation of low frequency harmonics at input / output side of the rectifier. Low frequency harmonics necessitates large filters at the input/output side of the converters. Microcomputer controlled PWM waveform generation to control converters has been the effort of researchers for a long time

Microcomputers have now been accepted universally in power electronics and drive systems and the spectrum of their application will grow continuously to include more time critical applications such as switching regulators. In the near future microcomputers will play a vital role not only in higher level supervisory control but also in lower level control of power electronics and drive systems.

In the year 1982 a firing scheme has been used based on a microprocessor to control a three phase thyristor dual converter using look-up table algorithm to speed up operational response [13]. It gives a full range of control of the firing angle between 0° and 180° for both positive and negative control. The implementation used complete digital circuits requiring no adjustments during operation and hence the system was less costly having more reliability.

Also, in 1982 a six pulse converter firing pulse generator had been successfully implemented using a standard eight bit microprocessor [14]. Programming technique employed in the system resulted in a fast enough unit for closed loop current control. The performance is comparable to analog design.

In the year 1985 [15] a novel fast response microprocessor based firing and control scheme for a phase controlled rectifier was developed, implemented and tested. With this technique the system responded within 20 micro seconds to a change in the desired output voltage. The controller is synchronized to the line through a software controlled phase locked loop. This allows the converter to adjust appropriately even with large variation of control parameters.

A microcomputer can also generate PWM signals for chopper or inverter controlled drives. In chopper control, the programmable pulse widths can be generated proportional to digital words through a hardware or software down counter. There may be several techniques for inverter PWM signal generation. Look up tables of digital words corresponding to pulse and notch width at different magnitudes of fundamental voltage can be stored in memory and then converted to a PWM pulse train with the help of down counters. The counter clock frequency is directly related to the fundamental frequency so that time intervals match the angular intervals. The look up tables may be based on selected harmonic elimination or minimum rms ripple current for a specified load. A computation-intensive method based on the comparison of sine modulating wave and triangular carrier wave is also possible.

So far the thyristor chopper control circuit have been composed of analog devices such as operational amplifiers, transistors, resistors and capacitors. In the year 1982 [16] digital control systems applying the microprocessors have been investigated to provide the compact, low cost, and maintenance-free thyristor chopper controllers. The first prototype controller was produced for the dual mode bus and the test results proved that the characteristics of the digital control was satisfactory and no harmful effect of noise was observed on the microprocessor. On the basis of this results the thyristor chopper controller applying a microprocessor for rolling stock had been developed.

A trigger circuit for a DC to DC thyristor chopper was presented in the year 1983 [17]. The basic objectives of investigation reported are to design and fabricate a fairly versatile and flexible microprocessor based firing circuit for a multiphase thyristor chopper and to compare its performance with that of the digitally controlled firing circuit. A close agreement between the current waveforms, the per unit ripple contents in the source as well as load currents and the machine torque-speed characteristics were observed and confirmed the successful implementation of the microprocessor based control.

Phase commutated static power converters of increasing power ratings are being used more and more in various manufacturing, processes, and transportation industries. In this context the problem is mainly harmonic distortion and low power factor. In the year 1981[18] microprocessor controller for a thyristor converter with an improved power factor was presented. The controller is used with a modified phase commutated converter which offers decreased harmonic distortion and improved power factor. Moreover, the control scheme and the microprocessor system presented in this work are directly applicable to a conventional six-thyristor bridge.

In the year 1984[19] microcomputer-based control of a residential photovoltaic power conditioning system was developed. The microcomputer is responsible for array current feedback control, maximum power tracking control, array safe zone steering control, phase locked reference wave synthesis, sequencing control, and some diagnostics. The control functions are implemented using Intel 8751 single-chip microcomputer-based hardware and software.

In 1978[20] several microprocessor based PWM techniques were investigated including a new one which minimizes distortion on an RL load. The optimum PWM technique is apparently dependent upon the fundamental output voltage and load conditions. The new PWM technique offers a method of adaptive inverter control which takes into account the load conditions.

The implementation of three phase sinusoidal pulse width modulated inverter control strategy using microprocessor was presented in 1982[21]. To save CPU time, the DMA technique was used for transferring the switching pattern from memory to the pulse amplifier and isolation circuits of individual thyristors in the inverter bridge. In this scheme neither a sine wave nor a triangular waveform was generated. The switching waves were directly generated by computation and stored in a memory and are applied to the thyristors through pulse isolation units in proper phase sequence. The memory requirement is minimum. Due to the use of DMA transfer method, the time used by CPU was saved substantially. When the CPU is not doing the calculation of switching pattern it can be used for feedback calculations.

Interest has been growing in microcomputer-based PWM schemes for ac drive systems in recent years. A high-performance Intel 8086 microcomputer-based pulse width modulator had been described in the year 1983[22]. The modulator operates on the computation intensive uniform sampling method in the low frequency region, whereas, the higher frequency region is based on a look up table.

1.4 Objectives of the Present Work

Real time implementation of PWM waveforms using microprocessors are based on solution of transcendental equations relating PWM switching angles and lacks generality when implemented by microcomputer. All implementation techniques reported so far used EPROM based look-up table using values of off-line basis calculation of switching points. The main disadvantage of ROM based look-up table is that it cannot provide smooth control over wide range of variations of control variable. On the other hand, most of these methods are either mathematically involved and computationally time consuming.

A newly developed technique for finding switching points opens avenues for easy microcomputer implementation of PWM switching of static converters having facility for on-line variation of modulation parameters. This method involves solution of simple algebraic equations requiring insignificant computation time. In this research the main objective is on-line computation of switching points of static converters according to sine pulse width modulation scheme and produce the waveforms by microcomputer to switch a three phase rectifier.

1.5 Outline of the Thesis

Chapter 1 covers the essentials of various modulation techniques and a brief literature review of microprocessor and microcomputer based works on static converters.

Chapter 2 is devoted to the description of the implementation procedures of the microcomputer controlled PWM rectifier. The implementation procedures cover the generation of PWM switching signals to switch a three phase MOSFET rectifier. Steps followed to generate PWM signals and microcomputer requirements are elaborately described in this chapter.

In chapter 3 a practical three phase microcomputer controlled PWM rectifier is described. Experimental results of the rectifier with resistive load are also provided.

Conclusions and suggestions for future work are outlined in chapter 4.

Chapter 2

Microcomputer Based Sine PWM Signal Generation For Three Phase Rectifier

2.1 Introduction

In the era of microcomputer control and automation it has been a constant effort of the researchers to implement various PWM switching strategies to control static power converters. For real time implementation of PWM waveforms using microprocessor, there are different techniques of realization depending on the type of modulation process. The implementation technique so far reported are based on transcendental equations relating PWM switching angles and lacks generality when implemented by microcomputer. The basic hardwares and softwares get modified depending on the technique of realization and the type of PWM waveforms. The main purpose of this thesis is to generate SPWM through microcomputer parallel port by using suitable software. Microcomputer implementation of SPWM for gating pulse generation for inverters has been reported in literature. The analysis and realization so far done considered the off-line calculation due to the inherent computational complexity. The objective of this research has been to investigate the microcomputer implementation of probable on-line gating pulse generation for a sine pulse width modulated rectifier. The features of microcomputer controlled SPWM rectifier operation are their flexibility, noise free operation having small low order harmonics at the intput and the output. Also the output voltage can be controlled by varying modulation parameters. Harmonic minimization can also be achieved in microcomputer based PWM control. High frequency carrier wave reduces output filter size of rectifiers. For controlled rectifier output a number of methods have so far been used and reported. All implementation techniques reported so far used ROM based look up table or OFF line calculation of switching points. The main disadvantage of ROM base look up table is that it cannot provide smooth control over wide range of variation of the control variable. For on-line gating pulse generation we have to calculate the on time and off time. To implement this we have to calculate the hexadecimal equivalent of on and off time and to free the processor for computation of new switching instants according to system requirement on user request.

2.2 SPWM Switching Point Calculation

SPWM is a popular technique of switching static PWM converters. This technique reduces low order harmonics and allows simultaneous voltage and frequency control of converters. High frequency carrier modulation also reduce filter size of converter applications. Triangular carrier is widely used modulation processes used in converters. In the triangular modulation, reference sine wave is modulated by a high frequency carrier triangular wave. There are many process such as synchronized and regular sampled sine PWM. Analytical solution for switching point determination has been the topic of interest for a long time. As a result, a number of methods evolved for solution of the switching points and their subsequent use in harmonic determination and performance predictions of converters. However, most of these methods are mathematically involved and computationally time consuming. In this chapter a new technique developed in a recent research for solution of switching points of triangular SPWM waveforms is presented [23]. The method involves solution of simple algebraic equations requiring insignificant computational time. This makes easier the implementation for microcomputer based PWM pattern generation on-line with facility of variation of modulation parameters.

2.2.1 Switching Points of Triangular Carrier Wave Modulated SPWM

There are many synthesis of switching points of triangular modulation. The simplest one reported so far is that of reference [23] in which following treatment was presented [Fig. 2.1].

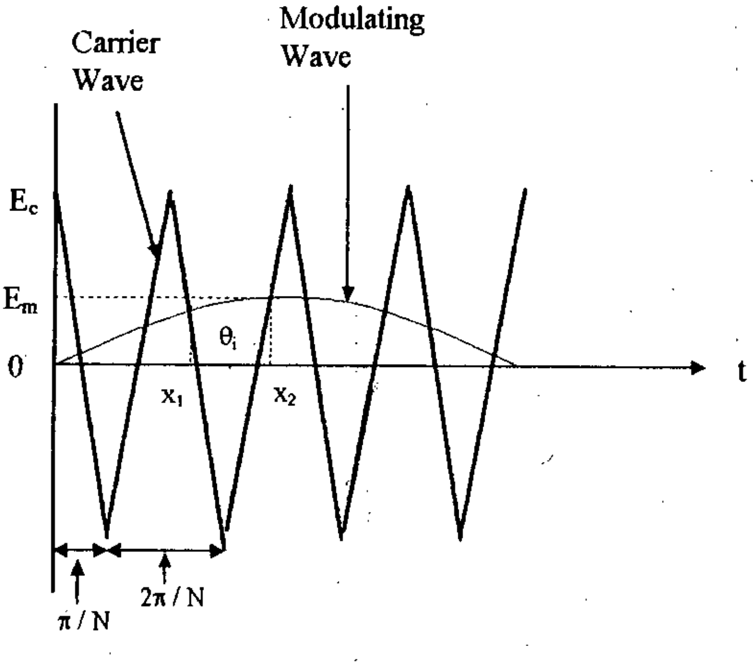


Figure 2.1 : Expanded view of triangular carrier modulation process of a sine wave

Assuming slope of the triangular wave to be s,
where,

$$s = 2E_c / (\pi / N) = (2NE_c) / \pi$$

E_c being magnitude of carrier wave, and
N is the ratio f_c / f_m (frequency of carrier wave to frequency of sine wave)

Position of i th pulse can be determined as,

Mid point of i th pulse θ_i is given as,

$$\theta_i = \frac{(2i-1)\pi}{N} \quad \text{...(2.1)}$$

Since width of i th pulse is directly proportional to $E_m \sin \theta_i$ (instantaneous value of modulating sine wave) and inversely proportional to the magnitude of carrier wave, the widths can be related as,

$$X_2 - X_1 = \delta_i$$

$$= K_i \frac{E_m}{E_c} \sin\theta_i$$

$$= K_i \, m \sin \theta_i \quad \text{... (2.2)}$$

where, K_i is the constant of proportionality and m is the modulation index.

From Figure 2.1, slope of rising edge (of triangular wave) can be related as,

$$s[x_2 - (\theta_i + \frac{\pi}{2N})] = E_m \sin x_2 \quad \text{..............................(2.3)}$$

and slope of falling edge (of triangular wave) can be related as,

$$-s[x_1 - (\theta_i - \frac{\pi}{2N})] = E_m \sin x_1 \quad \text{..............................(2.4)}$$

Combining equation 2.3 and 2.4 we have,

$$(x_2 - x_1) - \frac{\pi}{N} = \frac{E_m}{s}(\sin x_1 + \sin x_2)$$

or, $$\delta_i = \frac{E_m}{s}(\sin x_1 + \sin x_2) + \frac{\pi}{N} \quad \text{.............................(2.5)}$$

Since equation 2.2 and 2.5 represent the same thing, equating these two equations K_i can be extracted as,

$$K_i = \frac{E_m}{sm\sin\theta_i}(\sin x_1 + \sin x_2) + \frac{\pi}{Nm\sin\theta_i} \quad \text{.............(2.6)}$$

With fair degree of accuracy E_m ($\sin x_1 + \sin x_2$) can be approximated by $2E_m \sin\theta_i$ (this assumption is valid for regular sampled sine wave modulation where the reference signal is stepped sine wave).

Substituting E_m ($\sin x_1 + \sin x_2$) by $2E_m \sin\theta_i$, we have,

$$K_i = \frac{2E_m}{sm} + \frac{\pi}{Nm\sin\theta_i} \quad \text{..(2.7)}$$

where, values of

$$s = \frac{2NE_c}{\pi}$$

$$m = \frac{E_m}{E_c}$$

$$\theta_i = \frac{(2i-1)\pi}{N}$$

Once value of K_i is known, pulse widths can be determined using equation 2.2 as,

$$\delta_i = K_i \frac{E_m}{E_c} \sin\frac{(2i-1)\pi}{N}$$

However, pulse widths can directly be written without constant of proportionality from equation 2.5 as,

$$\begin{aligned}\delta_i &= \frac{E_m}{s}(\sin x_1 + \sin x_2) + \frac{\pi}{N} \\ &= \frac{2E_m}{s}\sin\theta_i + \frac{\pi}{N} \\ &= \frac{\pi}{N}(1 + \frac{2NE_m}{s\pi}\sin\theta_i) \\ &= \frac{\pi}{N}(1 + m\sin\theta_i)\end{aligned}$$

As the pulse widths are known, the switching points can easily be found as,

$$x_1 = (\theta_i - \frac{\delta_i}{2}) / 2\pi f_m$$

$$x_2 = (\theta_i + \frac{\delta_i}{2}) / 2\pi f_m$$

So by calculating X_1 and X_2 we get the values of t [1] and t [2] which will be used to generate sine pulse width modulated wave for switching three phase rectifier by a microcomputer.

2.3 Three Phase Rectifier Switching Pulse Requirement

A three phase rectifier is shown in figure 2.2:

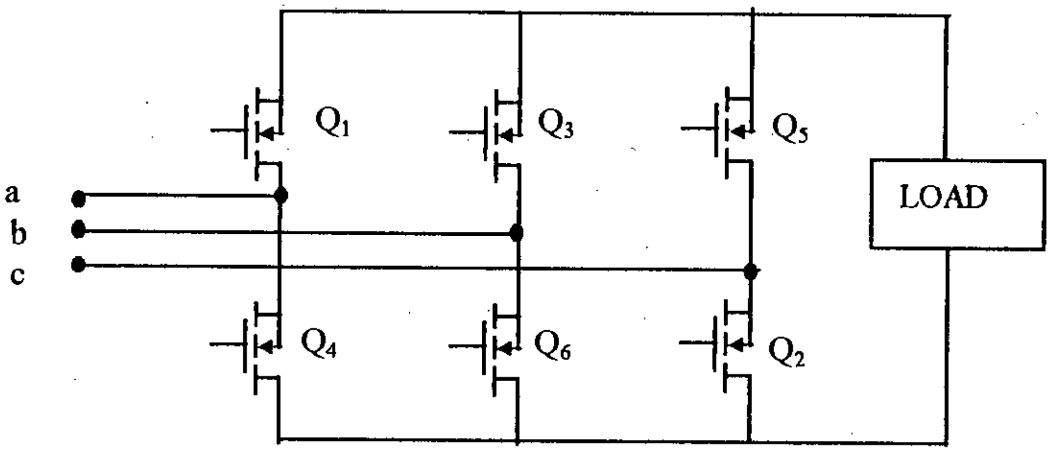


Figure 2.2 : Three phase MOSFET rectifier

If load current is to flow, then MOSFETs in at least two lines must be conducting. The factor determining which MOSFETs will conduct at any instant is the combination of the three source voltages V_{ab} , V_{bc} , V_{ca} which at that instant give the largest value of voltage. The combinations of the source voltages are of course simply the line to line voltages V_{ab} , V_{bc} , and V_{ca} and load voltages will reach a maximum value when any one of these three line to line voltages is at its positive or negative maximum. Let the line to line source voltages be,

$$V_{ab} = V_m \sin \omega t \qquad \text{V}$$

$$V_{bc} = V_m \sin(\omega t - \frac{2\pi}{3}) \quad \text{V}$$

$$V_{ca} = V_m \sin(\omega t - \frac{4\pi}{3}) \quad \text{V}$$

The gating signals of the MOSFETs in the three branches must have the same sequence and phase displacements as do the source voltages. Thus if the gating signals for Q_1 starts at $\omega t=0$, then that of Q_3 must be $2\pi/3$, and that of Q_5 must be $4\pi/3$. The gating pulse of Q_4 in line 'a' must start at π, that of Q_6 at $(2\pi/3)+\pi$, and that of Q_2 at $(4\pi/3)+\pi$. The resulting sequence of gating pulse is shown in Figure 2.3

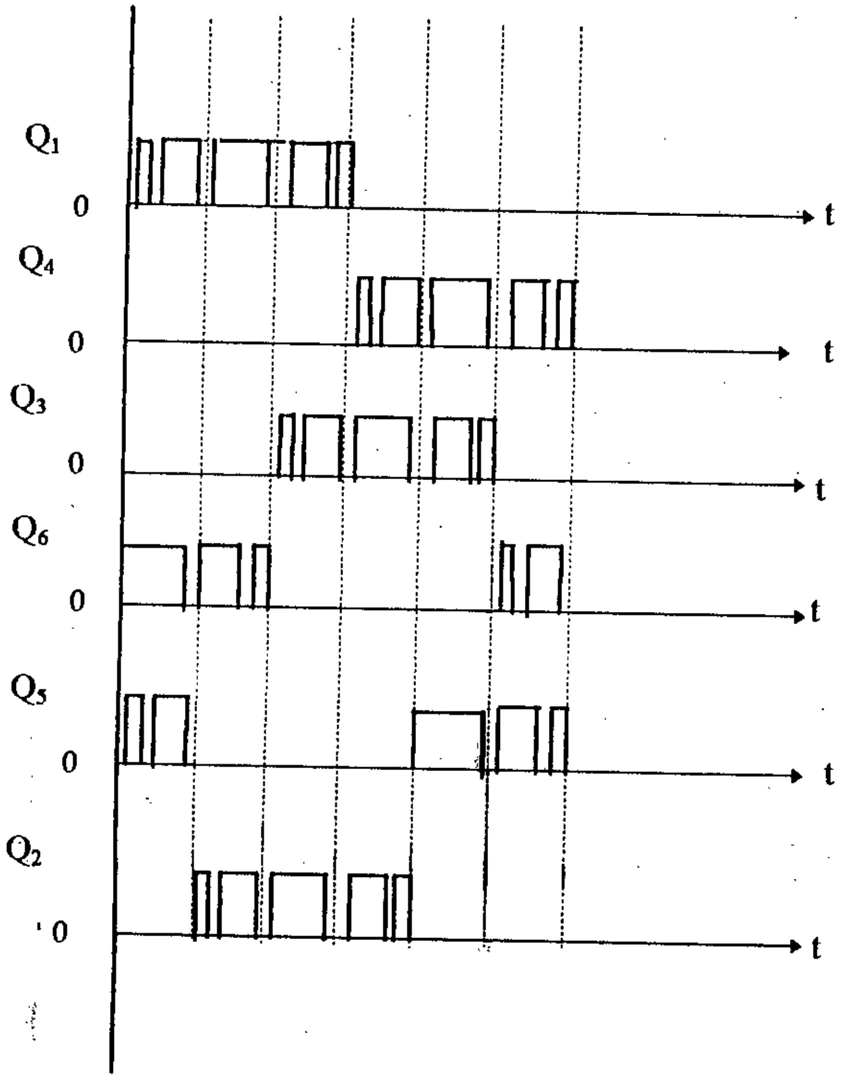


Figure 2.3 : PWM Gating signals for the MOSFETs

Starting instants for the switching devices are shown in the table 2.1.

Table 2.1

MOSFETs	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
Starting instant (in radian)	0	$\pi/3$	$2\pi/3$	π	$4\pi/3$	$5\pi/3$

It is evident that the gating pulses for the MOSFETs are displaced sequentially by $\pi/3$ from each other and this is necessary for three phase rectifier operation.

2.4 Microcomputer Facilities

A microcomputer is responsible for computation and decision making operations in a system. The heart of a microcomputer is the central processing unit (CPU) or microprocessor. In addition to the CPU, a microcomputer has the following elements:

- clock generator
- Address and data bus drivers (optional)
- Read only memory (ROM)
- Read/write or random access memory (RAM)
- Interrupt controller
- Digital input/output
- Analog input/output
- Serial and Parallel communication interface

A typical block diagram of a microcomputer system is shown in Fig 2.4.

2.4.1 CPU Architectur

The basic functional components of a CPU are shown in figure 2.5. The accumulator (ACC) registers the contents which are manipulated by the arithmetic logic unit (ALU). The CPU may contain additional general purpose registers (not shown) which will make data manipulation efficient and improve the processing speed. The ALU, as the name indicates, performs arithmetic and logic opeartions on binary data. The program counter (PC) stores the address of instructions to be executed. The CPU increments the PC each time it fetches a new instruction . The rule is overridden only when the CPU executes a jump instruction or performs a subroutine call. In a subroutine call, the PC is incremented and its content is stored in a memory area known as the 'stack'. The stack can also save the contents of other registers. The stack pointer (SP) maintains the address of the most recent stack entry. At 'return' after execution of the subroutine, the PC begins with the address retrieved from the SP. The instruction register (IR) stores the instruction or operation code fetched by the CPU. The instruction decoder translates this code to perform the specific instruction. The timing and control unit receives the clock signals and performs the CPU operation sequentially in an orderly manner.

2.4.2 ROM

A ROM (read only memory) device stores the program of a microcomputer. The program may contain data in the form of look-up tables or coefficients which cannot be altered. A program which does not need any alteration is stored in a mask-programmed ROM. In the initial stage of development, a program may require alteration, and therefore is stored in an electrically programmable, but ultra-violet light erasable memory (EPROM). An electrically programmable and electrically erasable memory (EEPROM) can be used for both read and write operations similar to RAM memory, but the information remains non-volatile.

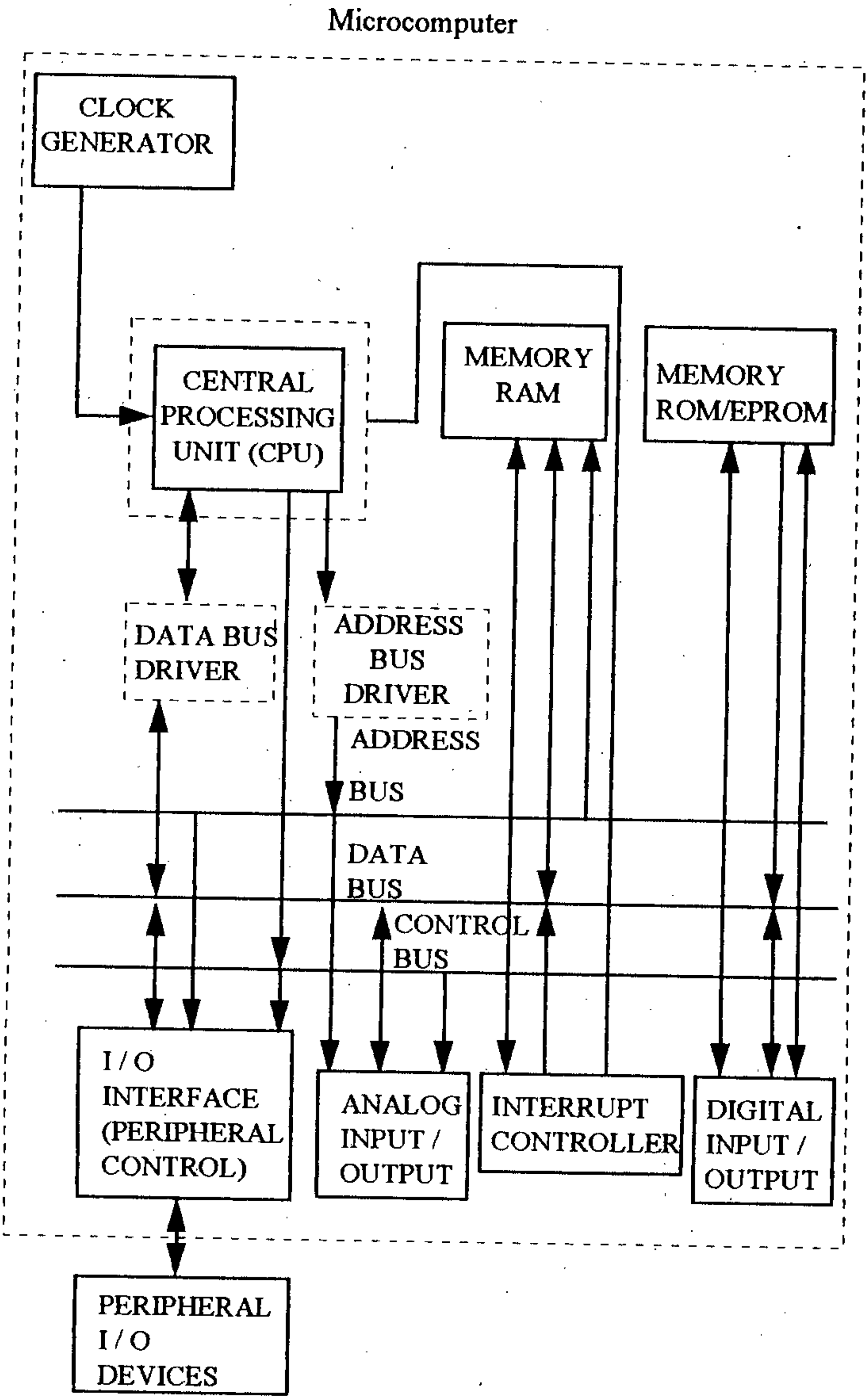


Fig 2.4 : Fundamental subsystems of a Microcomputer system

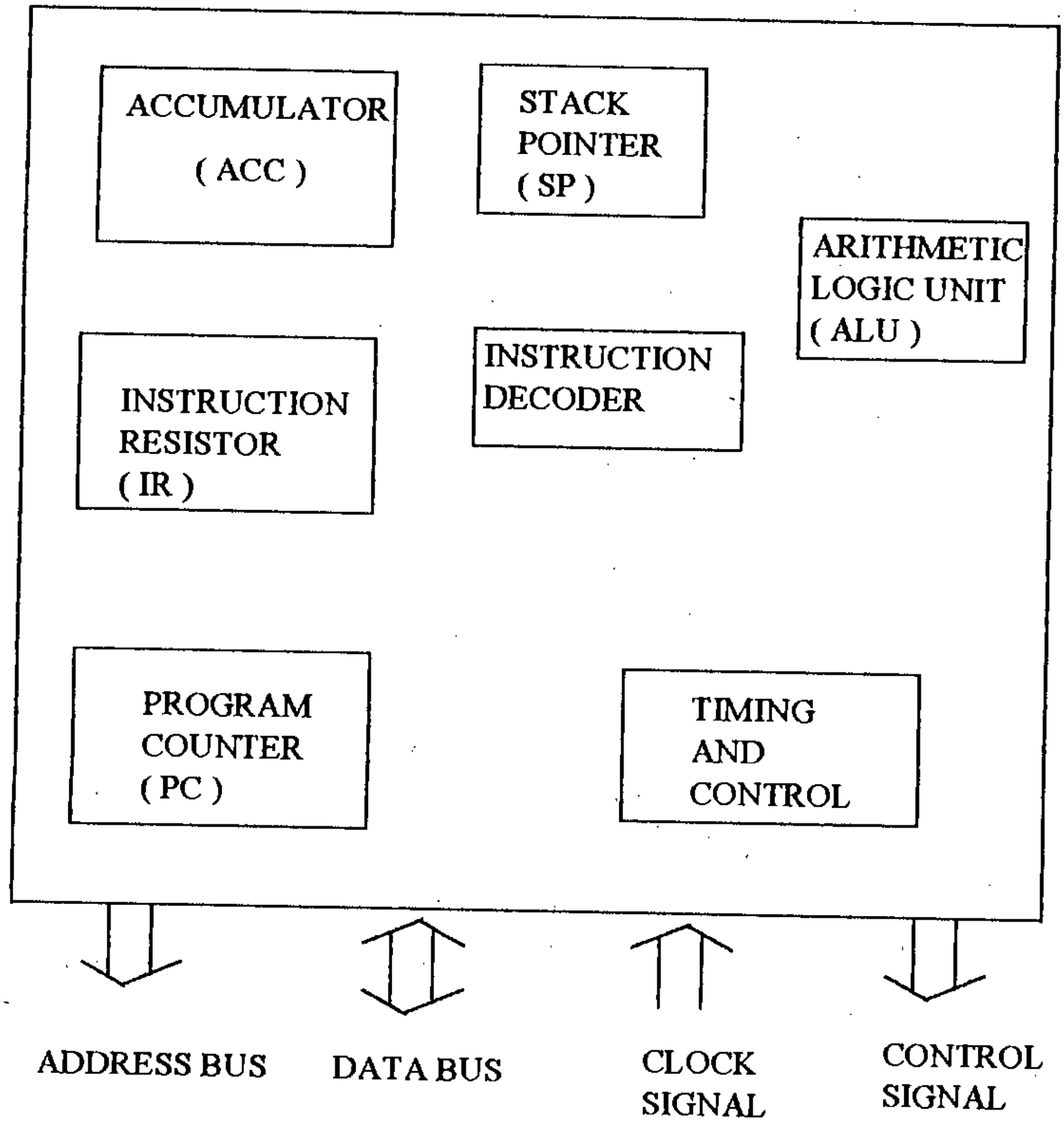


Figure 2.5 : The internal block diagram of a microcomputer central processing unit (CPU)

2.4.3 RAM

A RAM (random access memory) normally stores data which is generated by the program execution. In the initial stage of developement, a program can also be resident in a RAM . Both ROM and RAM have random access capability but

RAM is generally meant for READ / WRITE operation. A RAM has volatile storage. RAM memory may be static or dynamic.

2.4.4 Digital Input / Output

A microcomputer communicates with the outside world through peripheral I / O devices. The digital input / output signals may be in the form of individual logic or parallel byte / word form. The digital I / O signals can be controlled as if these are located in memory spaces or by simple IN/OUT instructions. The I/O ports have normally programmable features. The serial communication interface unit transmits or receives data in serial form, but internally, the microcomputer handles the data in parallel form.

2.4.5 Analog Input / Output

The analog signals of a physical system are interfaced to the microcomputer through analog-digital (A/D) and digital-analog (D/A) converters. The devices can be tied directly to microcomputer buses and can be mapped as memory locations. An analog input/output device can be unipolar or bipolar, and its bit-size determines the precision or resolution of the digital signal. The conversion time of an A/D converter should be short compared to the sampling interval within a microcomputer.

2.4.6 Interrupt Controller

The interrupt controller handles the interrupt request logic signals from external devices. When the CPU receives an interrupt signal, it suspends the normal sequence of operation, executes the interrupt subroutine, and then returns to the main program. An interrupt can be initiated internally either by software or by hardware, or externally by an outside system. When an interrupt signal is received, the CPU completes the current instruction, preserves the next address of the program counter in the stack, and attends to the interrupt service routine. An interrupt controller can handle a number of interrupt signals on a predefined priority basis in a nested manner. An interrupt signal can be maskable or nonmaskable. The complete interrupt system can be globally enabled or disabled.

A typical block diagram of a microcomputer is shown in figure 2.4. The CPU is connected to the other elements through the address, data and control buses. The width of the data bus normally determines the bit size of a computer. The address bus specifies the memory location, the control bus determines the operation from memory location, and the data bus bears data or instruction from the specified memory location. The computer interface contains adapters to support speaker, keyboard, display monitor, printer, and disk drives etc. The printer adapter or parallel adapter provides the parallel I/O port with a 25 pin D-shell connector. The printer is connected from printer adapter or parallel adapter of input / output (I/O) channel. The I/O channel supporting various adapter contains bi-directional data bus, address lines, interrupt, control lines for memory and I/O read or write, clock and timing lines, DMA control lines, memory refresh timing control lines, channel-check line, and power and ground for the adapters. Four voltage levels are provided for I/O cards: +5 volts, -5 volts, +12 volts, -12 volts dc. Of the connectors, the printer or paralel connector is the only parallel I/O port which is addressed as 378-37F in Hexadecimal range.

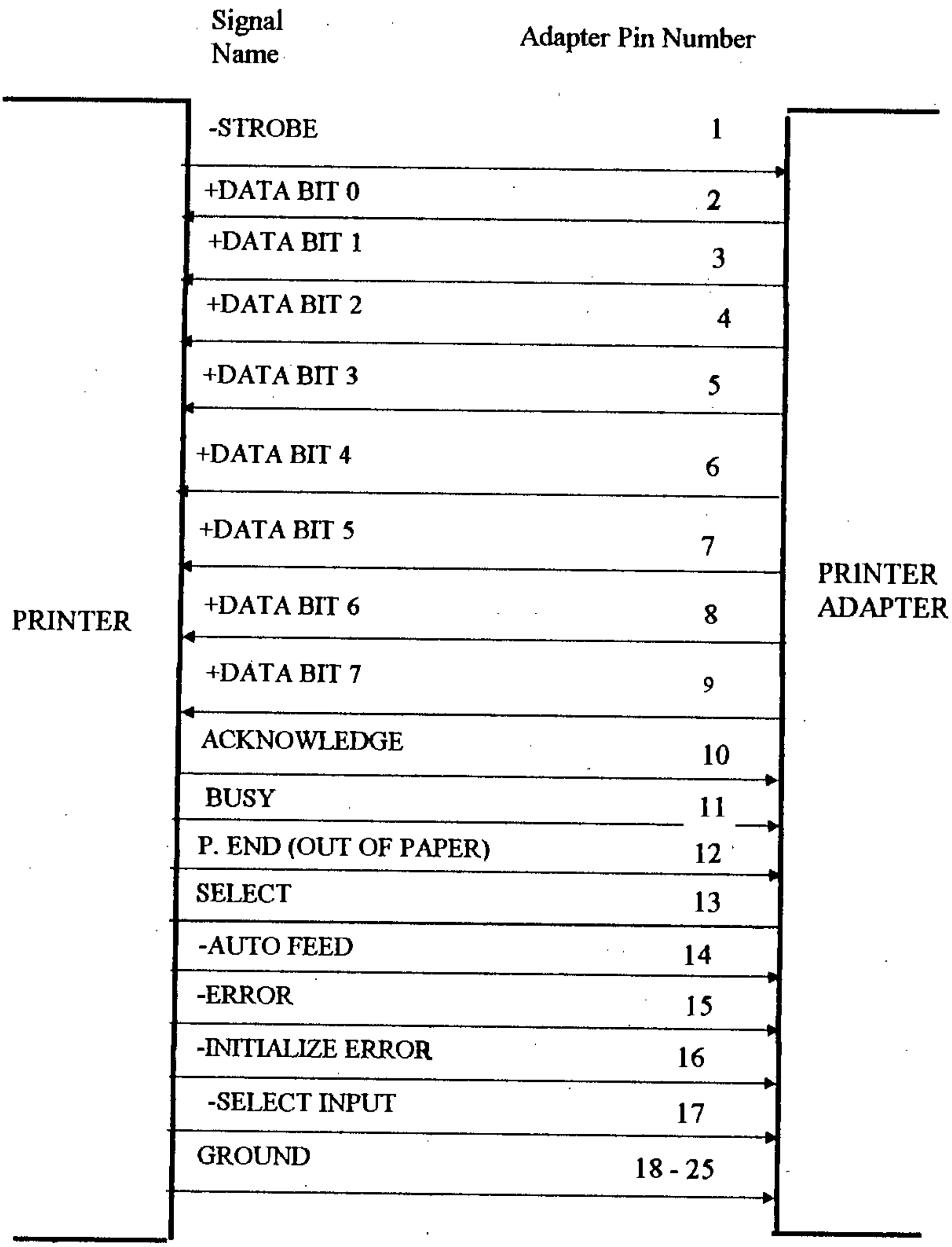


Figure 2.6 : Data-pin Specification of a 25-pin D-shell Parallel I / O Port

Here, 3 indicates parallel port 1. If we write 2 instead of 3, it will address parallel port 2. The simple communication ports i.e. parallel port and serial modem ports are non critical components in the personal computer machines and clones use a 25 pin D-shell female connector which is marked either parallel port or printer port. The parallel adapter or printer adapter makes possible the attachment of various devices that accept eight bits of parallel data at standard TTL levels. The block diagram of a printer adapter and connection specifications of a 25 pin D-shell connector is shown in figure 2.6. The SPWM switching pulses for a three phase rectifier system will be available at this connector.

2.5 Computer Requirements for SPWM Signal Generation

Generation of on-line microcomputer controlled SPWM switching pulse is implemented for a rectifier using a program in high level language (C++). Computer requirements with consideration of RAM size, storage capacity, ROM size, selection of language, speed of operation and instruction set and port availability and addresssability are discussed in the following sections:

2.5.1 RAM Size and Storage Capacity

Memory size of a computer means the RAM size required for proper accomodation of the program. In most of the cases estimation of memory size is difficult , and therefore it should be designed conservatively with provision for expansion. The assembly language program for a typical application can hardly exceed 4 kbytes, but the size will be higher in the case of high level language as used in this thesis. As the program is in C++ it will need about 5 Mbytes of hard disk storage for the compiler program. Besides, about 4.5 Mbytes of hard disk memory capability will be needed for the occupation of DOS operating environment where the soft-ware system works. Consequently, a total storage capacity of about 3.5 kbytes for the program compilation and about 57 kbytes of RAM size for program execution and 20 Mbytes of free hard disk space will be enough for the system developement in this thesis work.

2.5.2 ROM Size and Selection of Language

The program developed doesn't use any prerequisite data, hence no ROM size is needed besides the system ROM (system BIOS), key-board controller chip and the math co-processor chip.

The control of power electronic system are very time critical, and therefore assembly language is normally used which provides fast execution time. But a program developed in assembly language is time consuming, tedious, complex to interpret, and may require many iterations. On the other hand, with today's faster microcomputers, the amount of execution time saved by the assembly language other than high level language is not so important. Hence we have used high level-language (C++) to develop the software for our control system and thus made the program easy to develop, generous to perceive and compatible to other environment using microcomputer control.

2.5.3 Speed of Operation and Instruction Set

The most important cosideration for on-line microcomputer controlled PWM pulse generation is the speed of operation and instruction set. One of the reasons for which the speed of operation of the microcomputer to be used in the implementation is of utmost importance is that, a rectifier-fed drive require less than 20 µs of delay between two types of switching pulse patterns. That means when a new pattern of switching pulse are to be generated at the parallel port, on going pulse pattern at the same port must be replaced within a time of about less than 20 µs. Otherwise the drive connected will treat the rectifier as operating in off-line mode. This implies that on-line microconioputer controlled generation of PWM pulses will not allow any break of pulse availability at the parallel port longer than 20 μs during the calculation of timing instants and execution of other pre-generation steps.

The other and the most important reason, for which the speed of operation of the microcomputer bears remarkable consideration, is the reality of changing any modulation parameter in several steps to cope with the physical demands of the system stability and operation. The subfunction 'SPGDA ( )' of the main program performs the calculation of timing instants of the PWM pulses and the other pre-generation steps to enable the subfunction 'SPWM ( )' to generate the required pulse patterns. The subfunction 'SPGDA ( )' will take an execution time which depends on the values of modulation parameters f_m , f_c , and E_m as they determine the number of switching instants in a complete cycle.

To select a microcomputer for the implementation of on-line strategy of this thesis from the consideration of speed of operation, at first we have to define the range of variation of the modulation parameters to determine the maximum number of steps of variation to be performed to reach the final value from the initial value. The second thing is to know the maximum time allowed by the drive system performance within which the demanded pulse patterns are to be supplied.

2.6 Program Developement

We have developed a program to implement a sine pulse width modulated three-phase rectifier. The program named SFOR.CPP generates a pattern of six switching pulse at six individual pins of parallel port marked pin-2, pin-3, pin-4, pin-5, pin-6, and pin-7 for a three phase rectifier switching. In this case the pulses available at pin-2 and pin-5 can also be used for switching a single -phase rectifier. The program contains two subfunctions 'SPGDA ( )', and 'SPWM ( )' under the accomodation of main program. Detailed description and organization of the programs are outlined in following subsections.

2.6.1 Description of the Program 'SFOR.CPP':

The program SFOR.CPP, shown in Appendix A, contains two subfunctions 'SPGDA ( ), and "SPWM ( )', under the accomodation of main program. The program will be described as outlined in the Flowcharts 1-3. Initially all appropriate Header files for the library functions are to be included. It is noteworthy that TIMER.H and TIMER.CPP are included from classlib of the compiler. The 'Timer' is an instance class implementing a stopwatch. Then we have to define and initialize all the variables and the timer 'Ftimer' as per requirement as shown in Appendix A.

STEP-1

The progarm starts execution of the main body 'void main (void)' which includes an initial set of modulation parameters $E_m = 0.8$, $f_c = 550.0$ Hz, and $f_m = 50.0$ Hz. The execution time of a dummy do-while loop having a certain number of iterations is determined next. The execution time is recorded as 'del' by the stopwatch defined as 'Ftimer'.

STEP-2

The program calls the subfunction 'SPGDA ( )' to calculate the timing instants for PWM switching waveforms and to create an array deci[num_seg] which contains converted decimal numbers representing each and every time segment of time span 'del'. This array is used in the subfunction' to reproduce time segments to reveal the entire pulse pattern.

The part of the 'main ( )' uses the function 'kbhit ( )' which is a check for any recent keystroke made from the keyboard. i.e. 'kbhit ( )' is TRUE if any key is pressed and FALSE if no key is pressed.

STEP-3

In this stage the program checks the variable character 'state' using the function 'kbhit ( )'. If character 'q' is pressed on the keyboard the function 'getch ( )' makes the value of 'state'='q' and the 'main ( )' program quits the generation of pulses at the port and if not the following execution is made.

STEP-4

Again 'kbhit ( )' function is introduced to check the key pressed. In this stage it is desired that the user will strike-

'f' to change the value of 'f_m'

'c' to change the value of 'f_c', and

'e' to change the value of 'E_m'

One of the above character is assigned as 'state' by the function 'getch ( )' and for visual observation the monitor screen prints, 'I am ready to take new ['state']'. In this stage the value of 'z' becomes 0 (zero) and 'flag' becomes 1 (one) which was 0 (zero) before.

STEP-5

The keystroke is again checked with an additional check of 'flag'=1. In this stage it is desired that the user will strike a numeric character (i.e. from 0 to 9) or a decimal point '.' which will give a floating or integer number to be assigned as the new value of the 'state' selected in step-4. The character struck is assigned as 'ch' by the function 'getche ( )' and is echoed in monitor screen for visual validation. Another Function 'isdigit (ch)' is introduced to return non-0 if 'ch' is a digit, that is, 0 through 9, otherwise it returns 0. The program at this point checks the keystrike for 'ch' whether it is 0 to 9 or '.'. If 'TRUE', these digits are stored in the buffer memory 'val[z++]' one by one. The generation of pulses at the parallel port is not disturbed to perform the above work of storing as this is done by the computer in a time-sharing environment. When the new value is given by striking a key, we have to press any key other than 0 to 9 or '.'(e.g. the 'Enter' key). The storing process of buffer memory will then be terminated and the assigned buffer value 'val' is treated as new parameter $f_{m,}$, f_c, or E_m in accordance with the value of 'state'. As for example, if the 'state' is 'f' the buffer value 'val' will give the new value of f_m.

STEP-6

With new values of modulation parameter the 'main ( )' program again calls the subfunction 'SPGDA ( )' to recalculate the new timing instants t_i and refill the array deci[num_seg]. At this stage 'flag' is again made 0 (zero). The generation of pulses at the parallel port is not disturbed to perform the execution of the subfunction 'SPGDA ( )' as this is done by the computer in a time sharing environment.

STEP-7

Finally the program calls the subfunction 'SPWM ( )' which generates the pulse pattern as in the figure 2.3 at the prscribed pins using the new array deci[num_seg]. The new pulse pattern will replace the on going pattern at an instant only and only when all the pre-generation tasks are completed in a time-sharing environment of the CPU.

2.6.2 Subfunction 'SPGDA ( )'

The modulation parameters f_m , f_c , and E_m are the inputs of this subfunction. It prints 'running...' on the screen when execution starts.

STEP-1: Calculation of timing instants:

The subfunction will calculate the timing instants t[1], t[2], ..., t[n] for the PWM pulse pattern using the following equations.

N1=0 ; number of timing instants

E_c =1.0 ; amplitude of the carrier signal

$N=f_c/f_m$

am= E_m / E_c ; modulating index

$$\theta_i = \frac{(2i-1)\pi}{N}$$

$$\text{dev}[i] = \frac{\pi}{N}(1 + am\,sin\,\theta_i)\frac{1}{2\pi f_m}$$

$$\text{the}[i] = \theta_i / (2\pi\, f_m)$$

$$t[2i-1] = \text{the}[i] - \text{dev}[i]/2.0$$

$$t[2i] \quad = \text{the}[i] + \text{dev}[i]/2.0$$

calculation of timing instants will be continued up to the check of $t \geq \frac{\pi}{\omega_m}$ and the number of timing instants will be assigned to N1.

STEP-2 : Generation of the Array deci[num_seg]:

To explain the technique applied to scan the segment, let us consider the pulse pattern that should be available at the data pins of the parallel ports which is shown in figure [2.6]. The pulses are scanned at a definite time interval i.e. 'del'. So it will be a combination of logic-1(for positive voltage) and logic-0(for negative voltage). Now if we can send the converted decimal number of this binary 1/0 combination to the address 0x378 to be sustained for a time span of 'del', then the pattern of this time segment will be reproduced at the eight data pins. And similarly for the other segments we apply the same technique in a time sequential manner for which the pulse pattern will be generated in the total time period of t_m (t_m = num_seg x del) and this pattern of period t_m will be repeated again and again according to the on going or refill values of the array deci[num_seg]. As we scan the six individual pulses for a three-phase rectifier, the voltage of pin-8 and pin-9 will always be treated as negative (logic-0). From the figure 2.7 the conversion equation of the decimal numbers to create the array deci[num_seg] is

$$deci[r] = b_{r2}2^0 + b_{r3}2^1 + b_{r4}2^2 + b_{r5}2^3 + b_{r6}2^4 + b_{r7}2^5$$

where, r represent the segment number 0,1,2, ..., (num_seg-1)

b_{r2} represents the binary 1 or 0 for the element of segment no. r and pin-2 and so on for the other pins.

Here r2, r3, r4, ..., r7 are the element number.

Creation of deci[num_seg] is made easier by applying the technique that we do not require to sum up the decimal converted numbers of the elements which bears a logic-0. So we initially assign '0' to all the elements of the array deci[num_seg] by summing up the decimal converted values only for the elements where a logic-1 exists and the array will be formed as pin-wise instead of segment-wise executing the following formula,

$$deci[r] = deci[r] + 2^K$$

where,

K=0, for the pulse available at pin-2

K=1, for the pulse available at pin-3

K=2, for the pulse available at pin-4

K=3, for the pulse available at pin-5

K=4, for the pulse available at pin-6

K=5, for the pulse available at pin-7

At first we select a 'K' and with this 'K' (i.e. particular pin) the value 'r' is varied from 0 to (num_seg-1). And then r is varied for another K and so on for other pins. To produce PWM switching pulses with the timing instants calculated we make a logic-0 to be sustained from t[0] to t[1] and logic-1 to be sustained from t[1] to t[2] and so on upto t[n].

Six individual pulses have phase shift from one another in a three-phase rectifier system. The waveshapes of the switching pulses for every pin will be same but starts from different initial positions. The initial positions for the pulses are as follows:

sp=0.0 sec = 0^0, for K=0

sp=t_m / 6.0 sec = 60^0, for K=1

sp=t_m / 3.0 sec = 120^0, for K=2

sp=t_m / 2.0 sec = 180^0, for K=3

sp=2.0*t_m / 3.0 sec = 240^0, for K=4

sp=5.0*t_m / 6.0 sec = 300^0, for K=5

Suppose we are scanning segment number 10 and segment number 100. From the figure shown in figure 2.7 it is obvious what will be the decimal equivalent and binary equivalent of those segments. For segment number 10 and 100 parallel port will sustain 00000001 and 00011100 respectively for the time duration 'del'. So by executing different segments six complete PWM waveform will be available at the parallel port

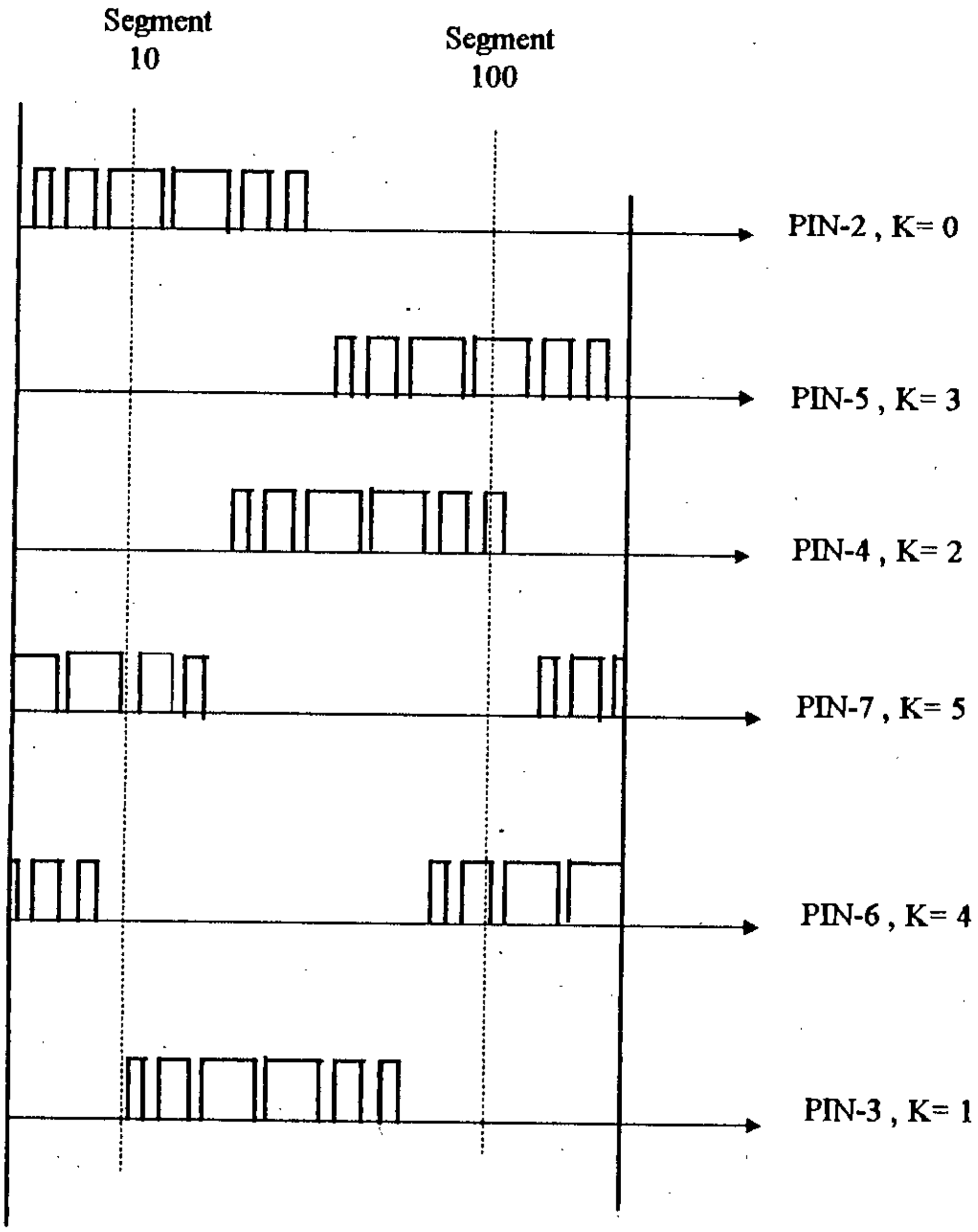


Figure 2.7 : SPWM pulse available at the parallel port

For segment number 10 :

$$\text{deci}[10] = 1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 0 \times 2^3 + 0 \times 2^4 + 0 \times 2^5$$
$$= 1$$

For segment number 100 :

$$\text{deci}[100] = 0 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 + 0 \times 2^5$$
$$= 28$$

2.6.3 Description of Subfunction 'SPWM ( )'

The input for this function is an array deci[num_seg] of decimal numbers each of which represents the required pulse level (1 or 0) at eight data pins to be sent as a byte for a time span of 'del'. At start, it executes the 'for loop' which starts from 0 and reaches (num_seg-1) with an increment of 1 at each step and it delivers the decimal numbers stored in deci[0], deci[1], ..., deci[num_seg-1] one by one. Each and every decimal number is sustained at the parallel port for a time span of 'del' by executing a dummy 'do-while' loop. The decimal numbers are sent to the address 0x378 using the following statement,

outport**b**(0x378, deci[r]) ; Here 'b' means byte

2.7 Typical SPWM Pulse Pattern

Typical pulse patterns for various modulation parameters generated by microcomputer to control three phase rectifier are shown in figure 2.8 to 2.15. Figure 2.8-2.11 shows the modulated switching pulse patterns of two of the six switching pulses for variation of carrier frequency of the modulation process. It is observed that number of pulses of the modulated waves increases with the increase of carrier frequency as expected and this would result in carrier frequency related harmonics to shift (higher the carrier frequency the shift will be towards the higher order). Figure 2.12-2.14 shows the effect of change in modulation index ($m = E_m / E_c$). The higher the modulation index is, the pulse widths increases, indicating that output voltage of the rectifier would increase with increasing modulation index. The controlled nature of the output of the rectifier will thus be obtained by the variation of the modulation index.

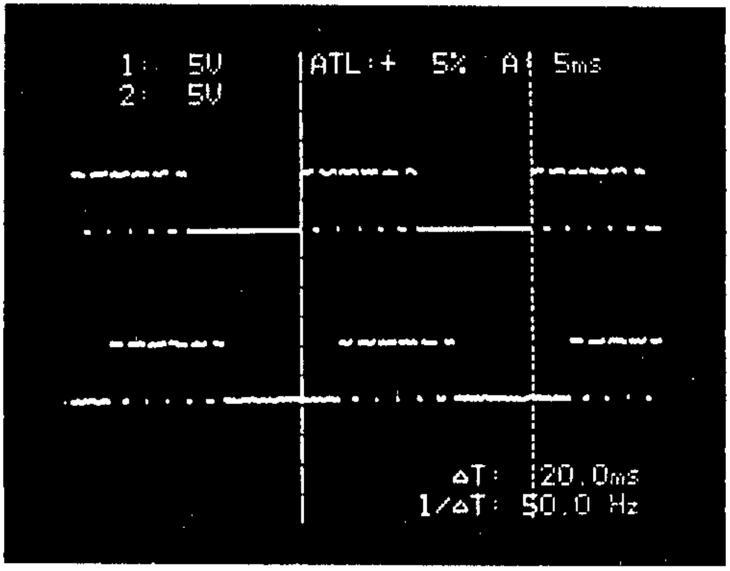


Figure 2.8 : E_m = 0.8, fc = 550 Hz, and f_m = 50 Hz

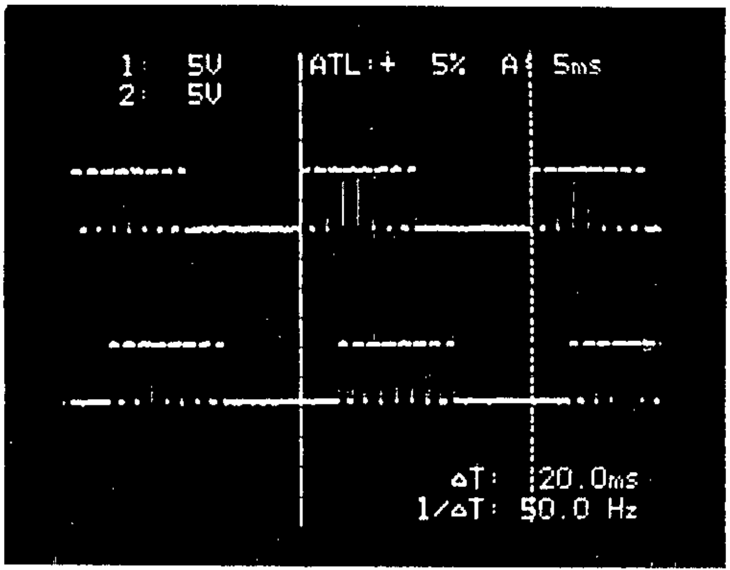


Figure 2.9 : E_m = 0.8, fc = 750 Hz, and f_m = 50 Hz

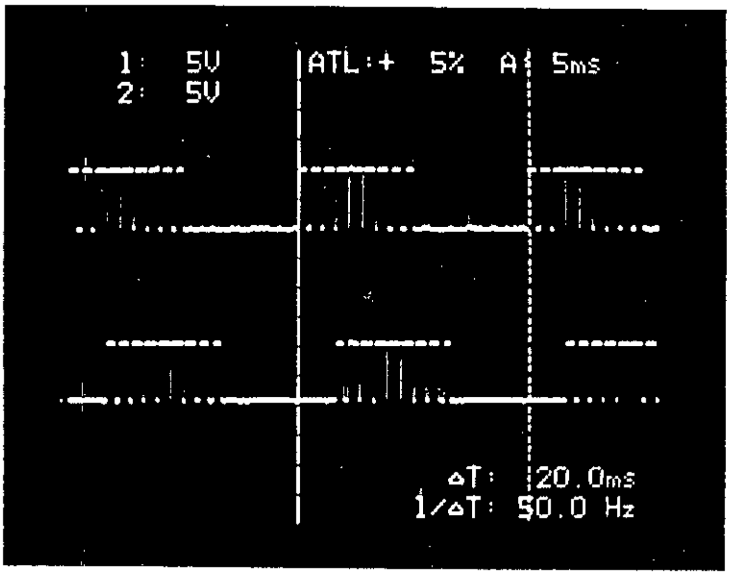


Figure 2.10 : $E_m = 0.8$, fc = 850 Hz, and $f_m = 50$ Hz

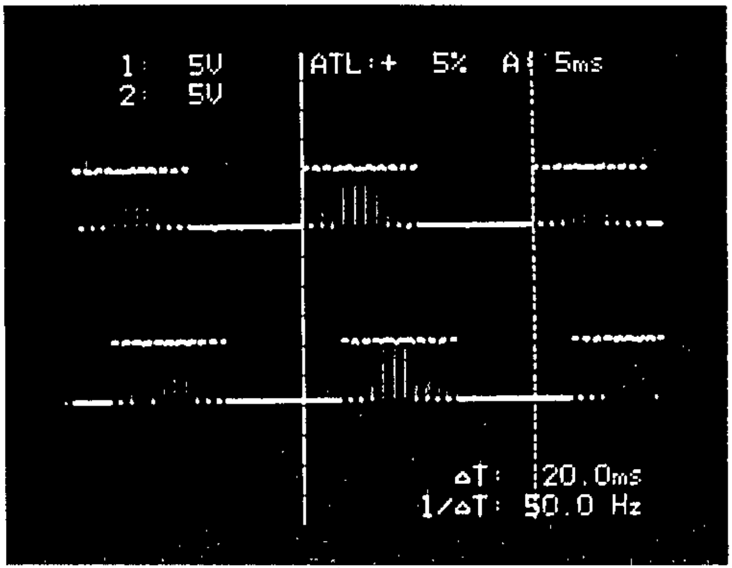


Figure 2.11 : $E_m = 0.8$, fc = 1050 Hz, and $f_m = 50$ Hz

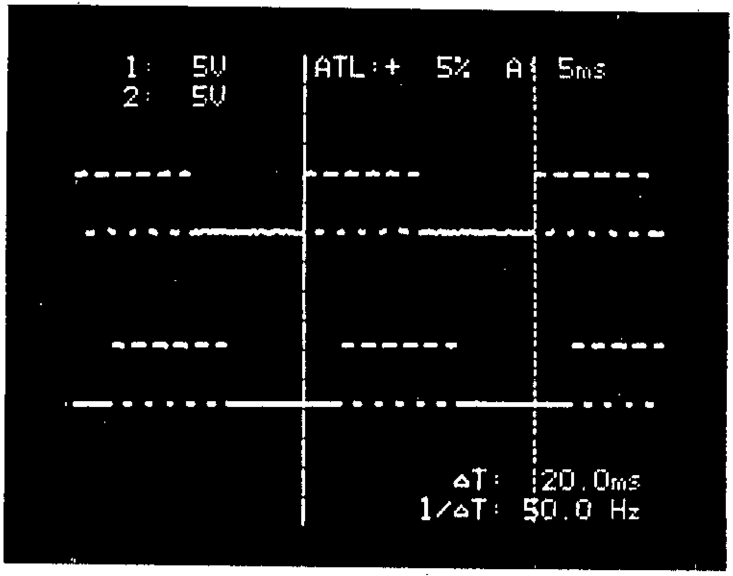


Figure 2.12 : $E_m = 0.4$, fc = 550 Hz, and $f_m = 50$ Hz

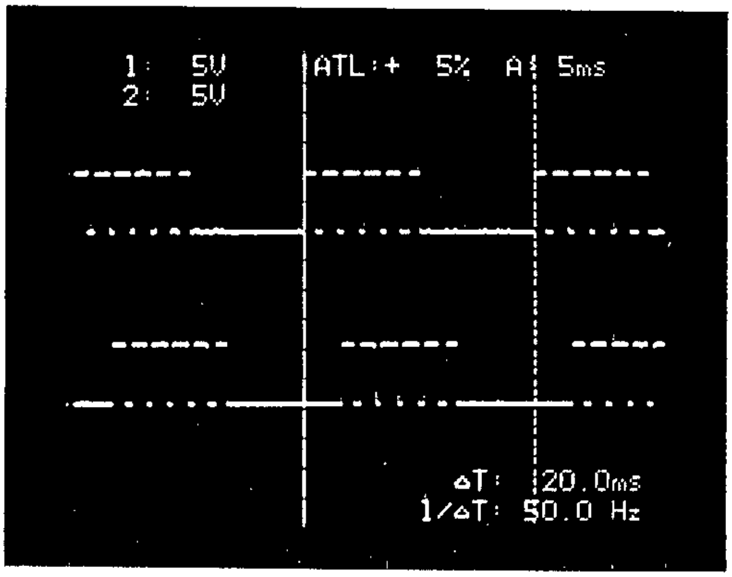


Figure 2.13 : $E_m = 0.6$, fc = 550 Hz, and $f_m = 50$ Hz

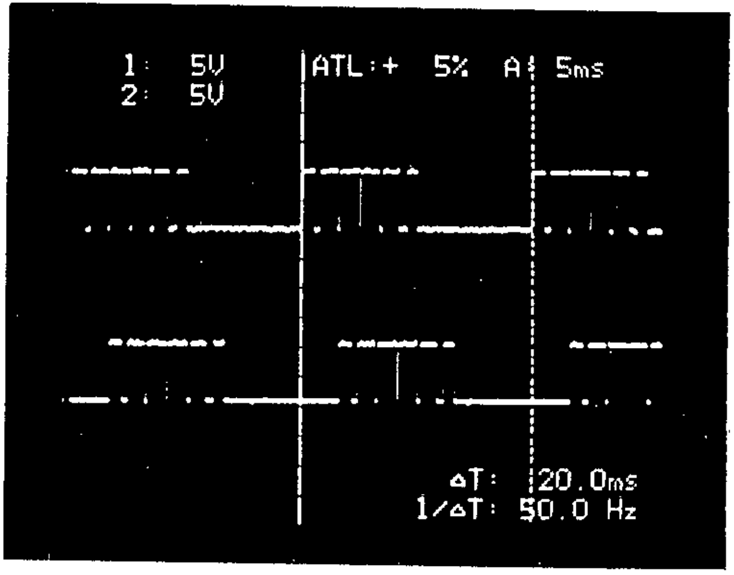


Figure 2.14 : $E_m = 0.9$, fc = 550 Hz, and $f_m = 50$ Hz

Switching points are given in tabular form in tables 2.2 to 2.8 for various modulation parameters as shown below.

Table 2.2 : $E_m = 0.8$, $f_c = 550$ Hz, and $f_m = 50$ Hz

	Switching Points	
	Theoretical	Practical
t_1	0.000352	0.000350
t_2	0.001466	0.001400
t_3	0.001998	0.001900
t_4	0.003457	0.003300
t_5	0.003731	0.003650
t_6	0.005360	0.005200
t_7	0.005578	0.005500
t_8	0.007149	0.006950
t_9	0.007531	0.007350
t_{10}	0.008833	0.008550
t_{11}	0.009545	0.009400

Table 2.3 : $E_m = 0.8$, $f_c = 750$ Hz, and $f_m = 50$ Hz

	Switching Points	
	Theoretical	Practical
t_1	0.000278	0.000300
t_2	0.001055	0.001050
t_3	0.001510	0.001500
t_4	0.002490	0.002450
t_5	0.002769	0.002750
t_6	0.003898	0.003850
t_7	0.004068	0.004100
t_8	0.005265	0.005200
t_9	0.005413	0.005400
t_{10}	0.006587	0.006450
t_{11}	0.006802	0.006800
t_{12}	0.007865	0.007750
t_{13}	0.008225	0.008200
t_{14}	0.009108	0.009050
t_{15}	0.009667	0.009650

Table 2.4 : $E_m = 0.4$, $f_c = 550$ Hz, $f_m = 50$ Hz

	Switching Points	
	Theoretical	Practical
t_1	0.000403	0.000400
t_2	0.001415	0.001450
t_3	0.002135	0.002100
t_4	0.003319	0.003300
t_5	0.003911	0.003800
t_6	0.005180	0.005100
t_7	0.005744	0.005700
t_8	0.006984	0.006950
t_9	0.007629	0.007550
t_{10}	0.008735	0.008750
t_{11}	0.009545	0.009550

Table 2.5 : $E_m = 0.6$, $f_c = 550$ Hz, $f_m = 50$ Hz

	Switching Points	
	Theoretical	Practical
t_1	0.000378	0.000350
t_2	0.001440	0.001450
t_3	0.002067	0.002050
t_4	0.003388	0.003250
t_5	0.003821	0.003650
t_6	0.005270	0.005250
t_7	0.005661	0.005650
t_8	0.007066	0.006950
t_9	0.007580	0.007500
t_{10}	0.008784	0.008700
t_{11}	0.009545	0.009500

Table 2.6: $E_m = 0.9$, $f_c = 550$ Hz, $f_m = 50$ Hz

	Switching Points	
	Theoretical	Practical
t_1	0.000339	0.000350
t_2	0.001479	0.001400
t_3	0.001964	0.001900
t_4	0.003491	0.003400
t_5	0.003686	0.003650
t_6	0.005405	0.005350
t_7	0.005537	0.005500
t_8	0.007190	0.007000
t_9	0.007506	0.007350
t_{10}	0.008858	0.008800
t_{11}	0.009545	0.009500

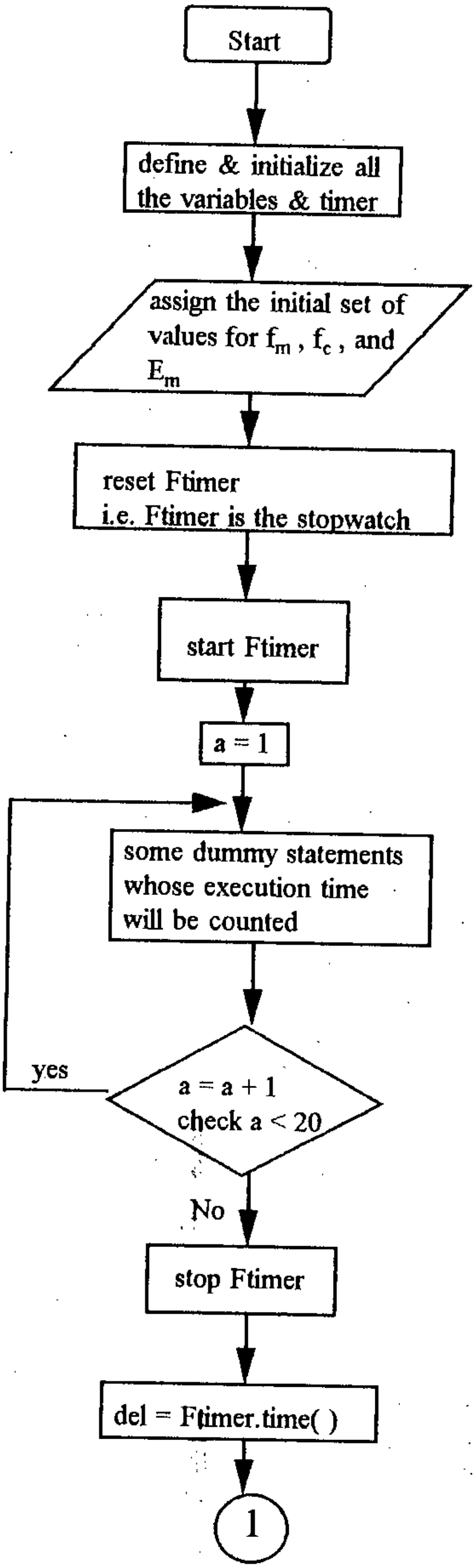
Table 2.7: $E_m = 0.8$, $f_c = 850$ Hz, $f_m = 50$ Hz

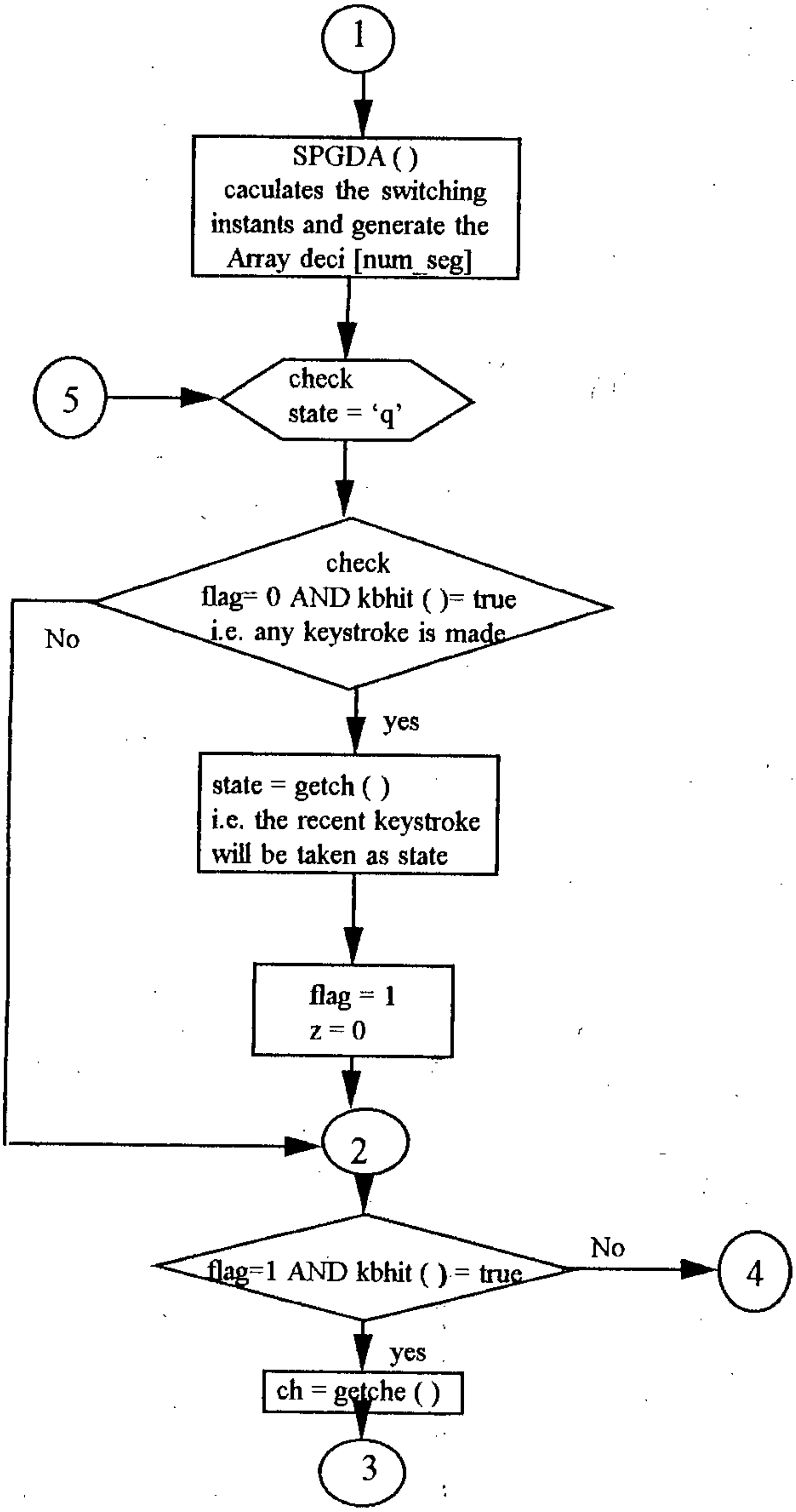
	Switching Points	
	Theoretical	Practical
t_1	0.000257	0.000250
t_2	0.000926	0.000950
t_3	0.001347	0.001350
t_4	0.002183	0.002200
t_5	0.002454	0.002450
t_6	0.003423	0.003350
t_7	0.003547	0.003650
t_8	0.004638	0.004500
t_9	0.004766	0.004700
t_{10}	0.005823	0.005700
t_{11}	0.005966	0.005950
t_{12}	0.006975	0.006800
t_{13}	0.007194	0.007100
t_{14}	0.008100	0.008000
t_{15}	0.008444	0.008400
t_{16}	0.009203	0.009100
t_{17}	0.009706	0.009600

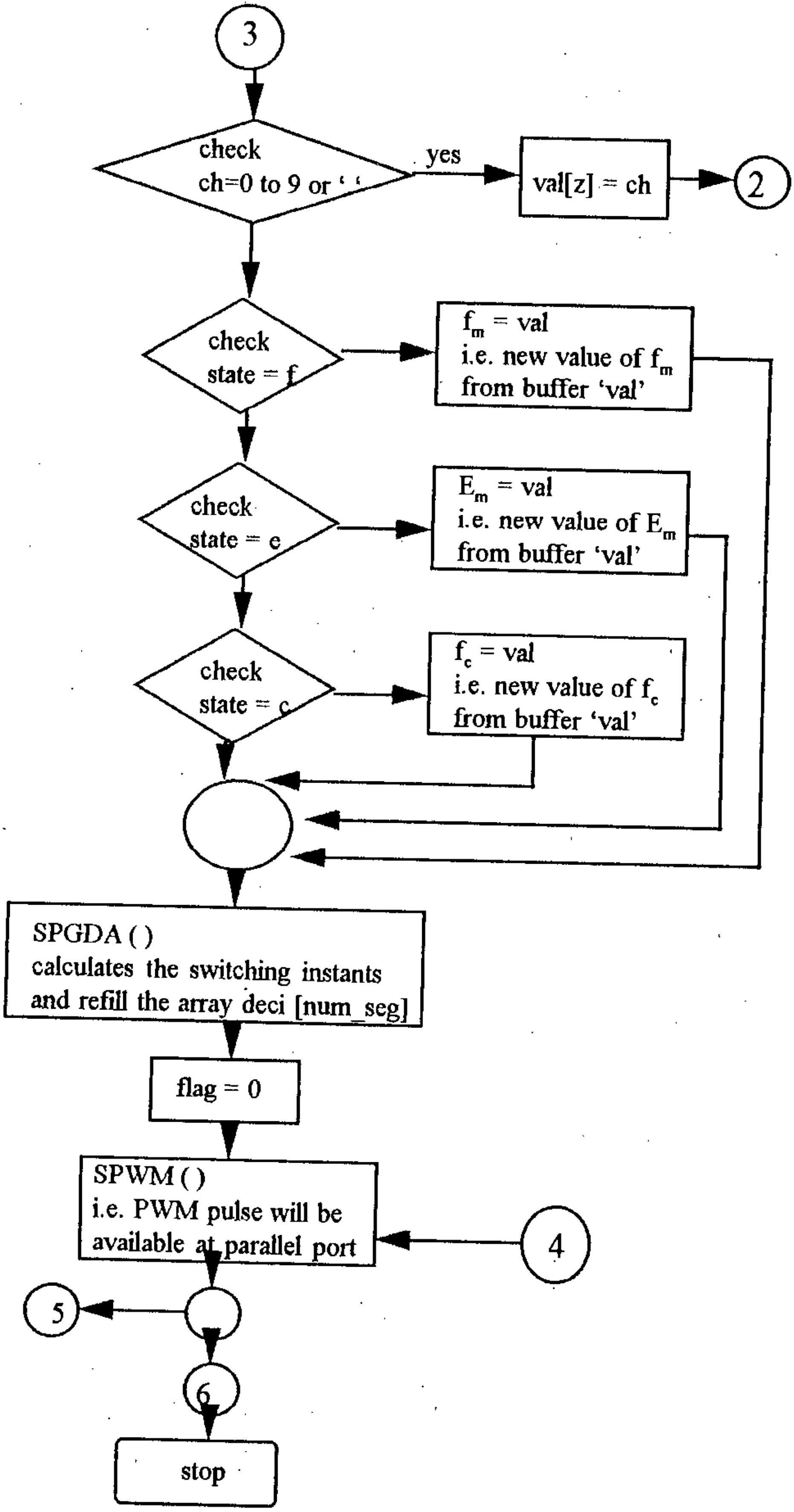
Table 2.8: $E_m = 0.8$, $f_c = 1050$ Hz, $f_m = 50$ Hz

	Switching Points	
	Theoretical	Practical
t_1	0.000210	0.000200
t_2	0.000743	0.000750
t_3	0.001108	0.001100
t_4	0.001749	0.001800
t_5	0.002013	0.002050
t_6	0.002749	0.002750
t_7	0.002930	0.002950
t_8	0.003736	0.003700
t_9	0.003862	0.003900
t_{10}	0.004710	0.004750
t_{11}	0.004810	0.004900
t_{12}	0.005666	0.005550
t_{13}	0.005775	0.005700
t_{14}	0.006606	0.006550
t_{15}	0.006756	0.006700
t_{16}	0.007530	0.007400
t_{17}	0.007750	0.007700
t_{18}	0.008441	0.008350
t_{19}	0.008753	0.008650
t_{20}	0.009342	0.009300
t_{21}	0.009762	0.009700

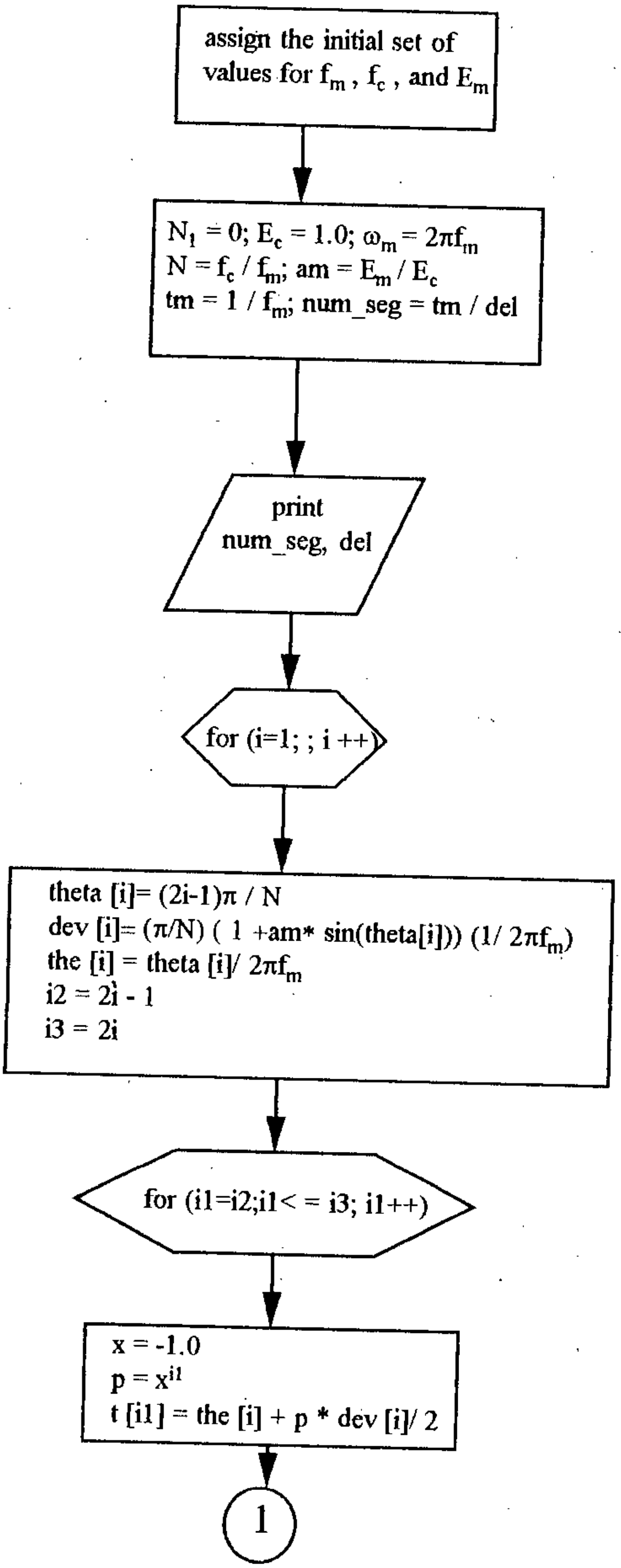
Flowchart-1

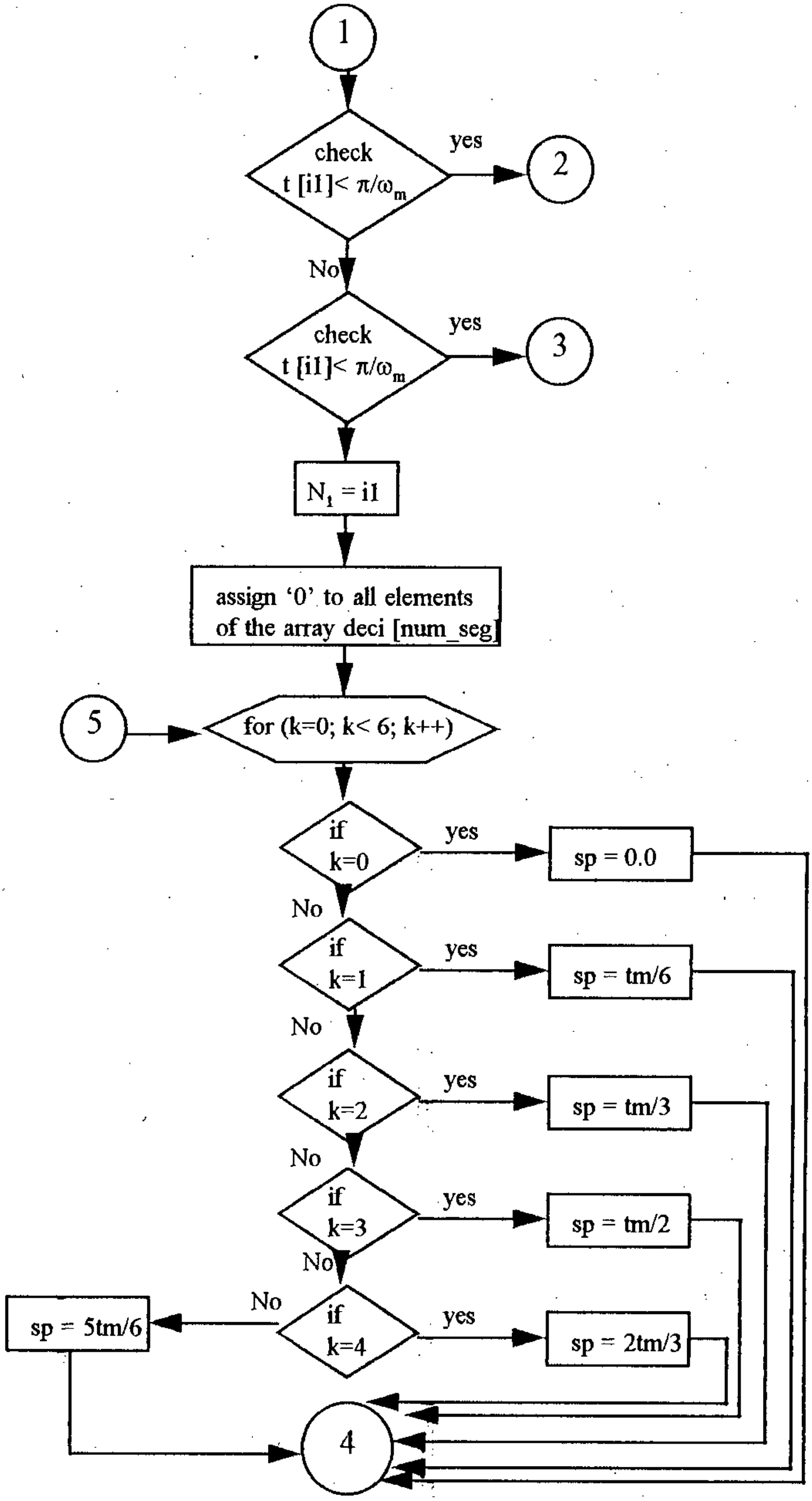

Start
define & initialize all the variables & timer
assign the initial set of values for f_m , f_c , and E_m
reset Ftimer
i.e. Ftimer is the stopwatch
start Ftimer
a = 1
some dummy statements whose execution time will be counted
yes
a = a + 1
check a < 20
No
stop Ftimer
del = Ftimer.time( )
1

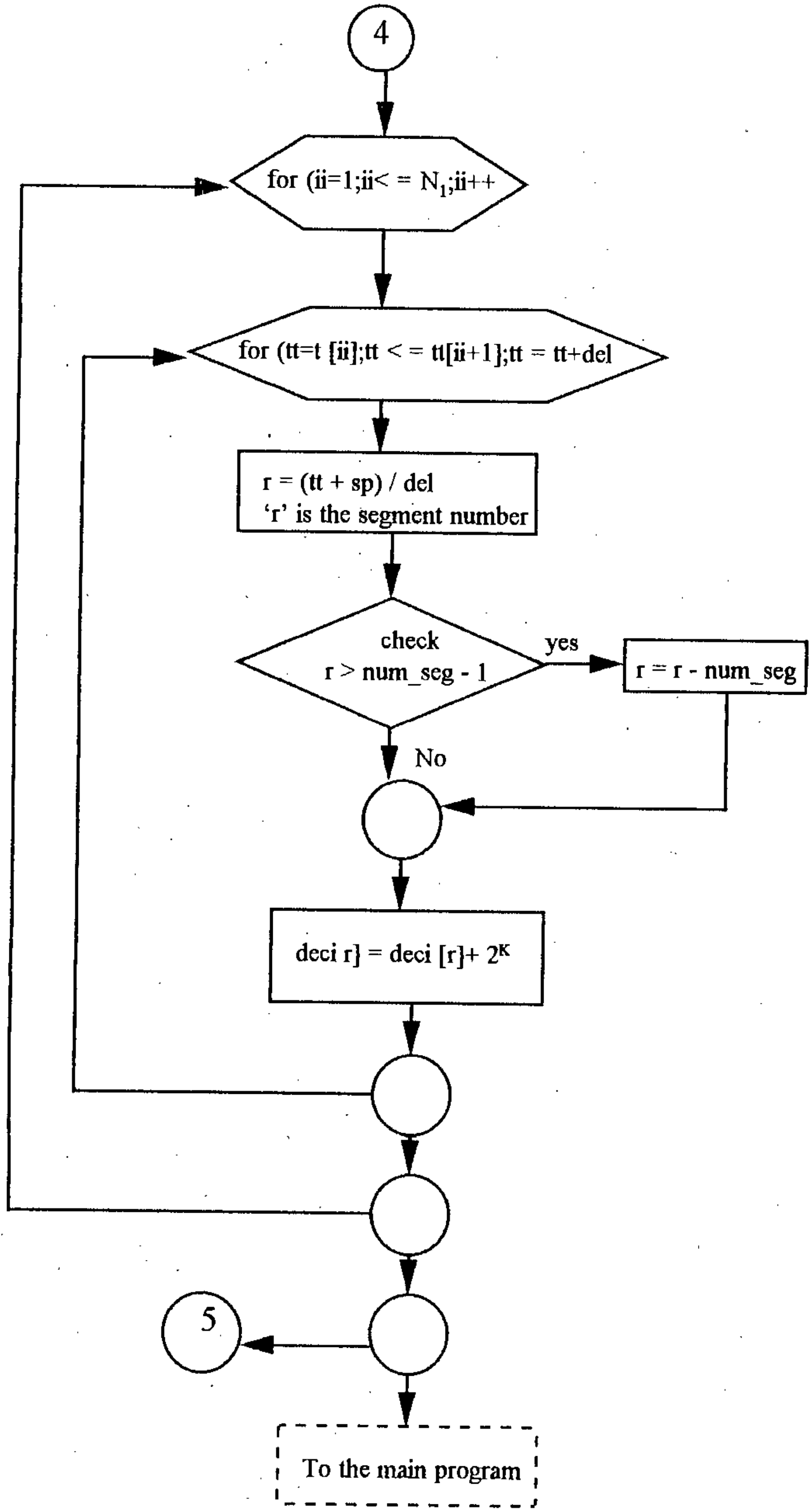

1
SPGDA ( )
caculates the switching instants and generate the Array deci [num_seg]
5
check
state = 'q'
check
flag= 0 AND kbhit ( )= true
i.e. any keystroke is made
No
yes
state = getch ( )
i.e. the recent keystroke will be taken as state
flag = 1
z = 0
2
flag=1 AND kbhit ( ) = true
No
4
yes
ch = getche ( )
3


3
check
ch=0 to 9 or ' '
yes
val[z] = ch
2
check
state = f
f_m = val
i.e. new value of f_m
from buffer 'val'
check
state = e
E_m = val
i.e. new value of E_m
from buffer 'val'
check
state = c
f_c = val
i.e. new value of f_c
from buffer 'val'
SPGDA ( )
calculates the switching instants
and refill the array deci [num_seg]
flag = 0
SPWM ( )
i.e. PWM pulse will be
available at parallel port
4
5
6
stop

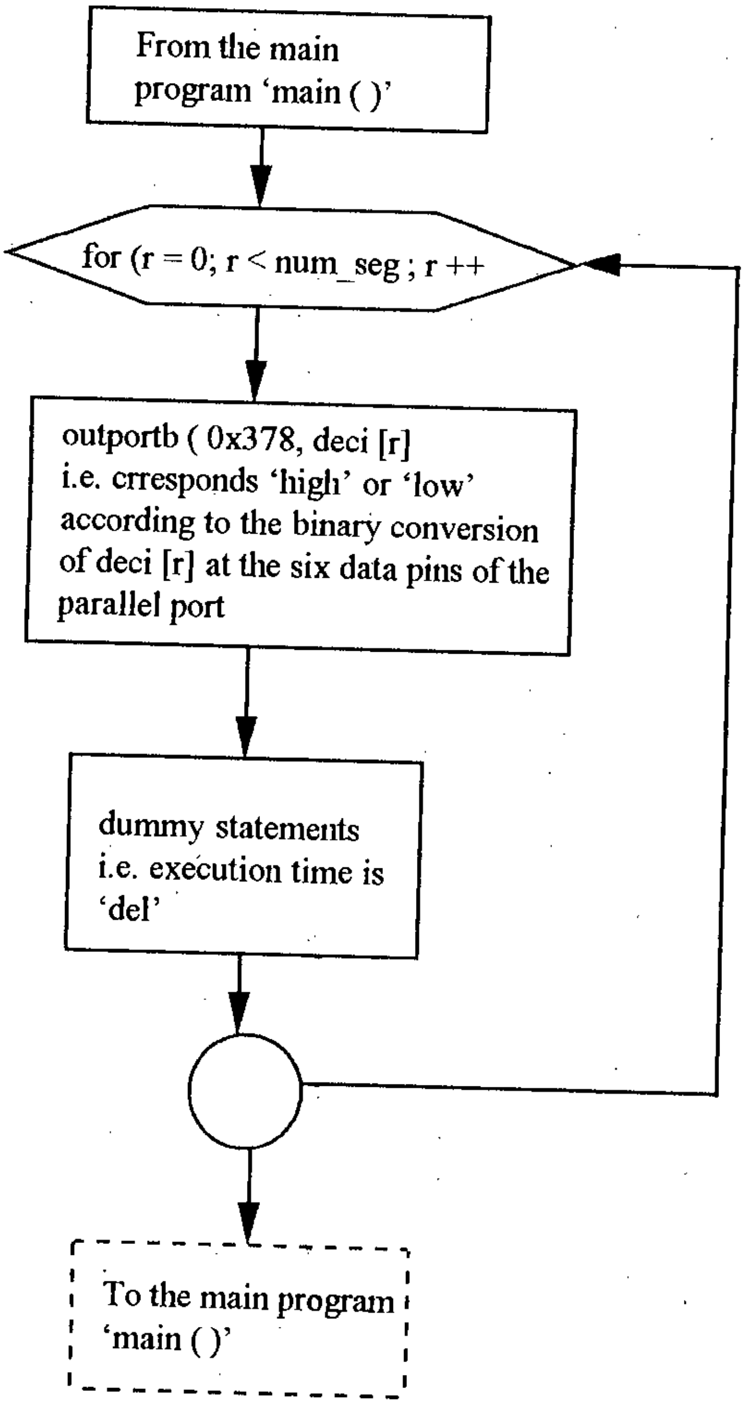
Flowchart 2


assign the initial set of values for f_m , f_c , and E_m
$N_1 = 0$; $E_c = 1.0$; $\omega_m = 2\pi f_m$
$N = f_c / f_m$; am = E_m / E_c
tm = $1 / f_m$; num_seg = tm / del
print num_seg, del
for (i=1; ; i ++)
theta [i]= (2i-1)π / N
dev [i]= (π/N) (1 +am* sin(theta[i])) (1/ $2\pi f_m$)
the [i] = theta [i]/ $2\pi f_m$
i2 = 2i - 1
i3 = 2i
for (i1=i2;i1< = i3; i1++)
x = -1.0
p = x^{i1}
t [i1] = the [i] + p * dev [i]/ 2
1


1
check
t [i1]< π/ω_m
yes
2
No
check
t [i1]< π/ω_m
yes
3
N_1 = i1
assign '0' to all elements of the array deci [num_seg]
5
for (k=0; k< 6; k++)
if k=0
yes
sp = 0.0
No
if k=1
yes
sp = tm/6
No
if k=2
yes
sp = tm/3
No
if k=3
yes
sp = tm/2
No
if k=4
yes
sp = 2tm/3
No
sp = 5tm/6
4


4
for (ii=1;ii< = N1;ii++
for (tt=t [ii];tt < = tt[ii+1];tt = tt+del
r = (tt + sp) / del
'r' is the segment number
check
r > num_seg - 1
yes
r = r - num_seg
No
deci r] = deci [r]+ 2K
5
To the main program

Flowchart 3


From the main program 'main ( )'
for (r = 0; r < num_seg ; r ++
outportb (0x378, deci [r]
i.e. crresponds 'high' or 'low' according to the binary conversion of deci [r] at the six data pins of the parallel port
dummy statements
i.e. execution time is 'del'
To the main program 'main ( )'

Chapter 3

Practical Results
Of A
Microcomputer Controlled
Three Phase Rectifier

3.1 Introduction

Rectification is the process of changing alternating current to direct current. To effect rectification one of the following methods can be used,

- Rotating converter and
- Semiconductor rectifiers

Rectification by semiconductor devices offers several important advantages compared with rotating converters. These are,

- No foundation is required,
- The apparatus has no moving parts,
- Maintenance is simpler and operation is noiseless and without vibration,
- The forward voltage drop across the semiconductor devices is smaller providing higher efficiency, and

- Solid state rectifiers are used extensively in low power radio and electronic equipments, Polyphase rectifier are preferable when large amount of direct current (dc) power is required. Polyphase rectifiers produce less ripple voltage superimposed on the dc output voltage. The efficiency of a polyphase rectifier is higher because it produces less wasted ac power in the dc operated load. Application of controlled rectifiers include the following :

- dc motor speed control system, widely used in steel mills, paper mills, etc.
- Electrochemical and electrometallurgical process,
- Magnet power supplies,
- Converters at the input end of dc transmission lines, and
- Portable hand tool drives etc.

In this thesis implementation of a microcomputer controlled three phase sine pulse width modulated rectifier is undertaken. In the previous chapter (chapter 2) successful generation of PWM switching pulses for a three phase controlled rectifier has been described. The switching pulses are used in the practical operation of a three phase power MOSFET rectifier with proper drive circuit and isolation. The experimental details and typical results obtained are discussed in the following section. As will be evident from the discussions, the switching patterns obtained for the three phase rectifier can also be used for the operation of a PWM single phase controlled rectifier if desired.

3.2 Necessity of Synchronized Switching

In SPWM, the mean output voltage is controlled according to a sine function. This is achieved by the repeated switching 'ON' and 'OFF' of the particular MOSFETs before the control is passed to the other MOSFETs. In this case, the pulse width is varied in a sine function in a half period as shown in Figure 3.1. The PWM waveform is high (when the sinusoidal voltage is greater than the triangular voltage) which makes the switching device 'ON' for that duration. The pulse widths are not constant throughout the half period. The widths are long at the middle region of the sine wave and gets shorter at the two zero

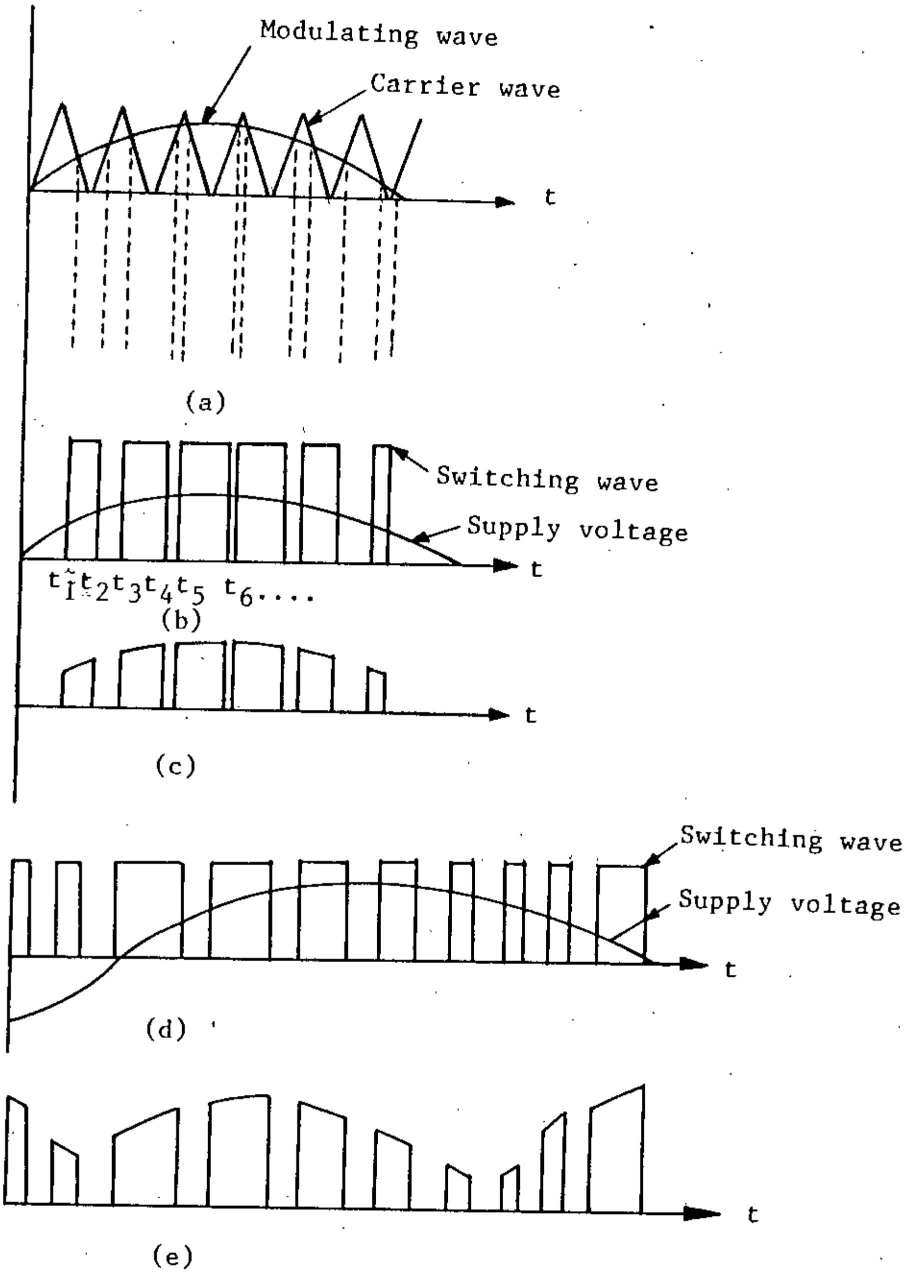


Figure 3.1 : Synchronized and Unsynchronized Rectifier

a. Modulating Sine Wave and Carrier Triangular Wave

b. Modulated Wave for Synchronized Switching

c. Output Wave for Synchronized Switching Single Phase Rectifier

d. Modulated Wave for Unsynchronized Switching

e. Output of Single Phase Rectifier for Non-synchronized Switching crossing area of the sinusoidal wave. In order to cope with this type of gating pulse, the gating signals must be synchronized with the input ac signal. If it is not synchronized then the switching devices will not be turned 'ON' or 'OFF' as predicted and desirable output of the controlled rectifier will not be achieved. If the input sine wave and the corresponding gating signals are not synchronized it may so happen that output of the rectifier will be low when it is expected to be high with large modulation index.

From the above figure 3.1 it is evident that why synchronism is necessary. It is evident that within the interval t_1 to t_2 or t_3 to t_4 the gating signal is high, consequently during this interval the switching devices are 'ON'. So if it is not synchronized then it will be mismatched i.e. it will not be 'ON' at the specified 'ON' time span. So synchronization is a must where input signal is an alternating quantity.

3.3 Practical Rectifier Circuit With Isolation

The rectifier section of the microcomputer controlled SPWM rectifier requires the design steps mentioned in the following sections. This article includes the design steps of practical isolation circuits between the microcomputer producing six individual PWM pulses and the MOSFET of the rectifier.

3.3.1 Computer Interface

1. 14-Pin Hex Inverter / Buffers

This is the first step of interfacing of the PWM pulses available at the parallel port of the microcomputer. IC 7404 Hex inverter / buffer is used at this stage. As the input impedance of an inverter is high (almost 100 MΩ), this stage is used for impedance matching between the microcomputer I / O card and the following stages. Each IC 7404 chip contains six inverters. To keep the original shape same we need 12 inverters i.e. two IC 7404 chips. The biasing voltage for

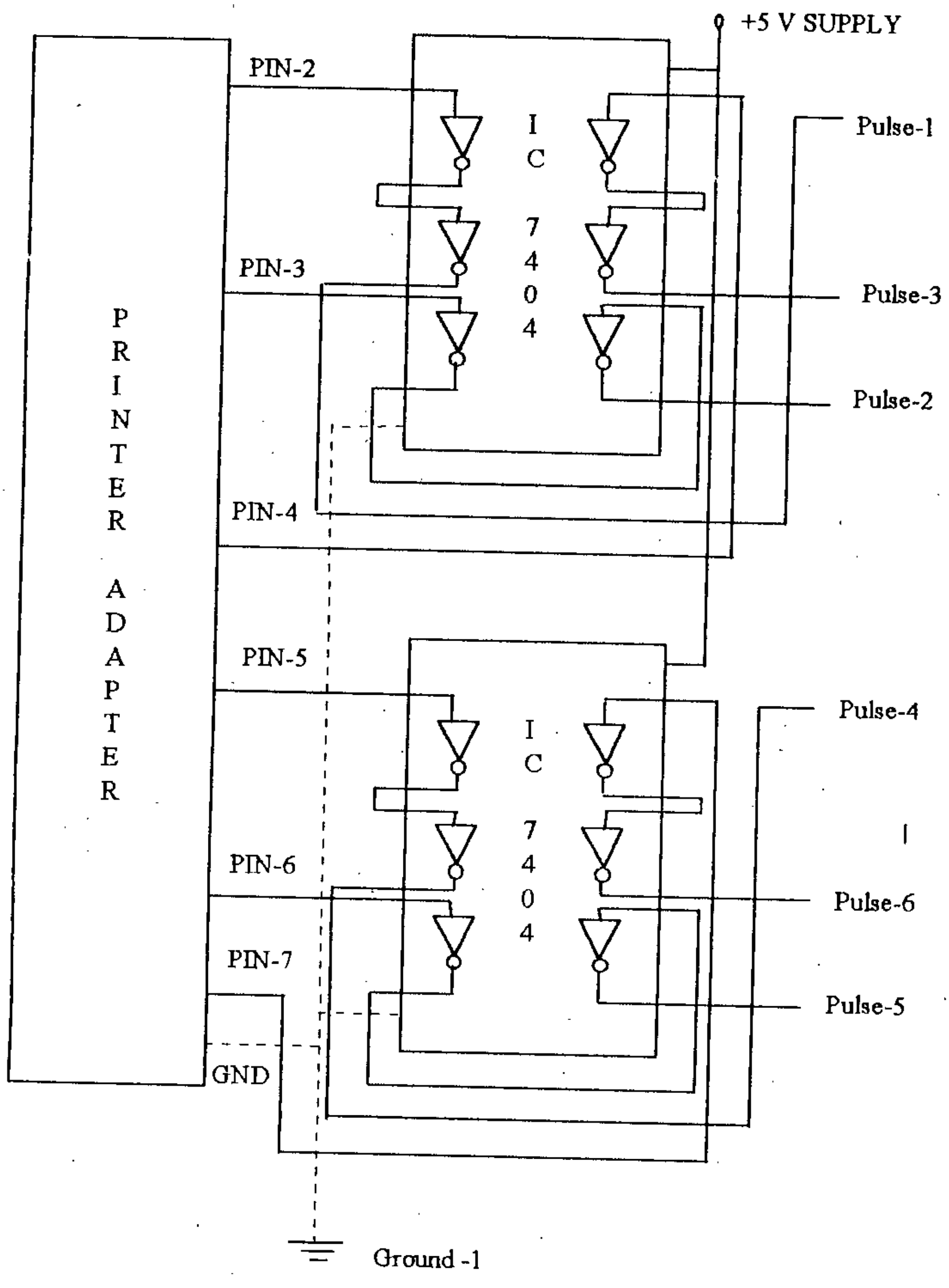
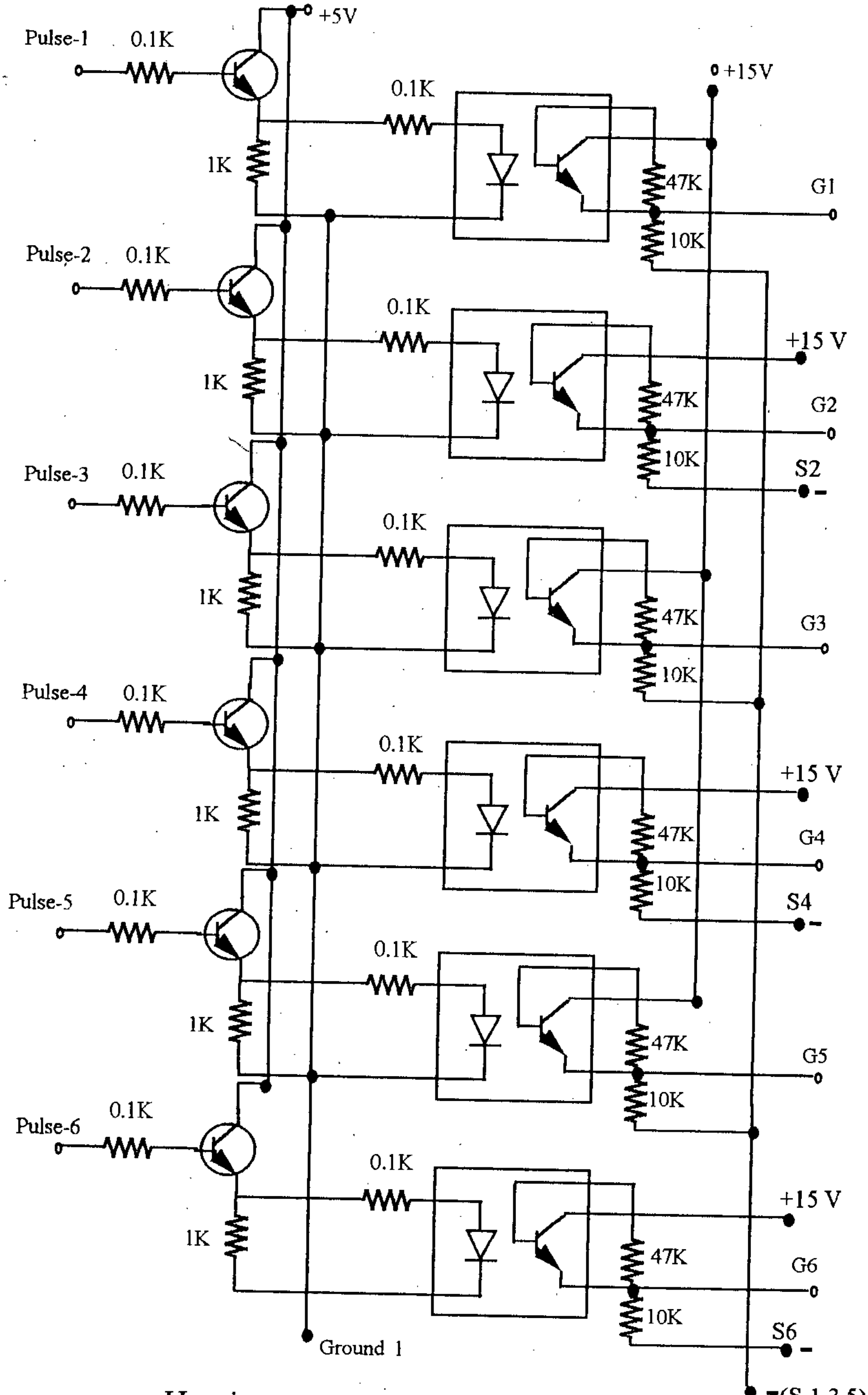


Figure 3.2 : Printer Adapter to Buffer stage



Hex inverter to opto-isolation stage

the chip is +5 Volts dc which is given from the dc power supply module. This stage is shown in figure 3.2.

2. Amplifier And Opto-Coupler Stage

As the signals at the output stage of the buffer are not sufficient to drive the opto-couplers it has to be amplified. In the practical circuit a transistor amplifier has been used for this purpose. The amplifier stage contains fixed bias npn BD243 transistor connected in common-collector configuration. Signals from the buffer stage is applied to the base of the transistor through a 100 Ω resistor in series with the base. The common-collector configuration has high current gain and is sufficient to drive the opto-couplers. The opto-couplers numbered 4N25 are used to provide opto-isolation between the microcomputer and the power MOSFET rectifier circuit. Opto-isolation prevent any current flow between two systems. To limit current, signals are applied by connecting a 100 Ω resistor in series with the LED's. The opto-coupler are in a package that contains both an infrared LED and a photodetector npn transistor. The wave-length response of each device is tailored to be as identical as possible to permit the highest possible coupling. They are designed with fast response times so that they can be used to transmit data in the megahertz range. The biasing voltage for the photo-transistor of the opto-coupler is +15 V. A 47 kΩ resistor is connected between base and emitter of the transistor which ensures a quick recovery of the minority carrier in the base of the phototransistor and helps to keep the signal shape intact while passing through the opto-coupler stage. The output of the opto-coupler stage is connected across the 10 kΩ resistor connected between emitter and ground of the phototransistor. The complete diagram of this stage is shown in figure 3.3.

3.3.2 Practical Rectifier Circuit

A three phase rectifier consisting power MOSFET's IRF 840 as the switching device is shown in figure 3.4. MOSFETs IRF 840 are n-channel enhancement type MOSFETs which can withstand a maximum drain to source voltage of 500 volts. Power MOSFET's are chosen as the switching device as their switching times are of the order of nanoseconds.

89812

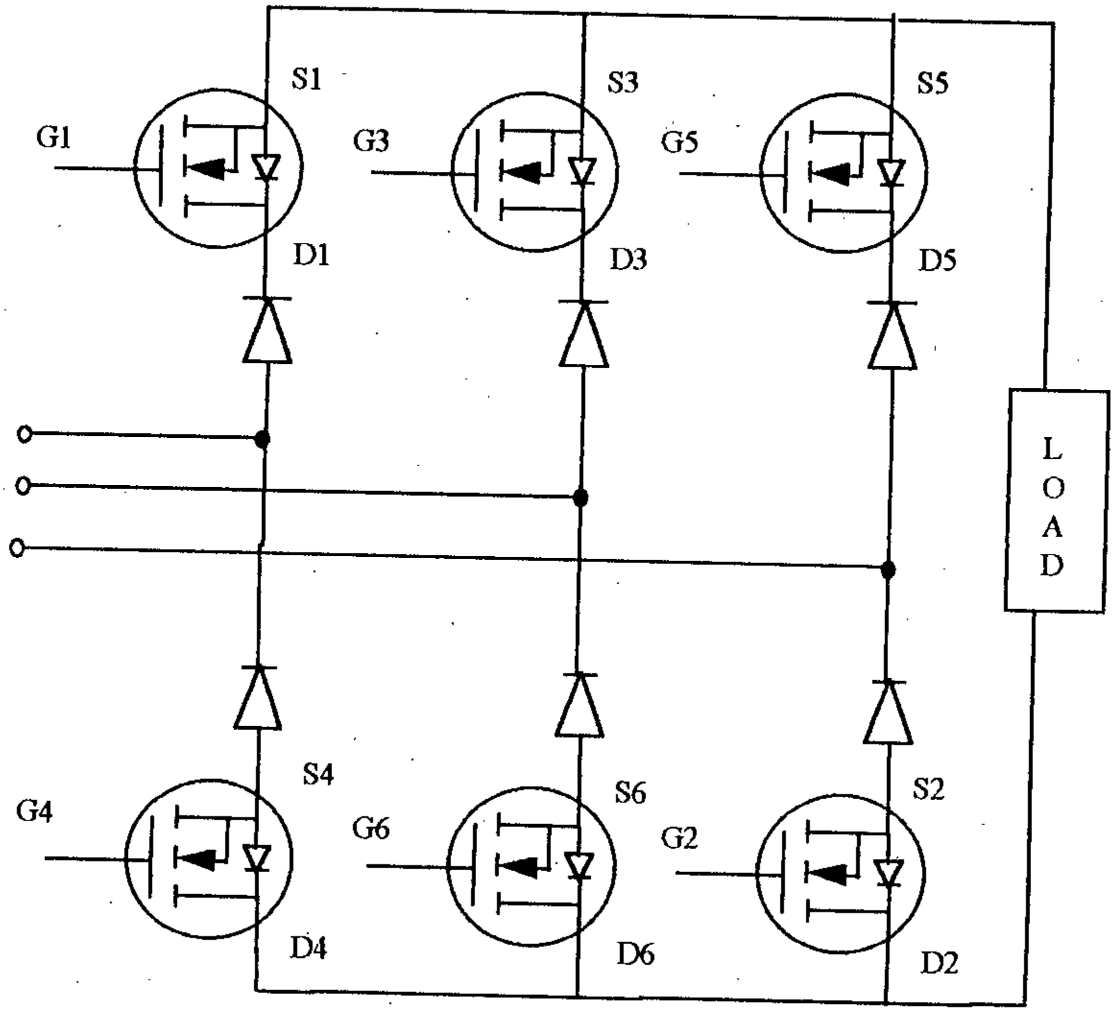


Figure 3.4: Practical Rectifier Circuit

3.4 Experimental Results

Representative waveforms of a three phase pulse width modulated rectifier controlled on line by a microcomputer are shown in figures 3.5-3.12. The waveform were recorded for various modulation parameter variations. The modulation parameters changed on line by Keyboard command of a microcomputer are the carrier wave frequency (f_c) and the modulation index ($m = E_m / E_c$). Waveforms of output voltage and input current were recorded. Figures 3.5 to 3.6 show rectifier output voltage waveform for the carrier frequencies 750 Hz and 1050 Hz, while other parameters of modulation were maintained constant. Figures 3.7 to 3.8 show the rectifier output voltage waveform for widely varied modulation index while other parameters of the modulation were maintained constant. In these illustrations it is observed that as the carrier frequency f_c is varied switching per cycle changes, i.e. increasing with the increase of carrier frequency. This would cause the dominant harmonic voltage waveform to shift. In this case of change in modulation index (Figs 3.7 and 3.8) it is observed that the pulse widths of the output voltage waveform changes. This would cause the output voltage variation of the rectifier. Hence by microcomputer control, rectifiers output voltage can be controlled from very low to full voltage in PWM mode. The above two phenomenon are the objectives of all modulated solid state converter control which enables output voltage and harmonic of the output voltage to be controlled in a single stage. The carrier frequency control enables the solid state converter users to replace filter sizes at both the output and the input ends of converters causing dominant harmonics to occur at higher frequencies requiring smaller L/C filters. In the case of output voltage the effect of carrier frequency is shown in Figs 3.5 and 3.6 for the microcomputer controlled rectifier. To get the idea of the same effect to reduce input filter size Figs 3.9 to 3.12 shows the typical input current

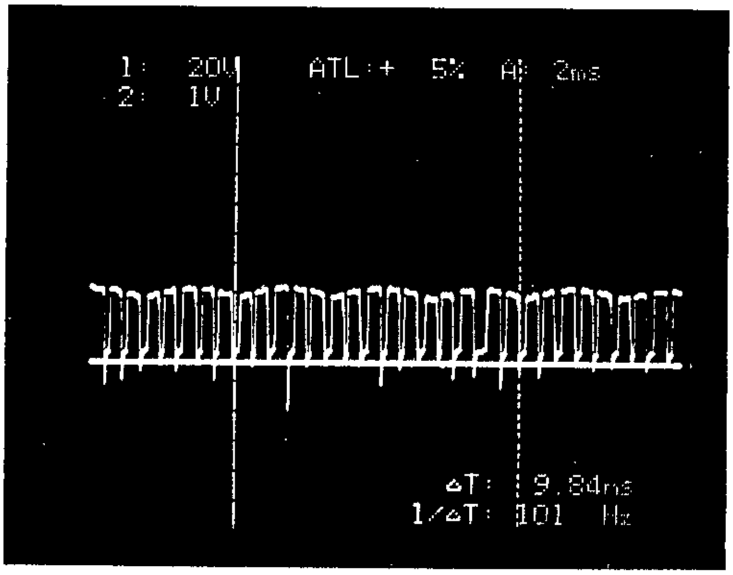


Figure 3.5 : Rectifier Output Waveform

$E_m = 0.6$, $f_c = 750$ Hz, and $f_m = 50$ Hz

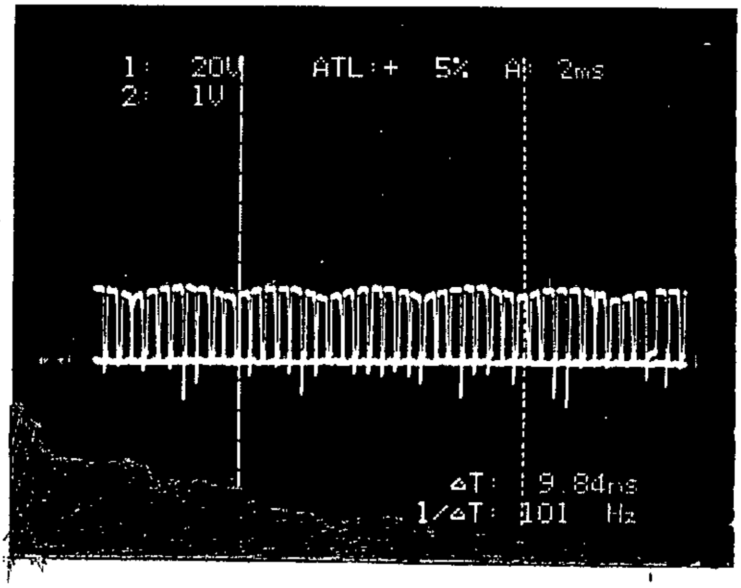


Figure 3.6 : Rectifier Output Waveform

$E_m = 0.6$, $f_c = 1050$ Hz, and $f_m = 50$ Hz

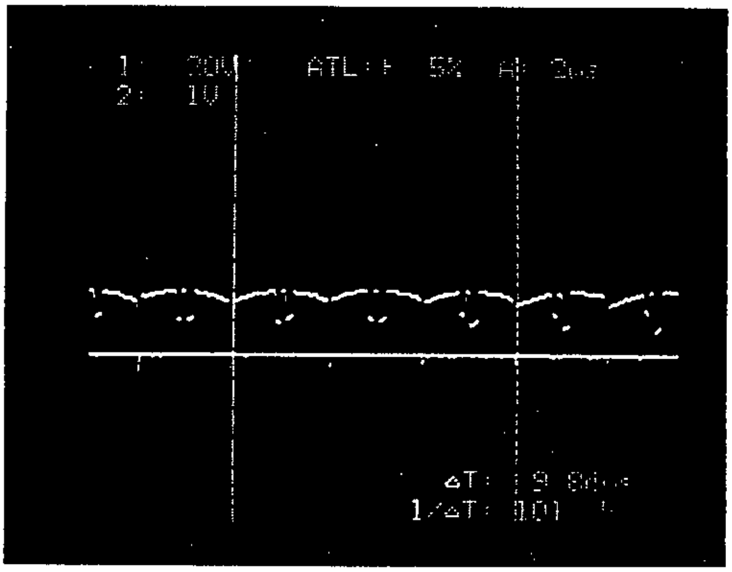


Figure 3.7 : Rectifier Output Waveform

$E_m = 0.8$, $f_c = 550$ Hz, and $f_m = 50$ Hz

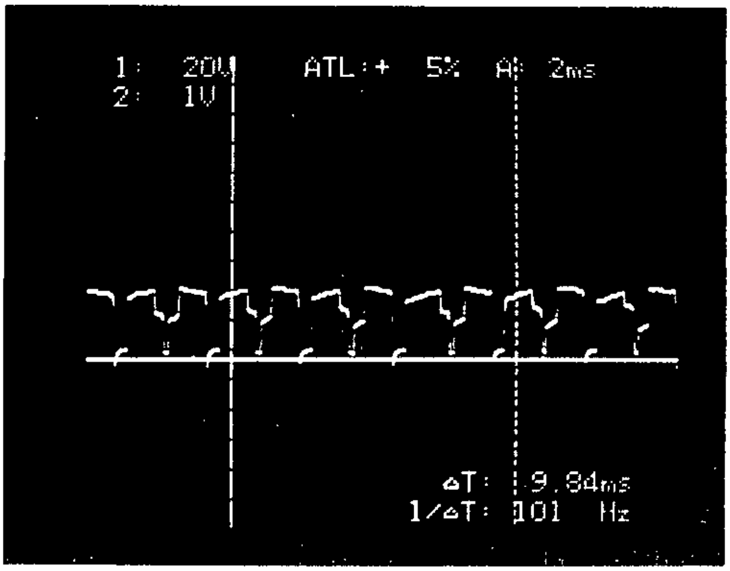


Figure 3.8 : Rectifier Output Waveform

$E_m = 0.4$, $f_c = 550$ Hz, and $f_m = 50$ Hz

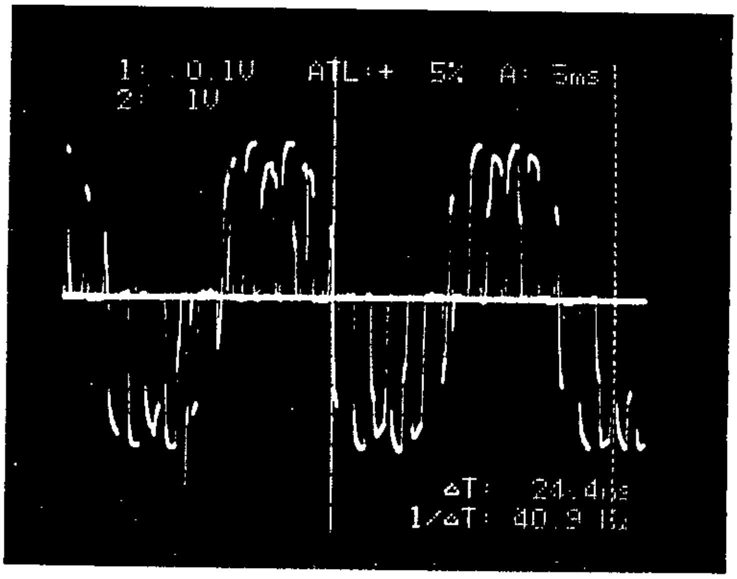


Figure 3.9 : Input Line Current Waveform

$E_m = 0.4$, $f_c = 550$ Hz, and $f_m = 50$ Hz

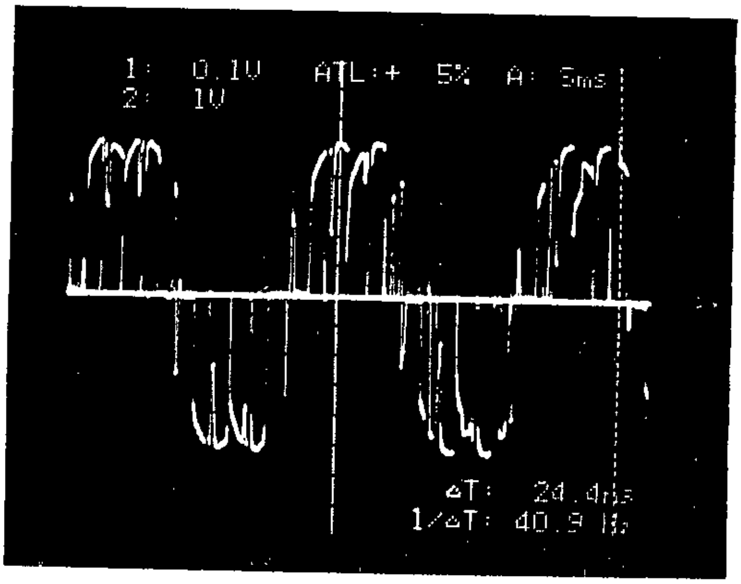


Figure 3.10 : Input Line Current Waveform

$E_m = 0.8$, $f_c = 550$ Hz, and $f_m = 50$ Hz

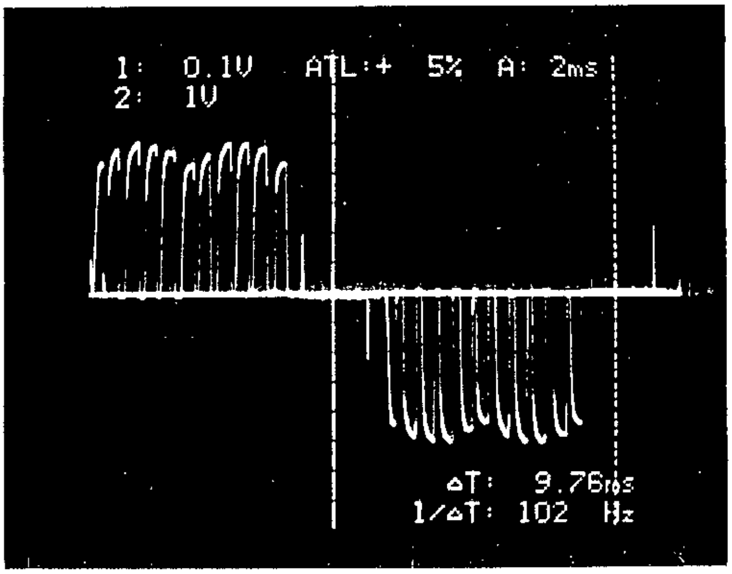


Figure 3.11 : Input Line Current Waveform

$E_m = 0.6$, $f_c = 750$ Hz, and $f_m = 50$ Hz

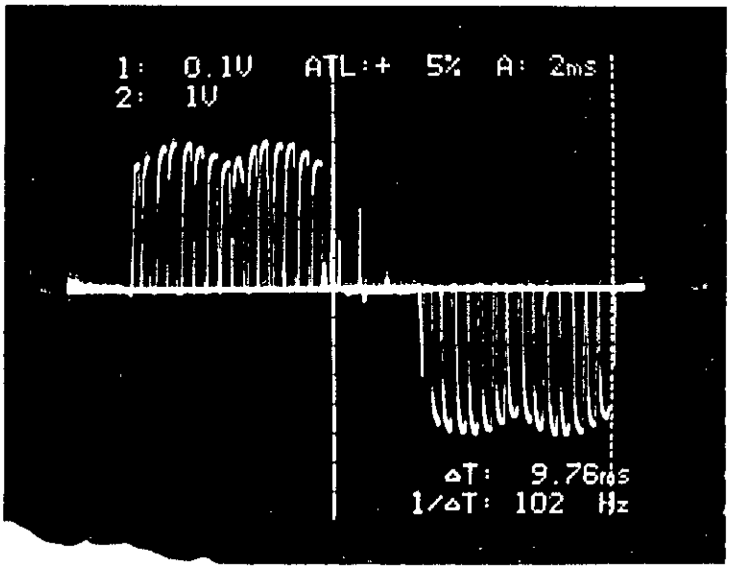


Figure 3.12 : Input Line Current Waveform

$E_m = 0.6$, $f_c = 1050$ Hz, and $f_m = 50$ Hz

waveshapes of the microcomputer controlled three phase rectifier. Figures 3.9 and 3.10 show the effect of modulation index variation, whereas, figures 3.11and 3.12 show the effect of variation of carrier frequency on the input line currents of a PWM three phase controlled rectifier.

3.5 Observation

Experimental part of this research only demonstrates the successful on-line microcomputer controlled operation of a three phase PWM rectifier. It is observed that increase of carrier frequency f_c causes the switching frequency of the modulated output voltage and input current to increase, whereas, the variation of modulation index causes the pulse widths of the output voltage and input current to vary. Microcomputer implementation of these parameters would allow one to vary the parameters simultaneously to achieve a desired operating condition of controlled rectifiers. Detail analysis and investigations were not carried out for such operation during the time of this work. Also performance analysis of microcomputer controlled rectifier was not done during this thesis.

Chapter 4

Summary and Conclusions

In this thesis the implementation of microcomputer controlled three phase rectifier has been presented. The practical implementation and observation have been carried out successfully in the laboratory for various settings of control parameters. The limitations of practical implementation of the work and the possible aspects of future research is discussed in this chapter.

On line microcomputer controlled sine pulse width modulated rectifier has been implemented and analyzed in the laboratory with an acceptable performance. The gating signals for the MOSFET rectifier have been provided from an INTEL DX-2 microcomputer through properly designed interface and isolation circuits. The gating signals available at the parallel port and the output waveforms of the three phase rectifier with resistive load have been observed and presented in this thesis. The frequency and timing instants of each and every switching pulse patterns have been verified with the guide menu of the storage oscilloscope which show close agreement between the theoretical and practical values. The success of this implementation complies with the on line control of modulation parameters from the key board.

Typical rectifier waveform has been recorded and the following observations are made.

- In PWM waveforms average dc voltage increases with increase in modulation index with other parameters remaining constant.
- The voltage variation is linear with modulation index. With increase in carrier frequency dominant harmonics occur at higher frequency.

4.1 Limitations

4.1.1 Limitations of the Software

Resolution of the generated pulse depends on the value 'del' which has been kept by a stop watch named 'Ftimer'. But the stop watch itself executes some of the embedded function built in timer.cpp and timer.h. So there is some error in the accuracy of the small time division 'del'. And because of this it cannot be as small as possible. In this work the value of 'del' has been chosen at a value where the gating signals are most stable.

4.1.2 Limitations of Practical Implementation

- Main problem of practical implementation is to synchronize the gating signals with the input AC signal which is a must for a rectifier to operate properly. In our work we have done it manually.
- To avoid short circuit problem through the MOSFETs we have placed diode in series with each MOSFETs at the input side of the MOSFETs. This is very important otherwise we could not avoid severe short circuit condition which in turn causes damages to the MOSFETs.
- To keep the gating signals isolated from each other we have used four DC supply units. From practical side, it is not a good implementation technique. We can minimize the number of DC power supply module by using pulse transformer for each gating signals. Isolation transformer of ferrite core is necessary for this purpose, but during the time of this research such ferrite core transformer were not available in the local market.

4.2 Recommendations for Future Work

Recommendations and avenues for future work include:

- To provide synchronization is the primary necessity for this work. As we are taking gating signals from the parallel port of the printer adapter so we can use the busy pin of the printer adapter to synchronize the input AC signal with gating pulses.
- Detailed analysis of the three phase sine pulse width modulated rectifier with both resistive and motor loads.
- Comparative evaluation of the performance of the rectifier with other PWM controlled rectifiers should be undertaken.
- The developed on-line microcomputer control system for sine pulse width modulated rectifier is an open-loop control system where the parameter is carried out manually by key press. Future work may design this system with feedback control where the switching parameters will be changed automatically sensing the output performances of the system to provide best operating condition.
- Hysteresis current limit technique in the PWM control should be investigated so that both the input and output current become more sinusoidal.

Appendix A

```
//*******************************************************
//Program for on line control of rectifier by SPWM
//*******************************************************
*******
#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
# include <iostream.h>
#include <dos.h>
# include <conio.h>
#include <math.h>
# include <fstream.h>
# include <\tc\classlib\include\timer.h>
# include <\tc\classlib\source\timer.cpp>

//This part identifies all the variable used in this program

        double p,x;
        int n,N,i,ii,i1,i2,i3,i5,N1,j,k,r;
        int deci[3000],a,b,num_seg,bb,bbb;
        float fc,Em,Ec,wm,fm,am,sp,tm,tt,del;
        float t[50],theta[50],the[50],dev[50];
        double Ftime=0.0;

        int flag = 0;
        char state = ' ', ch;
        char val[100], z;
        Timer Ftimer;


//This function calculates the timing instants and generates deci[num_seg]

void SPGDA( void )
{
        puts( "running..." );
        N1 = 0;
        Ec=1.0;
        wm=2.0*M_PI*fm;
        tm=1.0/fm;
        num_seg=tm/del;
        printf("\segment=%d\n",num_seg);
        printf("\del=%f\n",del);
        N=fc/fm;
```

```
          am=Em/Ec;
          for(i=1;;i++)
                 {
                 theta[i]=((2.0*i-1.0)*M_PI)/N;
                 dev[i]=(M_PI/N)*(1+am*sin(theta[i]))/(2.0*M_PI*fm);
                 the[i]=theta[i]/(2.0*M_PI*fm);
                 i3=2*i;
                 i2=2*i-1;
                 for(i1=i2;i1<=i3;i1++)
                 {
                 x=-1.0;
                 p=pow(x,i1);
                 t[i1]=(the[i]+p*dev[i]/2.0);
                  if(t[i1]>=M_PI/wm) break;
                 }
                 N1=i1;
                 if(t[i1]>=M_PI/wm) break;
                 }
// This portion generates deci[num_seg] array
         for(r=0;r<num_seg;r++)deci[r]=0;

         for(k=0;k<6;k++)
         {
          if(k==0)       sp=0.0;
         else if(k==1)sp=tm/6.0;
         else if(k==2)sp=tm/3.0;
         else if(k==3)sp=tm/2.0;
         else if(k==4)sp=2.0*tm/3.0;
         else            sp=5.0*tm/6.0;

        for(ii=1;ii<N1;ii=ii+2)
              { for(tt=t[ii];tt<=t[ii+1];tt=(tt+del))
                     { r=(int)((tt+sp)/del);
                           if(r>(num_seg))
                           r=(r-(num_seg));
                           deci[r]=deci[r]+(int)(pow(2.0,(float)(k)));
                     }
              }
        }
}
```

```
// This Function generates the required pulses

void SPWM( void )
{
        union REGS regs;
        do {
                regs.h.ah = 2;
                regs.x.dx = 0;
                int86( 0x17, &regs, &regs );
        } while( ( regs.h.ah & 0x80 ) == 0 );
        for(r=0;r<num_seg;r++)
        {
        outportb(0x378,deci[r]);
        a=1;
        do {
        b=2*3;
        bb=b*2;
        bbb=bb*2;
        } while( ++a<=15 );

        }
}

// This is the main program

void main(void)
{

        Em=0.8;
        fc=550.0;
        fm=50.0;
        Ftimer.reset();
        Ftimer.start();
        a=1;
        do {
        b=2*3;
        bb=b*2;
        bbb=bb*2;
        } while( ++a<=10 );
        Ftimer.stop();
        del=Ftimer.time();
```

```
    SPGDA();

      for ( ; state != 'q'; )
      {
            if( !flag && kbhit() )
            {       state = getch();
                    printf( "\n I am ready to take new [%c]...\n\n", state );
                    flag = 1;
                    z = 0;
            }
            if( flag && kbhit() )
            {       ch = getche();
                    if( isdigit( ch ) || ch == '.' )
                     { val[z++] = ch;}
                    else {  val[z] = 0;
                                    switch( state ) {
                                    case 'f' : sscanf( val, "%f", &fm ); break;
                                    case 'e' : sscanf( val, "%f", &Em ); break;
                                    case 'c' : sscanf( val, "%f", &fc ); break;
                                    }
                                    SPGDA();
                                    flag = 0;
                            }

            }
            SPWM();
      }
}
```

REFERENCES

1. D.E. Steeper, "Reactive Compensation and Harmonic Suppression for Industrial Power systems sing Thyristor Converters", IEEE Trans. Ind. Appl. vol. IA-12, pp. 232-254, May/June, 1976.
2. D.D. Shipp, "Harmonic Analysis and Suppression for Electrical Systems Supplying Static Power Converters and Other Nonlinear Loads", IEEE Trans. Ind. Appl. vol. IA-15, pp. 453-457, Sept./Oct. , 1979.
3. B.R. Pelly, "Thyristor Phase Controlled Converters and Cycloconverters", New York : Wiley Interscience, 1971.
4. S. Mukhopadhyaya, "A New Concept for Improving the Performance of Phase-Controlled Converters", IEEE Trans. Ind. Appl. vol. IA-15, pp. 193-200, March/April, 1979.
5. P.C. Sen, "Thyristor DC Drives", New York : Wiley Interscience, 1981.
6. P.C. Sen, and S.R. Doradla, "Evaluation of Control Schemes for Thyristor Controlled DC Motors", IEEE Trans. Ind. Elect. Cont. Inst. vol. IECI-25, pp. 247-255, August, 1978.
7. M.H. Rashid, and M. Aboudena, "Analysis of Forced Commutated Techniques for AC-DC Converters", Proc. First European Conf. on ower Electronics and Appl. , vol. 1, 1985, pp. 263-266.
8. T. Kataoka, K. Mizumachi, and S. Miyairi, "A Pulsewidth Controlled AC-to-DC Converter to Improve Power Factor and Waveform of AC Line Current", IEEE Trans. on Ind. Appl. , vol. IA-15, pp. 670-675, Nov./ Dec. , 1979.
9. S.R. Doradla, C. Nagamani, and S. Sanyal, "A Sinusoidal Pulse Width Modulated Three Phase AC-DC Converter Fed DC Motor Drive." IEEE Transaction on Industrial Applications, vol. IA-21, Nov./Dec., 1985, pp 1394-1408.

10. P.D. Ziogas, "Optimum Voltage and Harmonic Control PWM Techniques for Three-Phase Static UPS Systems", IEEE Trans. Ind. Appl. , vol. IA-16, pp. 542-546, July/August, 1980.
11. M.A. Rahman, A.R.D. Esmail, and M.A. Choudhury, "Analysis of Delta PWM static AC-DC Converters." IEEE Transactions of Power Electronics, vol.10, No.4, July 1995, pp 494-503.
12. T. Kataoka, K. Mizumachi, and S. Miyari, "A pulse Width Controlled AC-DC Converter to Improve Power Factor and waveforms of AC Line current." IEEE Transaction on Industrial Applications, vol. IA-15, Nov./Dec., 1979, pp 670-675.
13. P.C. Tang, S-S. Lu, and Y-C. Wu, "Microprocessor Based Design of a Firing Circuit for Three-Phase Full Wave Thyristor Dual Converter." IEEE Trans. Ind. Elec. vol. IE-29, no. 1, pp 67-73, Feb. 1982.
14. W.L. Fredrich and K.S. Swenson, "Microprocessor Based Converter Firing Pulse Generator." conference record - IEEE - Industry Applications society annual meeting, 1982.
15. A. Mirbod and A. El-Amawy, "A Novel Microprocessor Based Controller for a Phase Controlled Rectifier Connected to a Weak AC System." conference record - IEEE - Industry Applications society annual meeting, 1985.
16. E. Ohno, M. Ohta, M. Yano, H. Naito, and Y. Yamamoto, "Microprocessor Applications for Thyristor Choppers and High Voltage Thyristor Switches", U.S. - JAPAN Seminar conference record, 1982.
17. P.M.S. Nambison, R. Mulchandani, and C.M. Bhatia, "Microprocessor Implementation of a Four-Phase Thyristor Chopper Circuit", conference record IEEE Industry Applications Society Annual Meeting, 1983.
18. G. Olivier, V.R. Stefanovic, and G.E. April, "Microprocessor Controller for a Thyristor Converter with an Improved Power Factor", IEEE Trans. on Ind. Electronics and Control Instrumentation, August 1981.

19. B.K. Bose, P.M. Szezesny, and R.L. Steigerwald, "Microprocessor Control of a Residential Photovoltaic Power Conditioning System." conference record - IEEE - Industry Applications society annual meeting, September / October 1984.

20. J.B. Casteel and R.G. Hoft, "Optimum PWM Waveforms of roprocessor Controlled Inverter", 9th IEEE Power Electronics Specialists Conference, 1978.

21. K.S. Rajashekera and J. Vithayathil, "Microprocessor Based Sinusoidal PWM Inverter by DMA Transfer", IEEE Trans. on Industrial Electronics, February, !982.

22. B.K. Bose and H.A. Sutherland, "A High Performance Pulse-Width Modulation For an Inverter-Fed Drive System Using a Microcomputer." IEEE Trans. Ind. Appl., vol. IA-19, Mar.-Apr. 1983, pp 235-243.

23. Md. Bashir Uddin, "Stability Analysis of PWM Inverter Fed Synchronous Motors." Ph.D. thesis, Dept of EEE, BUET, Dhaka-1000.

24. M. N. Anwar, "Implementation of Microcomputer Controlled Delta Modulated Inverter.", M.Sc. thesis, Dept of EEE, BUET,Dhaka-1000.

25. M.A. Rahman, J.E. Quaicoe, and M.A. Choudhury, "A Comparative Study of Delta and Sine PWM Inverters." proceedings of the first European Conference on Power Electronics Application, Belgium, vol. 1, 1985, pp I.63-1.69.

26. M.A. Choudhury, "Analysis of Delta PWM Inverter." M.Sc. Engg. thesis, Memorial University of Newfoundland, Canada, August 1984, pp 5.5-5.11.

27. M.A. Choudhury, Md. Bashir Uddin, and M.A. Rahman, "Analysis of Single Phase Delta Modulated Cycloconverter." accepted for publication in IEEE Transaction on Industrial Electronics.

28. M.H. Rashid, and M. Aboudena, "Analysis of Forced Commutated Techniques for AC-DC Converters", Proc. First European Conf. onPower Electronics and Appl. , vol. 1, 1985, pp. 263-266.