MICROPROCESSOR IN POWER SYSTEM INSTRUMENTATION

A DISSERTATION

Submitted in partial fulfilment of
the requirements for the award of the Degree
of
MASTER OF ENGINEERING
in
ELECTRICAL ENGINEERING
(MEASUREMENT AND INSTRUMENTATION)

By

ATUL NIGAM

DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITY OF ROORKEE
ROORKEE (INDIA)
1981

A C K N O W L E D G E M E N T

The author humbly expresses his deep sense of gratitude to his guides Dr. V.K. Verma, Professor in Electrical Engineering and Mr. V.K. Bansal, Reader in U.S.I.C. for their keen interest, kind guidance and thoughtful advice rendered by them during the course of dissertation work.

ATUL NIGAM

C E R T I F I C A T E

Certified that the dissertation entitled "MICRO-PROCESSOR IN POWER SYSTEM INSTRUMENTATION" which is being submitted by ATUL NIGAM in partial fulfilment for the award of Degree of Master of Engineering in ELECTRICAL ENGINEERING (Measurement & Instrumentation) of the University of Roorkee, Roorkee, is a record of the student's own work carried out by him under my supervision and guidance. The matter embodied in this dissertation has not been submitted for the award of any other degree or diploma.

This is further certified that he has worked for a period of 18 months from January ~~1979~~ 1980 to June 1981 [illegible] for preparing this dissertation for the Master of Engg. Degree at this University.

[illegible]
(V.K.BANSAL)
READER
USIC
ROORKEE,U.P.

[illegible]
(V.K.VERMA)
PROFESSOR
ELECTRICAL ENGG.DEPTT.
UNIVERSITY OF ROORKEE

.. ..

ACKNOWLEDGEMENT

The author humbly expresses his deep sense of gratitude to his guides Dr. V.K. Verma, Professor in Electrical Engineering and Mr. V.K. Bansal, Reader in U.S.I.C. for their keen interest, kind guidance and thoughtful advice rendered by them during the course of dissertation work.

ATUL NIGAM

CONTENTS

CHAPTER 1

INTRODUCTION

In the modern power houses, the protection of equipments has become very important. Invariably the protection of machines is an interdependent phenomenon i.e. the protection of one machine is related to a group of machines. The system requires the monitoring of a large number of parameters and after processing the data the commands have to be issued to various machines for control purposes.

The invention of microprocessor has made possible the design of such equipments for protection and control purposes. It is possible to monitor a very large number of interdependent parameters, process them and then decide the control command for various machines with the help of a small desk top micro-processor based equipment.

Synchronization of generators on to the main bus is an important problem. It requires the continuous monitoring of voltage, frequency and phase angle of generator and the bus. The control commands have to be issued to the generator C.B., field circuit and governors to achieve safe synchronization.

A machine can be switched into the system, only when certain conditions necessary for its synchronization with the system are satisfied. Manual method using Voltmeter to judge the voltage and synchroscope to judge the slip frequency and phase difference is common, but not suitable when synchronizing is to be carried out rapidly and accurately. For example, in peak load stations, generators have to be operated only during peak loads and have to be shut down during slump load period. Hence there is requirement for a faster synchronization.

Solid state components based automatic synchronizers have already been reported (1,2,3,). Such synchronizers use a very large number of discrete components leading to a complexity of circuitary and lowering of its reliability.

A microprocessor based synchronizer can over come above mentioned a limitations of solid state components based automatic synchronizer. A.K. Ghai, H.K.Verma and P.Mukhopadhaya have developed and laboratory tested a microprocessor based synchronizer which is claimed to have a good prospect for small power houses to assist the operating engineer to carry out perfect and fast synchronization. They used on 8 bit SC/MP microprocessor system as a controller. First of all they measured and compared the voltages outside of microprocessor when it comes within the required band, the frequency of the machine and system was measured and compared in terms of the time periods inside the microprocessor. After the fulfilment of frequency condition, the C.B. is closed at minimum phase difference of the two voltages.

In the proposed synchronizer, the voltages are measured and compared inside the microprocessor itself. For this voltages are sampled and peak value is compared. When voltage and frequency of the incoming machine and system came within the required band, the rate of change of phase angle is calculated in order to send the closing command to circuit breaker to close the circuit at zero phase difference angle.

An 8-bit Intel 8080 A system has been used in the work presented here.

CHAPTER 2

PROBLEM FORMULATION

When an incoming machine has been paralled with busbars, the factors influencing the extent of any disturbances or surges which may arise are- the capacity of the running plant, voltage, frequency, and phase of the machine with respect to those of busbars, and the synchronising torque. Accordingly, the conditions are laid down in the reported work to obtain a perfect and accurate synchronizations ensuring minimum system disturbances.

2.01 Conditions for Synchronization :

Following are conditions for synchronization of incoming machine with the system :

(1) Phase sequence of the incoming machine voltages and that of busbars must be same.

(2) The difference between magnitudes of their voltages must be small. In proposed work $\pm$ 5% of system voltage is assumed,but adjustable to any other value.

(3) Difference in frequencies of the two must be small. A maximum slip frequency of $\pm$ 0.5 % of the supply frequency is assumed in proposed work. However to get more precision slip frequency can be adjusted to the desired value.

(4) The difference in time phase between the incoming machine and running voltage must be small and decreasing with time. The proposed system is designed to permit the circuit closure at zero degree phase difference,by

taking into account the effect of rate of change of phase angle and circuit breaker closing time.

2.1 <u>Synchronizer Details</u> :

The proposed synchronizer compares the machine voltage and frequency with those of busbars and gives signals for correcting the generator excitation and its speed. Once the voltage and frequency values are acceptable, the closing signal is sent to the circuit breaker, after taking into consideration the circuit breaker closing time, such that the circuit breaker closes at exact phase coincidence.

In proposed synchronizer, the busbar and incoming machine voltages from the respective transformers (V.Ts.) are fed to rectifiers through auxiliary VTs. The rectified voltages are fed to A/D convertors and then to microprocessor for voltage measurement. The largest value in one cycle of one voltage signal is compared with the largest value in one cycle of other voltage, signal. If the two values are within the acceptable limit, microprocessor starts frequency measurement. For this, the rectified signals are squared and fed to microprocessor. With the help of clock, the counting is done inside the microprocessor for the time periods of the two wave-forms. The two counts are compared and when the counts difference comes within the acceptable limit, microprocessor starts to calculate the rate of change of phase angle. Otherwise, the frequency high or frequency low signal is sent to governor for correction.

To calculate the rate of change of phase angle, the two signals which were used for frequency (time period) measurement are again used. Number of checking cycles between the two rising edges of waveforms are counted and when it starts decreasing, the rate of change is calculated and thus the instant of zero phase difference is calculated.

2.2 Block Diagram Description :

The system solving the autosynchronization problem can be represented in following blocks (Refer fig. 2.2) :

Auxiliary Transformer :- The order of the incoming bus and alternator voltage is 230 Volt. Which is too high to be used with other blocks of the system. Therefore a step down transformer is used.

Rectifier :- Input signals are rectified so that they can be used for voltage and frequency measurement.

Squaring circuit :- Rectified signal is fed to squaring circuit. The output of the squaring circuits are the square waves at the bus and generator frequencies respectively. The CPU carries out the time period measurement of the above waveforms.

Analog to Digital (A/D) convertor :- Signals from the rectifier circuits are fed to A/D convertors. As the name suggests the blocks convert the analog signals into digital signals for the voltage measurements, inside the CPU. Further the CPU also compares the bus and generators voltages and issues a MATCH command when the two are within programmed limits. Otherwise a "GENERATOR VOLTAGE HIGH" or " GENERATOR VOLTAGE LOW" commands are sent to the governors.

Input/Output Ports :- The signals from squaring circuits and Analog to Digital convertors are fed to microprocessor through the input ports. Those input ports which are connected to the signals coming from A/D covertors are used as Output port also for outputting the mesages and command such as voltage low, voltage high, frequency low, Frequency High and circuit Breaker closing command. Teletype is also connected with IP/OP Port.

Read Only Memory :- This is used to store the Monitor programme and other various programmes for Voltage measurement and comparision, Frequency measurement and comparision, calculation of rate of change of phase angle etc.

Random Access Memory (RAM) :- This is used for storing the intermediate results involved in the execution of program.

CPU :- This is heart of microcomputer where the actual execution of instructions takes place.

CRT :- With the help of CRT, the operator can communicate with the equipment. The tolerances for voltage and frequency comparisions can be modified if required. A print out of circuit breaker closing command can also be obtained.

2.3 Flow Chart description :-

Fig. 2.3.1 shows the basic flow chart for autosynchronizer.

For this, maximum value of bus voltage is achieved on which tolerance of 5% is calculated. The maximum value of generator voltage is calculated and compared with the tolerance band across bus voltage. If it does not lie in between tolerance band, a signal for adjustment of generator voltage is outputted and after a time delay of 5 seconds, same process is repeated.

When generator voltage comes within tolerance band, the time period corresponding to bus frequency is calculated on which tolerance of 0.5% is calculated. Then generators time period is calculated and compared with the tolerance band across the bus time period. If it does not lie in between tolerance band, a signal for adjustment of generator frequency is outputted and after a time delay of 5 seconds, same process is repeated.

When generator voltage and frequency comes under tolerance band , the phase difference between the bus and generator is calculated. The time for zero phase difference is anticipated. The circuit breaker closing command is sent, taking the circuit breaker closing time in to account.

Fig. 2.3.2 shows the flow chart for voltage measurement. The output of Andogy to Digital (A/D) convertor is inputted and stored. This value is compared with the next output of A/D convertor. Both the values are compared and larger value is stored. In this way, largest value of voltage signal is retreived.

Fig. 2.3.3 shows the flow chart for frequency measurement. Frequency is measured in terms of time period. The coutning starts only when the zero to one transition tapes place coutning is stopped when one to zero transition takes place thus the available counts are proportional to half value of frequency.

Fig. 2.3.4 shows the flow chart for phase measurement. Phase is measured in terms of counts between the bus signal and generator signal. To avoid mistake in counting,masking is done so that the counting initiates at the right moment.

CHAPTER 3

MICRO PROCESSOR BASIC

3.01 INTRODUCTION :

The advent of minicomputers enabled the designers to include digital computers as part of various process control systems. Unfortunately the size and cost of mini computers has limited their use in "dedicated applications". The advent of microcomputer has made it possible to have a one card computer (size 6" x 4") as an integral part of the Process control equipment Microcomputer provides immense flexibility to the electronics engineering designing systems. By simple reprogramming the microcomputer, it is possible for the system designer to have another system.

Consider the design of an automatic computing scale for the supermarket. The weighting unit finds out the weight of the commodity, and the price of the commodity is entered to the microcomputer through a keyboard and it is required to work out and display the total price.

The system block diagram is shown in figure 3.1. The weighing machine measures the weight of the object. The weight is converted in binary form and made available to the system through the Input Interface 1. The rate of the commodity is made available through Input interface 2. The central processing unit has multiplied the two numbers and OUTPUTS, the result to displays through output Interface. The steps can be summarised as below :

1. INPUT WEIGHT TO CPU FROM INTERFACE 1
2. INPUT PRICE TO CPU FROM INTERFACE 2
3. MULTIPLY WEIGHT AND PRICE
4. OUTPUT VALUE TO OUTPUT INTERFACE.

These commands are stored in programmable read only memory (PROM) in machine language instruction i.e. in binary form. The CPU FETCHES the instruction from PROM and EXECUTES. Random access memory (RAM), which can be used for both READ and WRITE operation can also be used for storing the programme or any data. The CPU and other devices communicate with each other on a bi directional bus.

Each Central processing unit has a set of instructions associated with it which can be broadly classified into the following three categories :

(a) Transfer instructions : These set of instructions are required to affect transfer of information between the three components of the microprocessor system i.e. CPU memory and Input/Output ports.

(b) Accumulator group instructions : These instructions are required to carry out arithmatic and logic operations like addition, subtraction, AND, EXOR, comparision etc. inside CPU, on the data which has been transferred inside CPU at an earlier stage.

(c) Control instructions : The programme instructions are arranged in the memory in the order of their execution. A program counter, inside CPU is normally incremented by one to recall the instructions one by one from the memory. The control instructions are required to alter the normal recalling of instructions to the CPU.

3.02 ARCHITECTURE OF CENTRAL PROCESSING UNIT :

The heart of the microprocessor based system is the central processing unit. In the following paragraphs is defined the architecture of a typical CPU Intel 8008 which is simpler and easier to understand. The other CPU's essentially follow the same architecture with few changes here and there.

Scratch pad registers : CPU contains 7 nos. of scratch pad registers each of 8 bit length. The registers are labelled as A,B,C,D,E,H and L. Each register is a signed a three bit binary code, which can be used to address any register. The codes as used in Intel 8008 CPU are given below :

Register	Binary code	Register	Binary Code
A	000	E	100
B	001	H	101
C	010	L	110
D	011	H-LL	111

A separate code has been provided for addressing H-L pair of registers, which can be used to store data of 16 bit length in the CPU.

3.03 INSTRUCTION SET :

As described earlier the complete set of instructions can be divided into the following three categories :

1. Transfer instructions
2. Accummulator group instructions
3. Control instructions

Each instruction is represented in the short form called Mnemonic.

3.03.1 TRANSFER INSTRUCTIONS :

(a) Data transfer inside CPU : An instruction to transfer the content of scratch pad register B to register D is shown below :

M O V D,B	1 1	0 1 1	0 0 1
Mnemonic	op code for the move instruction	Code for the destination register	Code for source register B

(b) Data transfer from CPU to memory : Whenever the memory has to be addressed it needs a 16 bit address (8 bit low order and 6 bit high order and 2 bit control code with the high order address). In order to transfer the data from CPU to memory first the memory address from the CPU has to be sent to the memory. This memory address is stored in H-L pair of registers by a preceeding instruction. An instruction to transfer the data from scratch pad register C to memory is shown below :

M O V M,C	1 1	0 1 0	1 1 1
Mnemonic	op code for move instruction	Code for reg. C	Code to recall H-L pair of register 1.

(c) Data transfer from memory to CPU : In this case also the memory address, where the data is available, is preloaded in H-L pair of registers by a preceeding instruction. A typical instruction is shown below :

Mnemonic	op code	code for 'D'	Code for memory
M O V D,M	1 1	0 1 1	1 1 1

The operation content of memory moved to register D is shown symbolically as (D) ⟵ (M).

(d) Immediate data instructions : The move instruction (CPU memory) described earlier needed preloading of H-L pair of registers with the memory address. The move immediate instructions are 2 byte instructions, designed to eliminate the preloading of H-L pair of registers.

Mnemonic	op code	destination	op code
MVI B	0 0	0 0 1	1 1 0
Data	b b	b b b	b b b

The operation in the symbolic form is shown below :

$$(B) \longleftarrow (\text{Data})$$

Here the data to be transferred from the memory to CPU is stored in the program itself, in a location which is next to the instruction. It can also be seen that there is no source for the more immediate instruction.

Hence the last three bite, normally reserved for the source code, have also been used to specify the op code. Move immediate instructions be used to store data directly in memory.

Mnemonic	op code	code for memory	op code
MVIM	0 0	1 1 1	1 1 0
DATA	b b	b b b	b b b

In this case the source and destination both are in the memory. The data stored in the program has to be transferred to some other location in the memory whose address has been preloaded in H-L pair of registers.

(e) Increment and decrement instructions for the scratch pad registers : The formats for the above two instructions are given below :

I N B	0 0	0 0 1	0 0 0
D C D	0 0	0 1 1	0 0 1

The opertions are as shown below :

$$(B) \leftarrow (B) + 1$$
$$(D) \leftarrow (D) - 1$$

3.03.2 ACCUMULATOR GROUP INSTRUCTIONS :

The scratch pad register a serves as the accumulator. Accumulator is also the destination register for accumulator group instructions. Hence the space allotted for destination code becomes part of the op code. The general format of the accumulator group instructions is as given below :

op code	Function	Source
b b	b b b	b b b

The central processing unit contains an Arithmatic logic unit for carrying out the arithmatic and logic operations.

The arithmatic logic unit in Intel 8008 can do the following operations :

a. Addition

b. Subtraction

c. Logic functions AND, EXCLUSIVE OR, OR and Compare.

The data in the source register are operated on by the arithmatic logic unit and the result is stored back in the accumulator. The source register could be either a scratch pad register or a register in the memory.

Several types of questions can be asked about the content of accumulator, and the answer to those questions are indicated by FLAGS. The arithmatic logic unit of Intel 8008 permits the following four flags.

<u>SIGN</u> : The most significant bit (MSB) of the 8 bit word indicates the sign. The SIGN flag is TRUE or ON when the MSB of the word stored in accumulator is '1'. The magnitude of the word is given by the remaining 7 bits.

<u>CARRY</u> : Consider the case when two words of 8 bits are added and the result is a 9 bit word or in other words a 'carry' is generated. Similarly consider the case when a larger number is subtracted from a smaller number and a 'Borrow' is generated. In both the conditions 'CARRY' flag is true.

__ZERO__ : The zero flag is TRUE when the content of the accumulator is zero.

__PARITY__ : When the word in accumulator contains even number '1', the parity flag is TRUE.

The accumulator group instructions for Intel 8008 are summarised in Table

__Mnemonic__	__Format__	__Function__	__Comment__
ADD r	10 000 SSS	(A)←(A) + (r)	r is unaffected
ADM	10 000 111	(A)←(A) + (M)	Memory address is preloaded in H-L
ADI	00 000 111	(A)←(A) + (data)	2 byte instruction
	b bbbb bbb		

In the above addition operation carry is not taken into account :

ADC r	10 001 SSS	(A)←(A) + (r) + (c)
ADC M	10 001 111	(A)←(A) + (M) C (c)
ACI data	00 001 100	(A)←(A) + (Data) + (c)

In the above operations carry bit as indicated by the carry flag is also added.

SUr	10 010 SSS	(A)←(A)− (r)
SUM	10 010 111	(A)←(A) − (M)
SUI	00 010 100	(A)←(A) − B
	(B2)	

SBr	10 010 SSS	(A)←(A) − (r) − (borrow)	
SBM	10 011 111	(A)←(A) − (M) − (borrow)	
Mnemoic	Format	Function	Comments
SBI	00 011 100	(A)←(A) − (B2)− (borrow)	
	(B2)		

The borrow is also indicated by the carry flag.

ANDr	10 100 SSS	(A)←(A). (r)	
ANDM	10 100 111	(A)←(A). (M)	AND
ANDI	00 100 100	(A)←(A). (B2)	Operation
XRr	10 101 SSS	(A)←(A) ⊕ (r)	Exclusive
XRM	10 101 111	(A)←(A) ⊕ (M)	OR
XRI	00 101 100	(A)←(A) ⊕ (B2)	Operation
ORr	10 110 SSS	(A)←(A) + (r)	OR
ORM	10 111 0111	(A)←(A) + (M)	Operation
ORI	00 110 100	(A)←(A) + (B2)	
CPr	10 111 SSS	(A) - (r)	Compare
CPM	10 111 111	(A) - (M)	operation
CPI	00 111 100 (B2)	(A) - (B2)	The result of the sub~~s~~truction operation is not loaded back in accumulator. HENCE (A) remains unchanged

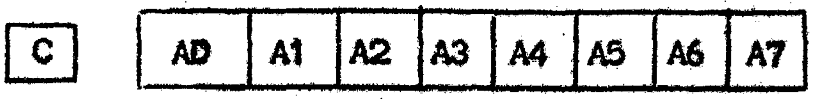
In compare operation, the function as indicated against each operation is carriedout and the result is stored in accumulator . It is possible to obtain separate indications for the following three conditions, by testing appropriate flags as indicated against each condition.

	Zero	Carry
(A) > (Source)	0	0
(A) < (Source)	0	1
(A) = (Source)	1	0

<u>Rotate accumulator instructions</u> :

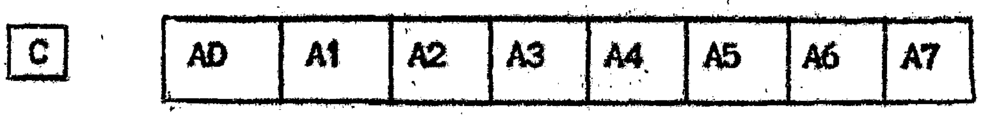
These instructions can be used to shift the content of the accumulator either alone or within the carry bit, to the right or to the left by one bit.

RAL, RAR



Rotate with carry

RLC, RRC



Rotate without carry.

The instructions are summarised below :

<u>Mnemonic</u>	<u>Format</u>	<u>Function</u>	<u>Comments</u>
RLC	00 000 010	(Am+1) ← Am (AO) ← (A7) (Carry) ← A7	Shift left
RRC	00 001 010	(Am) ← AM-1 (A7) ← (AO) (Carry) ← (AO)	Shift right
RAL	00 010 010	(Am+1) ← (Am) (AO) ← (Carry) (Carry) ← (A7)	Shift left into carry
RAR	00 011 010	(Am) ← (Am+1) (A7) ← (Carry) (Carry) ← (AO)	Shift right with carry

3.03.3 CONTROL INSTRUCTIONS :

As explained earlier the program counter, a 14 bit register, stores the address of the next byte to be brought to the CPU. The program counter is normally incremented by 1, after each byte is fetched. The control instructions can alter the content of the program counter. The following instruction are included in the set of control instructions.

a. Unconditional JUMP

b. Conditional JUMP

c. CALL and RETURN instructions

(a) UNCONDITIONAL JUMP : The format and the function is as shown below :

Mnemonic	Format	Function
JMP	01 xxx 100	(PC) ← (B3) (B2)
	(B2)	i.e. B2, B3 entered
	(B3)	in program counter.

It is a 3 byte instruction. The next instruction to be brought to the CPU shall be from memory location whose lower order address is (B2) while the higher order address is (B3). In the execution phase of the instruction the content of program counter PC is replaced by (B3)(B2).

(b) CONDITIONAL JUMP : These are also 3 byte instructions.The instructions along with their function are summarised below :

Mnemonic	Format	Function
JTc	01 1 C4 C3 000	Jump to location (B3) (B2)
	(B2)	if the condition flag is t
	(B3)	
JFc	01 0 C4 C3 000	Jump to location B3,B2 if
	(B2)	condition flag is flase.
	(B3)	

In the earlier sections four conditions flags i.e. zero carry,parity and sign were discussed. Each flag is specified by two control bite C4 C3 which are as given in the following Table

FLAG	TRUE CONDITION	C4 C3	Mnemonic
ZERO	When (A) is zero	00	JTZ
CARRY	When there is overflow or underflow.	01	JTC
PARITY	Even number of 1's in (A)	10	JTP
SIGN	When A7 or MSB is 1 in (A)	11	JTS

(c) <u>CALL INSTRUCTION</u> : Call instruction is different from the jump instructions defined above. Whenever a call instruction comes, the contents of program counter are saved in a special memory inside CPU called the STACK, so that one can return to the main program after executing the Call subroutine. The Call instruction is also a 3 byte instruction. Therefore if first byte of the CALL instruction is located at (PC), then (PC) + 3 may have to be saved in order to come back to the main program after executing the subroutine. The typical CALL and RETURN instructions are shown below :

<u>UNCONDITIONAL CALL INSTRUCTION</u> :

Mnemonic	Format	Function
CALL	01 xxx 110	Stack $\leftarrow$ (PC)
	(B2)	(PC) $\leftarrow$ (B3) (B2)
	(B3)	

Let (PC) = x, where the first byte of CALL instruction is stored. When the third byte of the CALL instruction is brought to CPU, (PC) = X + 2. At this stage, program counter as usual shall be incremented by 1, i.e. (PC) = X + 3. Now the program counter contents shall be saved in a special memory called STACK, and the program counter shall be loaded with B3, B2 which is the address of the first instruction of the subroutine.

<u>CONDITIONAL CALL INSTRUCTIONS</u> :

Mnemonic	Format
CTc	01 1C4 C3 010
	B2
	B3
CFc	01 0C4 C3 010
	B2
	B3

In case of CTc, if the flag c is true i.e. (c) = 1 then, the next two byte of instruction are brought to the CPU, and specified functions will follow. Otherwise the data bytes B2, B3 will not be brought to the CPU, and the instruction stored at (PC) + 3 shall be brought to the CPU for execution.

(d) <u>RETURN INSTRUCTIONS</u> : The CALL and RETURN instructions are operated in pair. The RETURN instructions are summarised as below :

Mnemonic	Format	Function
RET	00 xxx 111	(PC) $\rightarrow$ (Stack)
RTc	00 1 C4C3 011	If (C) = 1 then
		(PC) $\rightarrow$(Stack) otherwis
		(PC) $\rightarrow$(PC) + 1
RFc	000 C4C3 011	If (C) = 0 then
		(PC) $\rightarrow$(Stack) otherwis
		(PC) $\rightarrow$(PC) + 1

In case of conditional return instructions, if conditions are satisfied, the program counter content PUSHED in stack, is POPPED OFF from the stack and stored in program counter, to enable the CPU to return to the main programme. If conditions are not satisfied, the next instruction in the subroutine is executed.

(e) <u>HALT INSTRUCTION</u> : This instruction causes the CPU to go in STOPPED state and remain there until receipt of an INTERRUPT signal.

Mnemonic	Format	Function
	00 000 00x	(PC) $\rightarrow$ (PC) + 1
	11 111 111	CPU Stopped state

The program counter points to the next instruction.

3.03.4 <u>PROGRAM COUNTER</u> :

The memory addresses where the current instruction is available, is stored in a special register called Program counter. The program counter is a 14 bit wide register. The memory addresses is 14 bit wide and consists of a 6 bit Higher order address and a 8 bit lower order address. The memory address is sent to the memory on 8 bit wide bus in two parts. First the 8 bit lower order address and then the 6 bit higher order address is sent. The balance two bits in the high order address are called the control bits. The control bits occupying D6, D7 positions in the data bus decide the control cycle of the CPU.

3.04 CENTRAL PROCESSING UNIT ON A CHIP :

Intel 8008 microprocessor chip is shown in fig. 3.4

The CPU required two clock ϕ_1 and ϕ_2 and sync pulses at the SYNC terminal from a clock generator for synchronization of the complete system. The pulses at ϕ_1 , ϕ_2 and SYNC. are shown in fig. 3.5 .

Instruction cycle and machine cycles :

The time required to fetch and execute an instruction is called the Instruction cycle.

Every Instruction cycle consists of a number of machine cycle. A machine cycle is required each time the CPU access the memory or an I/o part.

Each machine cycle consists of a number of states (T1 to T5). The time period of each state is determined by the time period of clock pulses ϕ_1 and ϕ_2 . Begining of a new machine cycle is synchronized with the rising edge of ϕ_1 in state T1, which is also the first state of any machine cycle.

Execution of an instruction :

A basic instruction cycle consists of two phases :

1. FETCH Phase
2. EXECUTE Phase.

States T1 and T2 of any machine cycle are associated with the FETCH phase. The state T3 is reserved for actual transfer of data. The states T4 and T5 are reserved for actual execution of the instruction inside the CPU. The 8 bit low order address to the memory is sent during T1, while the 8 bit high order address is sent during T2. The type of machine cycle is indicatec by the control bits D6 D7 sent during T2 along with high order address. The function performed in each machine cycle and the control bits assigned to them are shown in the table given below:

D6 D7	Cycle code	Function performed
0 0	P C I	READ Ist byte of instruction
0 1	P C R	READ additional byte of instruct
1 0	P C C	Designate the data as I/O comman
1 1	P C W	WRITE the data bits in memory

The control bits are used by external circuitry to provide MEMORY WRITE And MEMORY READ Command.

Consider - for example the execution of MOVD, M instruction.

	Type of machine Cycle	State	Operation
M1	PCI	T1	Low order address of the instruction to memory
		T2	High order address of the instruction to memory
		T3	Instruction MOV D,M brought to CPU
M2	PCR	T1	(L) sent to memory
		T2	(H) sent to memory
		T3	Data read from memory and sent to CPU bus
		T4	Data latched in temp.reg.b
		T5	(b) transferred to D

As apparent the instruction cycle consists of two machine cycles (PCI and PCR). PCI cycle consists of only 3 states, while PCR cycle consists of five states. A machine cycle always begins with state T1.

Beginning of a new state coincides with the rising edge of $\emptyset_1$. The rising edge of SYNC pulse coincides with the rising edge of $\emptyset_2$. SYNC pulse identifies the first state T1 in every machine cycle.

The internal state of the CPU is indicated at S0, S1 and S2 terminals. These state signals are also used to synchronise the entire system. The state signals at S0, S1, S2 for various states are shown below :

S0	S1	S2	State	
0	1	0	T1	Low order address
0	1	1	T1I	CPU interrupted
0	0	1	T2	
0	0	0	Wait, memory not ready	
1	0	0	T3	
1	1	0	Stopped'Halt' state	
1	1	1	T4	
1	0	1	T5	

(a) <u>INTERRUPT</u> : Many real world situations like power failure, human decisions etc. require forced entry of instructions in CPU. An interrupt signal (active high) at the Interrupt terminal of CPU interrupts the normal flow of program. After the interrupt needs of the external system have been met, the normal flow of programme resumes. Since interrupt request comes from the external circuit, it may appear in the middle of an instruction. In such a case the CPU completes the current instruction and then acknowledges the receipt of interrupt request through the state signals (S0,S1,S2). That is once the current execution is over, the machine cycle to follow has its first state designated as T1I. The program counter at this stage points to the next instruction in the normal programe.

At this point a special instruction called RESTART instruction is zammed into CPU by the external device requesting the interrupt..

```
Mnemonic
RST               00 AAA 101   (Stack) ← (PC)
                               (PC)    ← 000 000
                                         00 AAA 000
```

The Restart instruction, allows the CPU to obtain the programme to be executed during the Interrupt period from memory location whose address has been loaded in the program counter during the execution of Restart instruction. The last instruction in the interrupt subroutine a is a Return instruction, which allows the normal program flow to resume.

(b) WAIT STATE : When memory is presented with address, it takes some time to output or Input the data. In slower memories, if this time, also known as the access time, is larger than the time of one state, the memory can request a wait state to the CPU, by pulling the READY line low. Since the request is coming from memory, the request can be synchronised with the ϕ_2 pulse during T2, that is when addresses have been transmitted and they have been allowed sufficient time to stabilise. The CPU responds by outputting state TW, instead of state T3. In the wait state CPU idles. The CPU comes out of the WAIT state, when the Ready line goes high during a ϕ_2 pulse. CPU acknowledges the receipt of signal on Ready line through state signals.

3.1 ARCHITECTURE OF CENTRAL PROCESSING UNIT :

The architecture of Intel 8080 is different from Intel 8008 in the following manner :

(a) Intel 8080 also has the same number of scratch pad registers (A, B, C, D, E, H, L). However the codes for them are different. In 8008 only scratch pad H and L registers could be accassed together, while in 8080 B and C, D and E and H and L registers can be accessed in pairs.

Codees for Registers	
A	111
B	000
C	001
D	010
E	011
H	100
L	101

Table 1

Codes for Register pairs	
B-C	00
D-E	01
H-L	10
S P	11

Table 2

(b) The last entry in Table 2 is for a register called stack pointer. Intel 8080 has the sack stack memory located inside An external Random access memory addressed from the stack pointer; a 16 bit register stores the address of the memory location filed last.

(c) Intel 8080 has a seperate 16 bit bus for sending the memory address, besides the 8 bit data bus.

(d) There are five flags in Intel 8080. The additional flag is called auxiliary. Which can be understood with the following example.

Intel 8080 has one instruction called D A A which is defined as below:

Mnemonic	Code
D A A	010 111 111

The Decimal adjust accumulator operation is defined to adjust the eight bit number in the accumulator to form bit numbers The instruction is useful while working with decimal numbers.

Consider the following example :

5.7		0 1 0 1	0 1 1 1
7.6		0 1 1 1	0 1 1 0
13.3	Add 6 to readjust	1 1 0 0	1 1 0 1
		0 0 0 0	0 1 1 0
		1 1 0 1	0 0 1 1

When the value of least significant 4 bits is more than 9, as in the above case, the readjustment can be done by adding 6 to the least significant bits.

The most significant 4 bits are also adjusted in the same manner except that the carry flag is checked instead of the auxiliary carry flag.' The flags and their codes are shown in the following table.

Flag	C_5	C_4	C_3
NOT ZERO	0	0	0
ZERO	0	0	1
NO CARRY	0	1	0
CARRY	0	1	1
PARITY ODD	1	0	0
PARITY EVEN	1	0	1
PLUS	1	1	0
MINUS	1	1	1

The auxiliary carry flag cannot be specified. The most significant bit indicates the sign of the number.

Intel 8080 CPU CHIP is a 40 pin device which is used for the following:

Designation	Number of pins	Remark
A_0 to A_{15}	16	Address bus
D_0 to D_7	8	Data bus
+ 12V		
+ 5V	4	Power supply and ground
- 5V		
Gnd		
READY		

INT CLOCKS RESET	6	Control inputs
HOLD INTE DBIN $\overline{WR}$ SYNC WAIT HALDA	6	Control output
Total	40 pins	

Intel 8080 has a seperate 16 bit address bus which is loaded by the memory address . The addresses remain stable until the Ist pulse after T_3.

TYPES OF MACHINE CYCLE AND STATUS WORD

The following types of machines cycles may occur within an instructions in Intel 8080.

(1)	FETCH INSTRUCTION	:	Same as PCI of 8008
(2)	MEMORY READ	:	Same as PCR of 8008
(3)	MEMORY WRITE	:	Same as PCW of 8008
(4)	STACK WRITE	:	
(5)	STACK READ	:	
(6)	INPUT	:	Same as PCC of 8008
(7)	OUTPUT		
(8)	INTERRUPT	:	
(9)	HALT		
(10)	HALT.INTERRUPT		

The cycles at 4,5,8,9 and 10 are additional. Since 8080 has a seperate address bus, data bus is free during T_1 and T_2.

Status Signal : The data bus D_0 to D_7 is also used to transmit status information which indicates the external circuitary the type of machine cycle currently running at any time in the CPU.

The status information is available on the bus during first state i.e., T_1 of every machine cycle and during the SYNC internal. The data bus is available free during T_1 and T_2 for such an operation. Each data bit is assugned a specific function which are detailed below:

Bit	Symbol	Definition
D_0	INTA	Acknowledge signal for interrupt request. The signal is used to latch a RESTART instruction when DBIN is active.
D_1	$\overline{\text{WO}}$	Indicates a WRITE memory or OUTPUT function, otherwise a READ MEMORY or INPUT Operation will be executed.
D_2	STACK	Indicates that the address bus holds the pushdown stack address from the stack pointer.
D_3	HLTA	Acknowledge signal for HALT instruction
D_4	OUT	Indicates that the address bus contains the address of an output device.
D_5	MI	CPU is in the fetch cycle for the first byte of an instruction.
D_6	INP	Indicates that the address bus contains the address of an input device and input data can be placed on the data bus when DBIN is active.
D_7	MEMR	The data bus will be used for memory read data.

Control inputs and Outputs and SYNC : A set of non-overlapping pulses (ϕ_1,ϕ_2) are used to synchronise all the operations. The rising edge of ϕ_1 is the beginning of new state. The SYNC pulse output from the CPU identifies the beginning of a new machine cycle. SYNC pulse is triggered by the rising edge of ϕ_2 and ends with the rising edge of ϕ_2 in the next state.

Ready and Wait : When READY to CPU is made low, the CPU goes in the WAIT state. As long as the ready input is low the processor remains in the wait state TW. Ready input is used essentially by slow memories which take more time to output. However, the request must be received during state T_2 to enable the CPU to acknowledge the request by outprinting WAIT signal. The WAIT signal will get reset during state T_3.

INT and INTE : Interrupt request is asynchronous and may originate at any time. The request is however acknowledged if it persists till INTE signal appears, in the last state of a cycle. Once this happens an interrupt m-achine cycle follows which is a normal machine cycle with five states. As usual the status word is transmitted on data bus, which however accompanied b an INTA status bit. This signal can be used by the external device to jam a RESTART instruction during T_3 which tells the processor the location of the programme.

DBIN: DBIN signal from microprocessor is used by external devices to enable the transfer of the data to CPU. The signal is available in the following machine cycles.

1. FETCH
2. MEMORY READ

3. STACK READ

4. INTERRUPT : for reading the Restart instruction from the external device.

DBIN is initiated by the rising edge of ϕ_2 during state T_2 and terminated by the corresponding edge of ϕ_2 during T_3. Any TW state, between T_2 and T_3 will therefore extend DBIN by one or more clock periods.

$\overline{WR}$: Similar to DBIN, CPU generates a $\overline{WR}$ signal which can be used by the external devices to enable transfer of data from CPU during T_3. The signal is available in the following machine cycles. The leading edge of $\overline{WR}$ signal is synchronized by the leading edge of ϕ_1 in a state following T_2 and the trailing edge of $\overline{WR}$ signal with the leading edge of ϕ_1 in a state following T_3.

CHAPTER 4

MICROPROCESSOR BASED SYSTEM DESIGN

4.1 HARDWARE DESIGN

4.1.1 MICROPROCESSOR SYSTEM

A micro-computer system using INTEL 8080 would consist of the following components.

1. Central Processing Unit chip 8080 : Which is heart of the system.
2. Clock generator 8224 : Which provides nonoverlapping clock pulses ϕ_1 , ϕ_2 to the C.P.U.
3. System Controller 8228 : Which buffers the data bus and also provides control signal (control bus 6 bit wide) using status word form data bus and DBIN,WR, HLDA from C.P.U.
4. Address bus buffers (optional) 8212 : The address bus can not provide sufficient current and hence the address bus has to be buffered. Intel 8212 load the address bus by 0.25 mA and provide a maximum loading of 75 mA at its output terminals.
5. Programmable Read only memories PROMS.
6. Random access memories RAMS
7. Input/Output interface devices

8. (a) 8251 for interfacing with teletype CRT,terminal etc.
   (b) 8255 for interfacing with a A/D converter.
8. Programmable Interrupt controller (8259, 8214): To manage the interruption of C.P.U. by a number of devices and to latch RESTART instruction.

9. Programmable DMA controller (8257) : To manage DMA service for a number of periplreal devices.

Description of Individual block :

1. Central Processing Unit :

CPU 8080 has been described earlier.

2. Clock Generator (8224) : (?Ref. fig. 4.1.2)

Oscilator frequency is 9 times the speed at which the microprocessor has to be run. The clock pemiod (tcy) is specified for the microprocessor. Consider, for example, 8080 A CPU chip for which tcy can be from 0.48 H Sec to 2.0 M Sec.

Considering the fastest speed, i.e. tcy = 0.48 micro-second.

$$\text{Crystal frequency} = \frac{1}{0.48} \times 9 \text{ MHz}$$

$$= 18.8 \text{ MHz}$$

Standard crystal of 18.432 MHz is available which gives a tcz = 0.49 micro second.

Clock generator consists of a devide by 9 counter whose four outputs Q_A to Q_B are combined in logic gates to obtain ϕ_1 and ϕ_2 clock pulses for CPU which are shown in fig. 4.1.3.

ϕ_2 TTL available at pin 6 which is similar to ϕ_2 except for a small delay, can be used for external timing purposes. It is especially useful in DMA dependent activities.

ϕ_2 (TTL) can be used to gate the requesting device on the bus once CPU issues HOLD ACKNOWLEDGE (HLDA) signal.

The clock generator also generates $\overline{\text{STSTB}}$ (status strobe) signal which is used to latch the status word on the control bus refer Fig. 4.1.4. The control bus is connected to the data bus through the SYSTEM CONTROLLER (8228). The states signal is available on the data bus during the SYNC pulse.

Automatic system reset feature on power on can be incorporated by adding components R_1, C_1 and D_1 in the circuit. The $\overline{\text{RESIN}}$ input to the Reset flip is through a Schmidt trigger which converts the slow transition into a clean fast edge for triggering the D flip flop.

3. <u>System Controller (8228)</u> : (Ref. Fig.4.1.5 and 4.1.6)

System controller generates all control signals required to directly interface RAM, ROM, and I/O components.

An eight-bit, bidirectional bus driver buffers the data bus from memory and Input/Output devices. Data bus which has an input requirement of 3.3 volts (min.) and can drive (sink) a maximum current of 1.9 mA is assured by data bus driver to fulfil these input requirement even in enhance moise immunity. Due to the availability of adequate driving current (10 mA typ.) on the system side of the driver, a large number of memory and I/O devices can be directly connected to the bus.

When $\overline{\text{STSTB}}$ input goes low, the status information that is issued by CPU to its data bus at the beginning of each machine cycle gets stored in the status Latch.

The Gating Array generates control signals ($\overline{\text{MEMR}}$, $\overline{\text{MEM W}}$, $\overline{\text{I/O R}}$, $\overline{\text{I/O W}}$ and $\overline{\text{INTA}}$) by gating the outputs of the status Latch with signals from the 8080 CPU (DBIN; $\overline{\text{WR}}$ and HLDA).

The $\overline{\text{BUSEN}}$ (Bus Enable) input to the Gating Array is an asychronous input that forces the data bus output buffers and control signal buffers into their high-impedance state if it is a 'one'. If BUSEN is a 'zero' normal operation of the data buffer and control signals take place.

<u>Programmable Communication Interface (8251)</u> :

It is used as a peripheral device and is programmed by CPU to operate using virtually any serial data transmission technique. Its functional configuration is programmed by the system's software for maximum flexibility.

In the problem, a CRT is used which communicate with CPU through this interface.

<u>Programmable peripheral interface (PPI 8255)</u> :

A block diagram of the device is shown in figure 4.1.7. The device is connected to the data bus through a 3 state data bus buffer. The output terminals of the device have been arranged in 4 ports (A, supper C, lower C, B). These ports have been arranged in two groups. Group A consists of Port A and Upper C while groub B consists of lower C and port B.

The device has a register to store the control word, which decides the manner in which the ports will behave. Besides this the device has a number of control inputs ($\overline{RD}$, $\overline{WR}$, Ao, Al, Reset and $\overline{CS}$).

The data is received or transmitted upon execution of IN port or OUT port instructions by the CPU, which are described as below :

IN port $(A) \leftarrow (data)$

11	011	011
	port	

The data placed by the specified port on the data bus is moved to Accummulator.

Similarly in OUT port instruction, data from the accumulator is put on the data bus for transmission to a specified port.

In both the cases the port is specified by the 2nd byte of the instruction becomes the lower order part of the address (A7 to Ao), upon execution of the instruction.

Ao	Al			
0	0	Port A	$\Leftrightarrow$	Data bus
0	1	Port B	$\Leftrightarrow$	Data bus
1	0	Port C	$\Leftrightarrow$	Data bus
1	1	Data bus	$\Rightarrow$	Control register

The control word sent from CPU through an OUT port instruction decides the mode in which various ports of 8255 operate.

There are three basic modes of operation in which the ports can operate. (Refer fig. 4.1.8).

Mode 0 —	Basic Input/Output
Mode 1 —	Strobed Input/Output
Mode 2 —	Bidirectional bus

It is also possible to set/reset any of the eight bits of port C by using a single OUT port instruction (Refer Fig.4.1.

When Reset input goes high all ports will be set to the Input mode (in the high impedance sot). After the RESET is removed 8255 can remain in the Input mode with no additional initialization required.

APPLICATIONS

MODE 0 OPERATION : When operated in Mode 0 the basic features are given below :-

1. Two 8 bit port and two 4 bit ports.
2. Any port can be input or output.
3. Outputs on the ports are latched.
4. Inputs on the ports are not latched.

Mode 0 is ideally suited to communicate simple switch closings to the CPU. Depending upon the control word number of configurations of the ports are possible (Refer Annexure 1 for details).

<u>Mode 1 Operations</u> :- In mode 1 the basic features are as given below :(Refer Fig.4.1.10).

1. Ports operate in groups only (Group A and Group B).

2. In group A, Port A operates as 8 bit data port while upper C acts as control/data port. Consider for example, the case when port A operates as an input port. A low on PC4 loads data into the input latch. A high on PC5 indicates that the data is loaded in the input latch although it has yet to be off loaded by CPU. When CPU of loads the data by accessing port A, PC5 goes low again. It is also possible to generate an Interrupt request for the CPU by the peripheral device connected on Port A. 8255 contains an INTE A flip flop which can set/reset by PC4, when CPU sends the following control word.

0 x x x 1 0 0 1 for setting the INTE flip flop

0 x x x 1 0 0 0 for resetting the flip flop.

As a first step the peripheral device on port A latches RST instruction on the port. As a result IBFA goes high. If INTE flip flop has been SET by CPU earlier, an INTERRUPTS the CPU. During T3 of the interrupt cycle RST instruction is off loaded. RD pulse also resets INTEA flip flop.

PC6, 7 the balance terminals of upper C port can be programmed as either input or output port as the case may be.

In a similar manner port B, and lower C port can be used to generate an Interrupt request.

In mode 1, port A or port B can be operated as output, ports.

The OBF output goes low to indicate that CPU has written data in the port. A low on $\overline{ACK}$ input informs 8255 that the data has been accepted by the peripheral device. It is also possible to generate an Interrupt request by the peripheral device connected on the specified port.

Once the peripheral device has accepted the data both OBF and ACK are high. At this stage if INTE flip flop is SET by sending an appropriate control word, Interrupt request shall be generated. In this case RST instruction shall be loaded by some other peripheral device connected on an Input port. Refer Table 3 in Annexure 1 for more details.

MODE-2 Operation

The basic features of Mode 2 operation are as given below : (Ref. Fig. 4.1.11).

1. Mode 2 is used in group A only (Port A and 5 bit upper C) for control.
2. In mode 2 both inputs and outputs are latched. In mode 2 port A operates as a bidirectional bus.

The terminals OBF, $\overline{ACK}$, $\overline{STB}$, $\overline{IBF}$ have their usual meaning. The interrupt request is generated at PC3.

Reading Port C status :

When the 8255 is programmed to function modes 1 or 2 Port C generates or accepts handshaking signals with the peripheral devices. Reading the status of Port C allows the programmer to verify the status of each peripheral device by sending a simple IN port C instruction. The word can be finally interpreted.

Example :

Consider the case when 12 bit D/A converter has to be connected as an output device while an 8 bit A/D converter as an Input device. The software for initialisation of the ports is as shown in fig. 4.1.12.

MVI, A	8E	This makes port A OUt,Upper C
82	82	Out,lower C-out,Port B-Inp
OUT	D3	in Mode 0.
Control register	17	

When CPU wants to request A/D converter for a data it sends a strobe pulse to A/D converter on PC3. The conversion takes place and the digital word is stored in the A/D converter. After some time CPU sends sample enable pulse at PC2 and data is communicated to CPU on the data bus.

MVI A	3E	Bit PC3 of port C is set and
07	0 7	then reset to produce a sharp
OUT	D 3	pulse at PC3, which is the
PORT C	1 6	strobe pulse to A/D converter.
DCR A	3 D	
OUT	D 3	
PORT C	1 6	

As a next step CPU would sent a pulse at PC2 to enable the sample on the data bus of CPU.

```
MVI A          3 E
05             0 5
OUT            D 3
Port C         1 6
DCR A          3 D
OUT            D 3
Port C         1 6
```

In a similar manner software programmes can be written to OUT put data through 12 bit D/A converter.

ANNEXURE - 1

FOR SDK - 80 KIT ONLY   TABLE - 1

BYTE 2	BYTE (HEX)	SELECTION
x x x 110 x x	18	8251
x x x 1 0 1 0 0	14	Port A/8255 (1)
x x x 1 0 1 0 1	15	Port B/8255 (1)
x x x 1 0 1 1 0	16	Port C/8255 (1)
x x x 1 0 1 1 1	17	Control register/8255 (1)
x x x 0 1 1 0 0	OC	Port A/8255 (2)
x x x 0 1 1 0 1	OD	Port B/8255 (2)
x x x 0 1 1 1 0	OE	Port C/8255 (2)
x x x 0 1 1 1 1	OF	Control register/8255 (2)

By sending OUT Port or IN port instruction any port (A,B,C) or the control register can be accessed. In case of control register only OUT port instruction is valid,because the data stored in Control register cannot be read.

<u>MODE-0 Operation</u> Table-2

CONTROL BINARY	WORD	HEX	STATUS OF GROUP A PORTS		STATUS OF GROUP B PORTS	
			GROUP A	UPPER C	GROUP B	LOWER C
1 0 0 0	0 0 0 0	8 0	OUT	OUT	OUT	OUT
1 0 0 0	0 0 0 1	8 1	OUT	OUT	OUT	INP
1 0 0 0	0 0 1 0	8 2	OUT	OUT	INL	OUT
1 0 0 0	0 0 1 1	8 3	OUT	OUT	INP	INP
1 0 0 0	1 0 0 0	8 8	OUT	INP	OUT	OUT
1 0 0 0	1 0 0 1	8 9	OUT	INP	OUT	INP
1 0 0 0	1 0 1 0	8 A	OUT	INP	INP	OUT
1 0 0 0	1 0 1 1	8 B	OUT	INP	INP	INP
1 0 0 1	0 0 0 0	9 0	INP	OUT	OUT	OUT
1 0 0 1	0 0 0 1	9 1	INP	OUT	OUT	INP
1 0 0 1	0 0 1 0	9 2	INP	OUT	INP	OUT
1 0 0 1	1 0 0 0	9 8	INP	INP	OUT	OUT
1 0 0 1	1 0 0 1	9 9	INP	INP	OUT	INP
1 0 0 1	1 0 1 0	9 A	INP	INP	INP	OUT
1 0 0 1	1 0 1 1	9 B	INP	INP	INP	INP

Ports can be initialised to operate as per the design required by sending oppropriate control words as detailed above.

TABLE - 3

MODE - 1 OPERATION

CONTROL WORD		DETAILS
BINAY	HEX	
0 x x x 0 1 0 0	0 4	INTE RESET PORT B INPUT/OUTPUT
0 x x x 0 1 0 1	0 5	INTE SET PORT B INPUT/OUTPUT
0 x x x 1 0 0 0	0 8	INTE RESET PORT A INPUT
0 x x x 1 0 0 1	0 9	INTE SET PORT A INPUT
0 x x x 1 1 0 0	0 8	INTE RESET PORT A OUTPUT
0 x x x 1 1 0 1	0 D	INTE SET PORT A OUTPUT
1 0 1 1 1 1 1 x	B E	PORT A INPUT PC6,7 INPUT PORT B INPUT
1 0 1 1 0 1 1 x	D 6	PORT C INPUT PC6,7 OUTPUT PORT B INPUT
1 0 1 1 0 1 0 x	B 4	PORT A INPUT PC6,C OUTPUT PORT B OUTPU
1 0 1 1 1 1 0 x	B C	PORT A INPUT PC6,C INPUT PORT B OUTPU
1 0 1 0 1 1 1 x	A E	PORT A OUTPUT PC4,5 INPUT PORT B INPUT
1 0 1 0 1 1 0 x	A C	PORT A OUTPUT pC4,5 INPUT PORT B OUTPU
1 0 1 0 0 1 1 x	A 6	PORT A OUTPUT PC4,5 OUTPUT PORT B INPUT
1 0 1 0 0 1 0 x	A 4	PORT A OUTPUT PC4,5 OUTPUT PORT B OUTPU

4.2 SYSTEM'S OPERATION (Refer figure 4.2.1)

Start-Up of the Kit :

When power is applied initially to the 8080, the processor begins operating immediately. To make the contents of program counter, stack pointer and the other working registers zero, it is necessary to begin the power-up sequence with RESET. So, the start-up procedure is as follows :

1. Plug the system communication Monitor (CRT) into the kit.
2. Turn power on at both the kit supply and communication monitor.
3. Press the Reset switch.

At this point, monitor will display the following message :

MCS - 80 KIT

Monitor Operations :

The monitor communicates with the operator via an interactive console, here a CRT.

The monitor requies each command to be terminated by a carriage return, with the exception of the 'S' and 'X' commands, the command is not acted upon until the carriage return is sensed.

Selected areas of addressable memory may be accessed and displayed by the D command. The D command produces a formatted listing of the memory area between < low address > and < high address > , inclusive, on the console device.

Control of the CPU is transferred from the monitor to the user program by means of the program execute command, G. The < entry point >should be an address in RAM which contains on instruction in the user's program. For example G 1400 means control is passed to location 1400H.

User program is entered into RAM with the I command. After sensing the carriage return terminating the command line, the monitor waits for the user to enter a string of Hexadecimal digits (0 to 9, A to F). Each digit in the string is converted into its binary value and then loaded into memory. Two Hexadecimal digits are loaded into each byte of memory.

The M command moves the contents of memory < low address > inclusive to the area of RAM begining at < destination> . The content of source field remain undesturbed, unless the receiving field overlaps the source field.

<u>Basic Operation</u> :

After the cold start-up of the kit, the program is inserted into the RAM with the help of I < address > command. The program is stored sequentially in the memory.

On issuing the G entry point command, the execution of program stored in RAM begins and the results are displayed on CRT according to program inserted in the memory. For example in the presented work message like voltage adjust high, voltage adjust low, Frequency adjust high, Frequency adjust low, close the circuit breaker will appear in the CRT.

4.3 SOFT WARE DESIGN

```
; PROGRAM FOR AUTO SYNCHRONIZATION OF ALTERNATOR WITH BUS

; - - - - - - - - - - - - - - - - - - - - - - - - - - -

;                VOLTAGE MATCHING

; - - - - - - - - - - - - - - - - - - - - - - - - - - -

; MEASUREMENT OF BUS VOLTAGE VB

 START :   MVI   A,93
           OUT   17        ; INTIALIZE PORT A AS INPUT
           IN    14
           MOV   B,A
 VB 1:     IN    14        ; FIND INCREASING VALUE
           CMP   B
           JC    VB 1
 VB 2 :    MOV   B,A
           IN    14
           CMP   B
           JNC   VB 2

; TOLERANCE CALCULATION
           CALL  VTOLR     ;VTOLR CALCULATES TOLERANCE
; VALUES ARE RETURNED IN LOCATION VB, VB+1 (+ 5%), VB+2(-5%)
; MEASUREMENT OF GENERATOR VOLTAGE VG
           IN    15        ; INPUT FROM B PORT
           MOV   B,A
 VG 1:     IN    15        ; FIND INCREASING VALUE
           CMP   B
           JC    VG 1
```

```
VG 2 :    MOV   B,A
          IN    15
          CMP   B
          JNC   VG 2
; CHECKING FOR  VB-5%≤VG ≤ VB + 5%
          LXI   H,VB+1
          CMP   M
          JNC   VHIGH
          INX   H
          CMP   M          ; COMPARING VB+2
          JNC   STFR       ; VOLTAGE MATCHED JUMP TO FREQUENCY
                           ; MATCHING
          MVI   D,VL
          CALL  MESSAGE
          CALL  VADJL
          JMP   VB1        ; RESTART MATCHING PROCEDURE
VHIGH :   MVI   D,VH
          CALL  MESSAGE
          CALL  VADJH
          JMP   VB1        ; RESTART MATCHING PROCEDURE
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
;              FREQUENCY  MATCHING
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
STFR :    MVI   D,04
          CALL  FREQ       ; GET BUS FREQ (IN B REG.)
          CALL  FTOLR      ; GET TOLERANCE (.5%) IN FR, FR+1
                           ; (+.5%), FR+2 (-.5%)
```

```
FRM 1 :    MVI    D,02
           CALL   FREQ      ; GET GENERATOR FREQ (IN B REG)
           LXI    H,FR+1
           MOV    A,B
           CMP    M
           JNC    FHIGH     ; BUS FREQ.HIGH
           INX    H
           CMP    M
           JNC    STPH      ;FREQUENCY IS MATCHED
           MVI    D,FL
           CALL   MESSAGE
           CALL   FADJL
           JMP    FRM 1
FHIGH:     MVI    D,FH
           CALL   MESSAGE
           CALL   FADJH
           JMP    FRM 1
;--------------------------------
;          ZERO PHASE TIME ANTICIPATION
;--------------------------------
STPH :     MVI    C,00
PH 1 :     CALL   PHASE     ; VALUE IS IN B REG
           MOV    A,B
           CMP    C
           JC     PH2       ; DECREASING PHASE OBTAINED
           MOV    C,A
           JMP    PH 1
```

```
;  CALCULATE OF ZERO PHASE TIME
            CALL   CBCL      ; CLOSES CIRCUIT BREAKER
            HLT
   VH       EQU    00
   VL       EQU    0E
   FH       EQU    1C
   FL       EQU    2A
;  END OF MAIN PROGRAM
;  SUBROUTINES
;-------------------------------------------------------------
;  FREQ. SUBROUTINE
   FREQ:    IN     16        ; IN C PORT
            ANA    D
            CMP    D
            JZ     FREQ
            MVI    B,00
   F 1 :    IN     16
            ANA    D
            CMP    D
            JNZ    F 1
   F 2 :    INR    B
            IN     16
            ANA    D
            MVI    E,4
   F 3 :    DCR    E
            NOP              ; FOR RESTRICTING THE COUNT
            NOP              ; IN 8 BITS ONLY
            JNZ    F3
            CMP    D
```

```
              JZ      F2
              RET
;-------------------------------------------------------------------
; PHASE ROUTINE
PHASE:        IN      16
              ANI     04
              CPI     04
              JZ      PHASE
P 2 :         IN      16
              ANI     04
              CPI     04
              JNZ     P2
              IN      16
              ANI     02
              CPI     02
              JZ      PHASE
              MVI     B,0
P 3 :         INR     B
              IN      16
              ANI     02
P 4 :         MVI     E,04     ; DELAY FOR
              DCR     E        ; SLOWING DOWN OF COUNTING
              JNZ     P4
              CPI     02
              JNZ     P3
              RET
```

```
;-------------------------------
; MESSAGE ROUTINE
;-------------------------------
 MESSAGE:  LXI    H,BASE
           MOV    A,L
           ADD    D
           MOV    L,A
           MVI    B,OE
 MES 1:    MOV    C,M
           CALL   CO
           INX    H
           DCR    B
           JNZ    MES 1
           CALL   CROUT
           RET
 BASE:     DB     25,20,56,4F. . . .
           DB
           DB
           DB
           DB
;-------------------------------
; ADJUST ROUTINES
; OUTPUT AT PIN C7
 VADJH:    MVI    A,80     ; PC 7
           JMP    ZERO
 VADJL:    MVI    A,40     ; PC 6
           JMP    ZERO
```

```
FADJH:    MVI    A,20     ; PC5
ZERO:     OUT    16       ; START PULSE
          MVI    A,00
          OUT    16       ; STOP PULSE
          CALL   DELAY    ; GIVE TIME FOR ADJUSTMENT
          RET
; FREQUENCY TOLERANCE ROUTINE FTOLR
FTOLR:    LXI    H, FR
          MOV    M,B
          INR    B
          INX    H
          MOV    M,B
          DCR    B
          DCR    B
          MOV    M,B
          RET
FR:       DS     3        ; RESERVE 3 BYTES FOR FR,FR+1,FR+2
; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; VOLTAGE TOLERANCE ROUTINE VTOLR
VTOLR:    LXI    H,VB
          MOV    M,B
          MOV    A,B
          RLC             ; ROTATE A 3 TIMES
          RLC             ; LEFT WITHOUT CARRY
          RLC
          ANI    07
          MOV    C,A
          RRC
          ANI    OF       ; SHIFT A ONE DIGIT
```

```
            MOV     C,A     ; C CONTAINS 5% OF VB
            ADD     B
            INX     H
            MOV     M,A     ; STORE VB+5% IN VB+1
            MOV     A,B
            SUB     C
            INX     H
            MOV     M;A     ; STORE VB - 5% IN VB+2
            RET
  VB:       DS      3
; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; DELAY SUBROUTINE (FOR 5 SECONDS DELAY)
  DELAY:    MVI     B;10
            L1 :    MVI     C,C8
            L2 :    MVI     D;FR
            L3 :    DCR     D
                    JNZ     L3
                    DCR     C
                    JNZ     L2
                    DCR     B
                    JNZ     L1
                    RET
; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
; MULTI SUBROUTINE FOR 8X8 MULTIPLICATION
; ARGUMENS ARE IN A AND E REG.RESULT IN HL PAIR
  MULT :    LXI     H,0000  ; CLEAR RESULT REGISTER
            MVI     B;08    ; INITIALIZE BIT COUNTER
            MVI     D;00
  LOOP 1:   DAD    H
            RLC
            JNC    DEC
  DEC  :    DCR    B
            JNZ    LOOP 1
            RET
; --------------------------------------------------------------
; CIRCUIT BREAKER CLOSE ROUTINE CBCL
  CBCL:     MOV    A;C
            SUB    B
            STA    TEMP
            MVI    A;F
            MOV    E;B
            CALL   MULT     ; RESULT IN HL PAIR
            LDA    TEMP
            CALL   DIV      ; RESULT IN ACCUMULATOR
; A CONTAINS B* F/(C-B)
            SUI    CORR
; WAIT FOR ANTICIPATED TIME
  LOOP:     DCR    A
            NOP
            NOP
            JNZ    LOOP
; ZERO PHASE TIME IS REACHED
            MVI    D,CL
            CALL   MESSAGE
            MVI    A;92     ; INITIALIZE C PORT
            OUT    17       ; AS OUTPUT
            MVI    A;01
```

```
         OUT    16        ; CLOSE CB BK PCO
         MVI    A,00
         OUT    16
         RET
TEMP:    DS1
         END
DIV :    MOV    B,A
         MVI    C,0
L 1 :    MOV    L,A
         CMP    B
         JN     L2
         SUB    B
         INR    C
         MOV    L,A
         JMP    L 1
L 2 :    MOV    A,H
         CPI    0
         JNZ    L 3
         DCR    H
         MVI    A,FF
         SUB    B
         ADD    L
         MOV    L,A
         INR    C
         JMP    L 1
L 3 :    MOV    A,C
         RET.
```

CHAPTER 5

MATHEMATICAL MODELLING AND COMPUTER SIMULATION

5.1 INTRODUCTION

In this chapter mathematical model of machine, excitation system and governor system has been discussed. The conditions for synchronization of alternator with bus bar are simulated in order to see the circuit breaker closing to synchronize the alternator.

5.2 MATHEMATICAL MODEL OF SYNCHRONOUS MACHINE

The following theory is developed from the fundamental starting point that the machine consists of several inductively coupled circuits, the self and mutual inductance of which vary periodically with the angular position of the rotor.

Assumption for Mathematical Descriptionof Synchronous Machine

a. The machine is assumed to be magnetically linear and iron losses have been ignored.

b. The stator has balanced, sinusoidally distributed 3 phase windings. The picture repeats every 2- pole pitches.

c. Stator slot causes very negligible variation of any of the rotor circuit inductances with rotor angle.

d. Rotor magnetic circuits and all of its electrical circuits are symmetrical both about the pole axis and interpolar axis.

Figure 5.2.1 shows the schematic representation of synchronous machine (salient pole type) per pole pair basis. The electric circuits on rotor, namely K_d, K_q, are the equivalent d-axis and q-axis damper circuits, f is the machine field. The positive direction of various quantities is shown in Fig. θ is the angular position of the d-axis (field axis) from the axis of phase a measured in the direction of rotation of the machines. ψ denotes the flux linkage, e the terminal voltage and i the current of any electrical circuit.

Park's Transformation

Due to time changing nature of most of the inductance parameters, the machine voltage relations are nonlinear differential equations with time-changing coefficients and thus, are difficult to solve. The phase variables (e, i or ψ) are accordingly transformed to axis (Park) variables through the use of Park's transformation.

The machine is accordingly represented by a two axis model (as far as the phenomenon of interaction taking place across the air gap is considered, the zero-squence interaction is represented separately) as shown in Fig. 5.2.2.

Machine expressions in terms of Park's variables are as follows (here p is the d/dt operator).

$$e_d = p\psi_d - \psi_q p\theta - ri_d$$

$$e_q = p\psi_q + \psi_d p\theta - ri_q$$

$$e_o = p\psi_o + ri_o$$

$$e_{kd} = 0 = p\psi_{kd} + r_{kd}\, i_{kd}$$

$$e_{kq} = 0 = p\psi_{kq} + r_{kq}\, i_{kq}$$

where,

$$\psi_d = - L_d i_d + L_{md}(i_f + i_{kd})$$

$$\psi_q = - L_q i_q + L_{mq}\,(i_{kq})$$

$$\psi_f = - L_{md} i_d + L_f i_f + L_{md}\, i_{kd}$$

$$\psi_{kd} = - L_{md} i_d + L_{md} i_f + L_{kd} i_{kd}$$

$$\psi_{kq} = - L_{mq} i_q + L_{kq}\, i_{kq}$$

$$\psi_o = - L_o i_o$$

Motional Equation or Swing $_e$quation of machine

$$\frac{H}{\pi\, f_{base}} \frac{d^2\theta}{dt^2} = T_m - T_e - T_{fw} \qquad .. \quad (1)$$

Excitation Voltage

$$e_f = p\psi_f + r_f i_f \qquad .. \quad (2)$$

Damper winding voltage

$$\underline{e_{kd}} = 0 = p\psi_{kd} + r_{kd} i_{kd} \qquad .. \quad (3)$$

$$e_{kq} = 0 = p\psi_{kq} + r_{kq}\, i_{kq} \qquad .. \quad (4)$$

$$\psi_f = - L_{md}\, i_d + L_{ff} i_f + L_{md} i_{kd} \qquad .. \quad (5)$$

$$\psi_{kd} = - L_{md} i_d + L_{md}\, i_f + L_{kk}\, i_{kd} \qquad .. \quad (6)$$

$$\psi_{kq} = - L_{mq} i_q + L_{kkq}\, i_{kq} \qquad .. \quad (7)$$

$$e_d = (L_q i_q - L_{mq} i_{kq}) p\theta \qquad .. \quad (8)$$

Tie Line Equation

$$e_d = v_d - L_e i_q\, p\theta \qquad .. \quad (9)$$

$$e_q = v_q + L_e i_d p\theta \qquad .. \quad (10)$$

$$v_a = - \sqrt{2}\;\; v_b \sin \theta_b \qquad .. \quad (11)$$

$$v_d = \sqrt{2}\;\; v_b \sin (\theta - \theta_b) \qquad .. \quad (12)$$

$$v_q = \sqrt{2}\;\; v_b \cos (\theta - \theta_b) \qquad .. \quad (13)$$

System's Dynamical Equation

$$\frac{d\theta}{dt} = w \qquad .. \quad (14)$$

$$\frac{d^2\theta}{dt^2} = \frac{dw}{dt} = (T_m - T_e - T_{fw}) / \left(\frac{2H}{\pi f_{base}} \right) \qquad .. \quad (15)$$

From Eqn (8) and (9)

$$e_d = -(-L_q i_q + L_{mq} i_{kq}) p\theta = v_d - L_e i_q\, p\theta \qquad .. \quad (16)$$

$$\therefore \; i_q = (v_d + L_{mq} i_{kq} p\theta) / (L_q + L_e) p\theta \qquad .. \quad (17)$$

Similarly

$$i_d = (L_{md}p\theta\ (i_f+i_{kd})-v_q)/\ ((L_d+L_e)p\theta) \quad .. \quad (18)$$

$$=(x_{md}(i_f+i_{kd})\ \frac{p\theta}{w_s} - v_q)/\ (x_{dt}\ p\theta\ /w_s) \quad .. \quad (19)$$

Equations (2),(3), and (4) can be written in the following form

$$p\psi_f = e_q - r_f i_f$$

$$p\psi_{kd} = - i_{kd}\ r_{kd}$$

$$p\psi_{kq} = - r_{kq}\ i_{kq}$$

From Eq. (7)

$$\psi_{kq} = - L_{mq} i_q + L_{kkq}\ i_{kq}$$

Putting value of i_q from Eq. (17) and solving

$$i_{kq} = \frac{\psi_{kq} + \frac{x_{mq}}{x_{qt} w}\ v_d}{(x_{kkq} - \frac{x_{mq}}{x_{qt}})\ \ / w_s}$$

5.3 COMPUTER REPRESENTATION OF EXCITATION SYSTEMS

(i) Block Diagram

In the development of the excitation system block diagrams it has been found necessary to establish a per unit voltage base. For the following discussion, one per unit generator voltage is defined as rated voltage. One per unit excitor output voltage is the voltage required to produce rated generator voltage on the generator air gap line.

The excitation system shown in Figure 5.3.1 is representative of the majority of modern system now in service. This includes most continuously acting systems with rotating excitors.

Figure 5.3.1 shows the significant transfer functions which should be included for satisfactory representation in computer studies. Many other system types may be represented if excitation system ceiling voltage is assumed to be independent of generator terminal conditions.

In the figure , V_T is the generator terminal voltage applied to the regulator input. The first transfer function is a simple time constant T_R representing regulator input filtering. For most systems, T_R is very small and may be considered zero.

The summing point compares the regulator reference with the output of the input filter to determine the voltage error input to the regulator amplifier. Also , the voltage error input is combined with the excitation major damping loop signal Y. Most computer programs do not require an input of V_{REF}. but rather internally calculate the proper value by assuming V_T at $t = 0$ is at the proper value.

The main regulator transfer functions is represented as a gain K_A and a time constant T_A. Following this, the maximum and minimum limits of the regulator are imposed so that large input error signals cannot produce a regulator output which exceeds practical limits. The output of this block is exciter output voltage or generator field voltage E_{fd}.

Major loop damping is provided by the feed back transfer function $pK_F/(1 + pT_F)$ from the exciter output E_{fd} to previous summing point.

(ii) Mathematical Equations

Output of damping loop

$$Y = pK_F E_{fd}/(1+pT_F)$$

Output of regulator

$$X = K_A(V_{REF} - V_T - Y)/(1+pT_A)$$

(iii) State Space Representation

$$\dot{E}_{fd} = (K_A(V_{REF} - V_T - Y) - E_{fd})/T_A$$

and $\dot{Y} = (K_F \dot{E}_{fd} - Y)/T_F$

It should be emphasised that there is an inter-relation between exciter ceiling $E_{fd\ MAX}$ and regulator ceiling $V_{R\ MAX}$

If $V_{R\ MIN} < X < V_{R\ MAX}$ then $E_{fd} \equiv X$

If $X \geq V_{R\ MAX}$, then $E_{fd} = V_{R\ MAX}$ and $\dot{E}_{fd} = 0$

If $X \leq V_{R\ MIN}$, then $E_{fd} = V_{R\ MIN}$ and $\dot{E}_{fd} = 0$

5.4 COMPUTER REPRESENTATION OF GOVERNOR SCHEME

(i) Block Diagram

Block diagram of the governing scheme is shown in Fig.5.4.1

Following are the deviation of variables from initial steady state conditions:

w_{ref}	reference speed per unit
w	shaft speed, per unit
z	gate position, per unit
Z	gate opening, per unit
T_g	governor response time, seconds
T_w	water starting time, seconds
T_r	Dash pot time constant, seconds
σ	permanent speed dropp, pe~~r~~ unit
δ	temporary speed droop, per unit
T_M	shaft torque,per unit

(ii) Mathematical Equation

$$\xi = (\delta\ T_r p z)/(1+T_r p) \quad .. \quad (i)$$

$$((w_{REF} - w)/2\pi\ f_{base} - (\sigma z + \xi))/(T_g p) = z \quad .. \quad (ii)$$

$$Z = T\mu_o + z \quad .. \quad (iii)$$

$$z(1 - T_w p)/(1+0.5\ T_w p) = T_M \quad .. \quad (iv)$$

(iii) State Space Representation

From Eq (i)

$$\dot{\xi} = (\delta T_r\ \dot{z} - \xi)/T_r$$

From Eq (ii)

$\dot{z} = [((w_{REF} - w)/2\pi\, f_{base}) - (\sigma z + \phi)]/T_g$

From eq (iii)

$\dot{T}_M = (z - T_w\dot{z} - T_M)/(0.5\, T_w)$

From the diagram it is clear that

if $Z \not> T_{MAX}$, $z = T_{MAX}$ and $\dot{z} = 0$

if $Z < 0$, $z = 0$, and $\dot{z} = 0$

if $T_{MAX} > Z > 0$, only then Eq (iv) will be followed.

5.5 INITIALIZATION

We know that

$$e_a = e_d \cos\theta - e_q \sin\theta$$

and $e_d = -\psi_q\, p\theta = 0$

and $e_q = \psi_d\, p\theta$

$$= L_{md}\, i_{fo}\, w_o \qquad = L_{md}\, \frac{v_{fo}}{r_{fo}}\, w_o$$

$$\therefore e_a = -L_{md}\, \frac{v_{fo}}{r_{fo}} . w_o \sin\theta \qquad \ldots (A)$$

Also $e_a = -\sqrt{2}\, V_{To} . \sin(\theta + 0)$ ... (B)

Equalling equation (A) and (B)

$$L_{md}\, \frac{v_{fo}}{r_{fo}}\, w_o \sin\theta = \sqrt{2}\, V_{To} \sin\theta$$

$$\therefore e_{fo} = \frac{\sqrt{2}\, V_{To}\, r_{fo}}{L_{md}\, w_o} = \sqrt{2}\, \frac{V_{To}\, r_f}{X_{md}(w_o/w_{base})}$$

$$\therefore E_{fdo} = e_f/[\sqrt{2}\, r_f/X_{md}] = \frac{V_{To}}{(w_o/w_{base})}$$

From block diagram it is clear that

$$[V_{ref} - V_{To}]\, K_A = E_{fdo} = \frac{V_{To}}{(w_o/w_{base})}$$

$$\therefore \quad V_{ref} = \frac{V_{To}}{K_A\,(\frac{w_o}{w_{base}})} + V_{To}$$

5.6 FLOW CHART

A simplified programme flow chart to carry out the operation is shown in Figure 5.6 . It is checked that alternator field voltage (E_{fd}) is within the prescribed limit. After ensuring this, the machine dynamics are solved. Next, the terminal voltage is compared with bus voltage. If it is outside the operating range, excitation level of alternator is adjusted. In both the cases, when terminal voltage V_T is within or outside the specified band, the speed (w) of alternator is compared with speed of bus. If it is outside the specified band, the level of mechanical power is adjusted and machine dynamics is solved. The whole process is repeated until the voltage and frequency of machine come underthe specified zone of bus voltage and frequency respectively.

Once the machine terminal voltage , speed is within the specified ranges, system dynamical equations are slightly modified so as to calculate the angle between the machine and infinite bus polar axis. Based on the actual difference in the speeds of the two machines to be synchronize, the phase angle difference between their voltages is calculated corresponding to the circuit breaker closing time . Ifthis anticipated phase angle difference is within a certaintolerance angle, a command is given to close the Circuit Breaker. The CB actually closes after a time called circuit breaker closing time (TC).

After the CB has closed, the line current is calculated and checked whether it is within the safe limit. In case it is dangerously large a command is given to open the CB. The circuit breaker opens after a time called CB opening time.

An attempt is again made to synchronize the generator.

5.7 DISCUSSION

It is found that CB is closing at the moment Speed is building up upto synchronous value. Since current transient is taking large time to settle down, voltage build up will take large time.

5.8 APPENDIX

1. Numerical Value used

d axis sync. reactance	XD	=	1.09 pu
d axis Magnetizing reactance	XMD	=	0.847 pu
q-axis synchronous reactance	XQ	=	0.66 pu
q-axis magnetizing reactionce	XMQ	=	0.507 pu
field reactance	XF	=	1.097 pu
d-axis damper ckt. reactance	XKD	=	1.007 pu
q-axis damper ckt reactance	XKQ	=	0.767 pu
Line reactance	XE	=	1.1224 pu
Field reactance	RF	=	0.000392 pu
d-axis damper ckt reactance	RKD	=	0.028 pu
q-axis damper ckt reactance	RKQ	=	0,034 pu
Inertia constant	H	=	6.6
Regulator gain	AK	=	25.0
Damper gain	AKF	=	0.04
Regulator's time constant	TA	=	0.15 sec.
Damper's time constant	TF	=	0.6 sec
Maximum voltage limit of Regulator	VRMX	=	1.0
Minimum voltage limit of Regulator	VRMN	=	-1.0
Permanent speed droop	SIGMA	=	0.04
Temporary speed droop	DLT	=	0.31
Governor Response time	TG	=	0.20 sec.
Dashpot time Constant	TR	=	5.0
Max. value of primemover torque	TMX	=	1.0 pu

Water starting time	TW = 1.0 sec
Voltage deviation (pu) from normal within which the syn. comes into action	VER = 0.05
Speed -do-	WER = 0.005
Incremental voltage deviation (pu) within which a transient response is assumed to have reached S.S.	EPV = 0.002
Speed -do-	EPW = 0.0002
Initial Machine's Terminal Voltage	VTI = 0.6 pu
Machine's initial speed	WI = 0.8 pu
Bus voltage	VB = 1.0 pu
Frequency of Bus	FBUS = 50.0
Frictional Torque	TFW = 0.2
Retardation Factor for voltage	ACCV = 1.0
Retardation factor for frequency	ACCW = 1.0
Incremental Time for an iteration	DT = 0.002
Maximum simulation time for sync. action	TLMX = 3.0 sec
Tolerance	TLRN = 5.0
Circuit breaker closure time	TC = 0.1 sec
Circuit breaker opening time	TCO = 0.1 sec
α is denoted by	ALPM = 20.0
Max.value of current beyond which CB will trip off	EIMX = 2.0
Results will be printed after every MP th iteration	MP = 25.0

2. Variables used in Programme attached

X(1)	Angular speed
X(2)	Slip speed

X(3)	Field flux linkage
X(4)	Flux linkage of the d axis damper ckt.
X(5)	Flux linkage of the q axis damper ckt
X(6)	Field terminal volate (referred to stator)
X(7)	Stabilizer output in the excitation chamber
X(8)	Gate Position
X(9)	Governor dash pot output
X(10)	Mechanical torque

F(1) to F(10) are time derivative of X(1) to X(10) respectively.

3. The programme is attached separately.

CHAPTER 6

CONCLUSION

In the thesis, the system's protection aspects and the excitation system's and governor's characteristic have not been considered.

In mathe matical analysis by computer, the excitation system and governor characteristic has been taken into account. As a first step, the problem formulation was done and a computer programme was written to develop more generalize system. It should be possible to develop the generalize system on microprocessors.

The auto sychronizer presented in the thesis has no provision for the protection of the system, in the event of any mal function. It should be possible to develop the generalized system wherein microprocessor would protect the system also.

R E F E R E N C E S

1. C.P.Adamson, and O.P. Mosland, "Automatic check synchronizing Equipment Using static relaying principles" Proc. IEE (GB), Vol. 108A, pp 331-39, 1961.

2. W.D. Humpage, B.D.Nellist and B.P.Sabberwal," Phase comparision methods in automatic synchronizing" Proc. IEE (GB) Vol. 112 pp 396-404, 1965.

3. A.D.Rajkumar, A.Kuppragulu and S.Hariharan,"Solid State automatic synchronizer" Journal of Inst.of Engineers(India) Electrical Engg.Division, Pt. EL-6, Vol.57, pp 268-273,June 77.

4. A.K.Ghai, H.K.Verma, P.Mukhopadhyay," A new synchronizer for small power houses", proc.of the Symposium on Computer Applications in large scale power systems,New Delhi,Aug.16-18,79

5. A.J.Nichols and Kenneth McKenzie," Build a compact microcomputer by starting with a Microprocessor like the 8080",Electronic Design, Vol.24, pp. 84-92, May 10,1976.

6. Edwin E.Klingman," Microprocessor System Design",Prentice Hall Inc.,Englewood Cliffs,New Jersey,1977.

7. John L.Hilburn,Paul M.Julich," Microcomputers/Microprocessor Hardware,Software,and Applications", Prentice-Hall of India Pvt.Ltd.,New Delhi,1979.

8. User's Manmual,Intel 8080, September 1975.

```
AUTOSYCHRONIZATION OF ALTERNATORS
DIMENSION X(10),Q(10),F(10),EX(2)
OPEN(UNIT=1,DEVICE='DSK',FILE='MAHESH.DAT')
READ(1,101)XD,XMD,XQ,XMQ,XF,XKD,XKQ,XE
PRINT 101,XD,XMD,XQ,XMQ,XF,XKD,XKQ,XE
READ(1,101)RF,RKD,RKQ,H
PRINT 101,RF ,RKD,RKQ,H
READ(1,101)AK,AKF,TA,TF,VRMX,VRMN
PRINT 101,AK,AKF,TA,TF,VRMX,VRMN
READ(1,101) SIGMA,DLT,TG,TR,TMX,TW
PRINT 101,SIGMA,DLT,TG,TR,TMX,TW
READ(1,101) VER,WER,EPV,EPW
PRINT 101, VER,WER,EPV,EPW
READ(1,101) VTI,WI,VB,FBUS,TFW,ACCV,ACCW,VREF
PRINT 101 ,VTI,WI,VB,FBUS,TFW,ACCV,ACCW,VREF
READ(1,101)DT,TLMX,TC,TCO,ALPM,TLRN,EIMX
PRINT 101,DT,TLMX,TC,TCO,ALPM,TLRN,EIMX
READ(1,102)MP
FORMAT(8F10.5)
FORMAT(I5)
EFD=VTI/WI
IF(EFD-VRMX)2,1,1
PRINT201
FORMAT(15HFLD.IS.ABNORMAL)
GO TO 100
IF(EFD-VRMN)1,1,3
T=0.0
MP=0
WBAS=314.15926
X(1)=0.0
X(2)=WI*WBAS
X(3)=XF*SQRT(2.0)*VTI/(XMD*X(2))
X(4)=VTI*SQRT(2.0)*VTI/(XMD*X(2))
X(5)=0.0
X(6)=EFD
X(7)=0.0
X(8)=0.0
X(9)=0.0
X(10)=TFW
EX(1)=1.0E+10
EX(2)=XE
WBUS=2.0*3.1415926*FBUS
WREF=WBUS
AM=H/(3.1415926*50.0)
VTN=VTI
VTO=VTI
EI=EIMX
WN=WREF
WO=WREF
=0

=1
JJ=1
GO TO 19
IF(ABS(VTN-VB)-VER*VB)70,71,71
KVR=1
VREF=VREF+ACCV*(VB-VTN)
IF(ABS(WN-WBUS)-WER*WBUS)72,73,73
WREF=WREF+ACCW*(WBUS-WN)
GO TO 75
IF(KVR)100,74,75
JJ=2
GO TO 19
IF(ABS(VTN-VTO)-EPV)76,76,75
IF(ABS(WN-WO)-EPW)77,77,75
KVR=0
GO TO 11
ALPA=TC*(WN-WBUS)
K=1
IF(ABS(ALPA)-ALPM*3.1415926/180.0)81,80,80
IF(ABS(X(1)+ALPA)-TLRN*3.1415926/180.0)82,80,80
TCLS=T+TC
WMC=WN/WBAS
PRINT 110,TCLS,VTN,WMC
FORMAT(5X,7HTCLOSE=,F10.5,3HVT=,F10.5,5HWM/C=,F10.5,2HT=,F10.5)
JJ=3
GO TO 19
```

```
 13     IF(T-TCLS)84,83,83
 83     JCB=2
 85     JJ=4
        GO TO 19
 14     IF(EI-EIMX)85,86,86
 86      TOPN=T+TCU
        JJ=5
        GO TO 19
 15      IF(T-TOPN)19,4,4
 19      VTO=VTN
        WO=WN
        XDT=XD+EX(JCB)
        XQT=XQ+EX(JCB)
        D=XMD*XMD/XDT
        A=(XF-D)/WBAS
        B=(XMD-D)/WBAS
        C=(XKD-D)/WBAS
        DEN=A*C-B*B
        J=1
        THTB=T*WBUS
        GO TO (60,62),K
 20      DO 21 I=K,10
        X(I)=X(I)+0.5*DT*F(I)
 21      Q(I)=DT*F(I)
        J=2
        GO TO (60,62),K
 22     DO 23 I=K,10
        X(I)=X(I)+0.29289*(DT*F(I)-Q(I))
 23      Q(I)=0.58579*DT*F(I)+0.12132*Q(I)
        J=3
        GO TO (60,62),K
 24     DO 25 I=K,10
        X(I)=X(I)+1.7071*(DT*F(I)-Q(I))
 25     Q(I)=Q(I)+1.7071*(2.0*DT*F(I)-3.0*Q(I))
        J=4
        GO TO(60,62),K
   26 DO 27 I=K,10
   27 X(I)=X(I)+0.16667*(DT*F(I)-2.0*Q(I))
      T=T+DT
      WN=X(2)
      IF(K-1)100,120,130
  130 QKI=X(5)*WBAS/XKQ
      FI=(X(3)*XKD-X(4)*XMD)*WBAS/(XF*XKD-XMD*XMD)
      DKI=(X(4)*XF-X(3)*XMD)*WBAS/(XF*XKD-XMD*XMD)
      DI=0.0
       QI=0.0
      GO TO 140
  120 VD=SQRT(2.0)*VB*SIN(X(1))
      VO=SQRT(2.0)*VB*COS(X(1))
      W=X(2)
      QKI=(X(5)+VD*XMQ/(XQT*W))/((XKQ-XMQ*XMQ/XQT)/WBAS)
      QI=(VD+XMQ*QKI*(W/WBAS))/(XQT*W/WBAS)
      FEN=XMD*VQ/(XDT*W)
      FI=((X(3)-FEN)*C-(X(4)-FEN)*B)/DEN
       DKI=(A*(X(4)-FEN)-B*(X(3)-FEN))/DEN
      DI=((XMD*(FI+DKI)*W/WBAS)-VQ)/(XDT*W/WBAS)
  140 SID=(-XD*DI+XMD*(FI+DKI))/WBAS
      SIQ=(-XQ*QI+XMQ*QKI)/WBAS
      TE=0.5*WBAS*(SID*QI-SIQ*DI)
      VTN=W*SQRT((SID*SID+SIQ*SIQ)/2.0)
      CI=SQRT((DI*DI+QI*QI)/2.0)
      ACCN=F(2)/WBAS
      IF(T-TLMX)103,103,100
  103  P=NP+1
      IF(NP-[illegible]P)105,104,104
  104 NP=0
      PRINT111,T,VTN,X(2),X(6),X(10),VREF,WREF
   PRINT 111,F(8),F(9),F(10),X(8),X(9),X(10)
  111 FORMAT(5X,5(E14.8,2X))
      IF(K-1)100,125,105
  125 PRINT111,X(1),EI,TE,ACCN
  105 GO TO (11,12,13,14,15),JJ
   60 F(1)=X(2)-WBUS
      VD=SQRT(2.0)*VB*SIN(X(1))
      VQ=SQRT(2.0)*VB*COS(X(1))
      W=X(2)
       QKI=(X(5)+VD*XMQ/(XQT*W))/((XKQ-XMQ*XMQ/XQT)/WBAS)
```

```
          QI=(VD+XMQ*QKI*(W/WBAS))/(XQT*W/WBAS)
          FEN=XMD*VQ/(XDT*W)
          FI=((X(3)-FEN)*C-(X(4)-FEN)*B)/DEN
          DKI=(A*(X(4)-FEN)-B*(X(3)-FEN))/DEN
          DI=((XMD*(FI+DKI)*W/WBAS)-VQ)/(XDT*W/WBAS)
          SID=(-XD*DI+XMD*(FI+DKI))/WBAS
          SIQ=(-XQ*QI+XMQ*QKI)/WBAS
          TE=0.5*WBAS*(SID*QI-SIQ*DI)
          F(2)=(X(10)-TE-TFW)/AM
          GO TO 64
       62 F(2)=(X(10)-TFW)/AM
          W=X(2)
          QKI=X(5)*WBAS/XKQ
          FI=(X(3)*XKD-X(4)*XMD)*WBAS/(XF*XKD-XMD*XMD)
          DKI=(X(4)*XF-X(3)*XMD)*WBAS/(XF*XKD-XMD*XMD)
          DI=0.0
          QI=0.0
          SID=XMD*(FI+DKI)/WBAS
          SIQ=XMQ*QKI/WBAS
  64      F(3)=-RF*FI+X(6)*SQRT(2.0)*RF/XMD
          F(4)=-RKD*DKI
           F(5)=-RKQ*QKI
          VT=W*SQRT((SID*SID+SIQ*SIQ)/2.0)
          F(6)=(AK*(VREF-VT-X(7))-X(6))/TA
          VF=F(6)
          IF(X(6)-VRMX)35,30,30
  30      X(6)=VRMX
          VF=0.0
          GO TO 37
  35      IF(VRMN-X(6))37,40,40
  40      X(6)=VRMN
          VF=0.0
  37      F(7)=(AKF*VF-X(7))/TF
          F(8)=(((WREF-W)/WBAS)-SIGMA*X(8)-X(9))/TG
          F(9)=(-X(9)+DLT*TR*F(8))/TR
          CF=F(8)
          Z=TFW+X(8)
          IF(Z-TMX)51,50,50
  50      Z=TMX
          CF=0.0
          GO TO 53
  51      IF(Z)52,52,53
  52      Z=0.0
          CF=0.0
  53        F(10)=2.0*(Z-TW*CF-X(10))/TW
          GO TO (20,22,24,26),J
          CLOSE(UNIT=1)
 100      STOP
          END
```

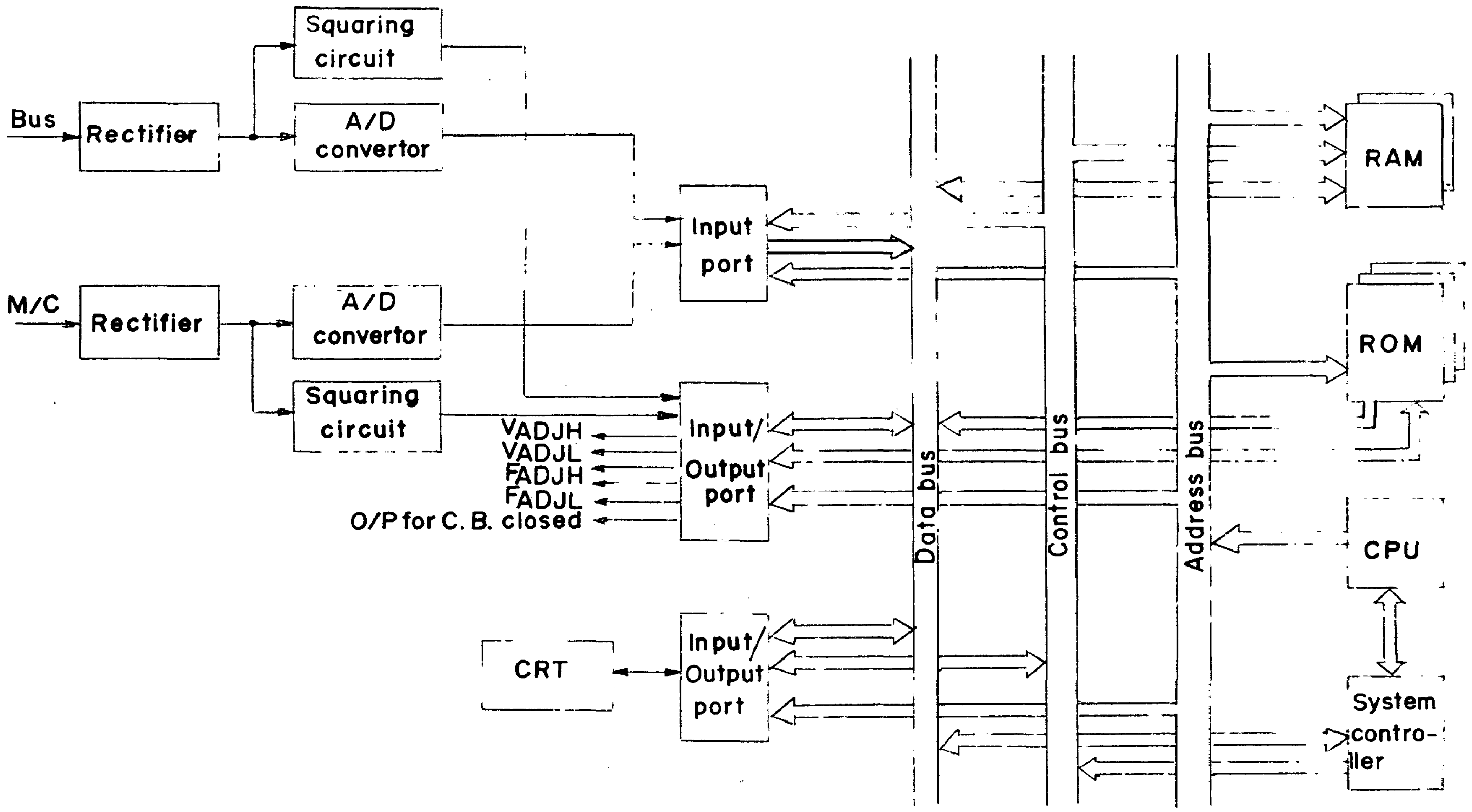


FIG.2.2 BLOCK DIAGRAM.

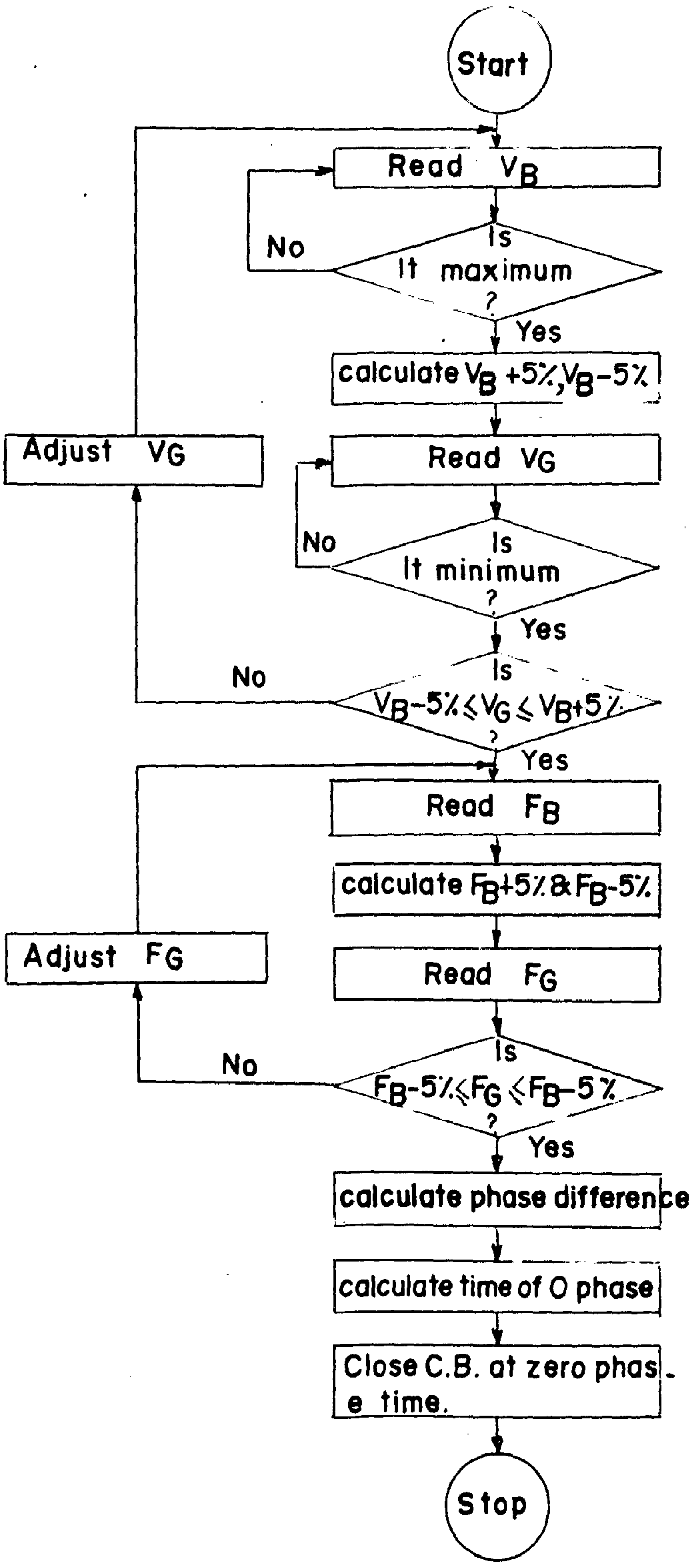

Start
Read V_B
Is It maximum ?
No
Yes
calculate V_B +5%, V_B -5%
Read V_G
Is It minimum ?
No
Yes
Adjust V_G
Is $V_B-5\% \leq V_G \leq V_B+5\%$?
No
Yes
Read F_B
calculate F_B+5% & F_B-5%
Read F_G
Adjust F_G
Is $F_B-5\% \leq F_G \leq F_B-5\%$?
No
Yes
calculate phase difference
calculate time of 0 phase
Close C.B. at zero phase time.
Stop

FIG.2.3.1 BASIC FLOW CHART.

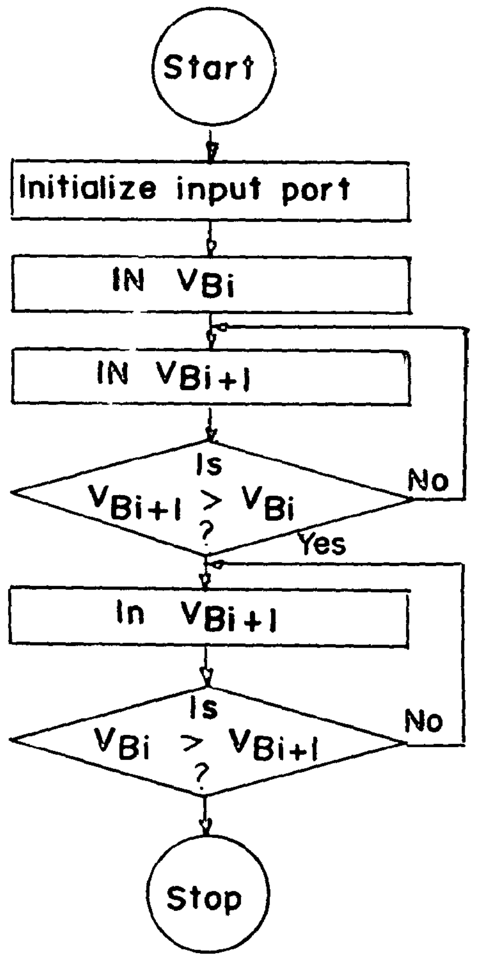


FIG.2.3.2 FLOW CHART FOR VOLTAGE MEASUREMENT.

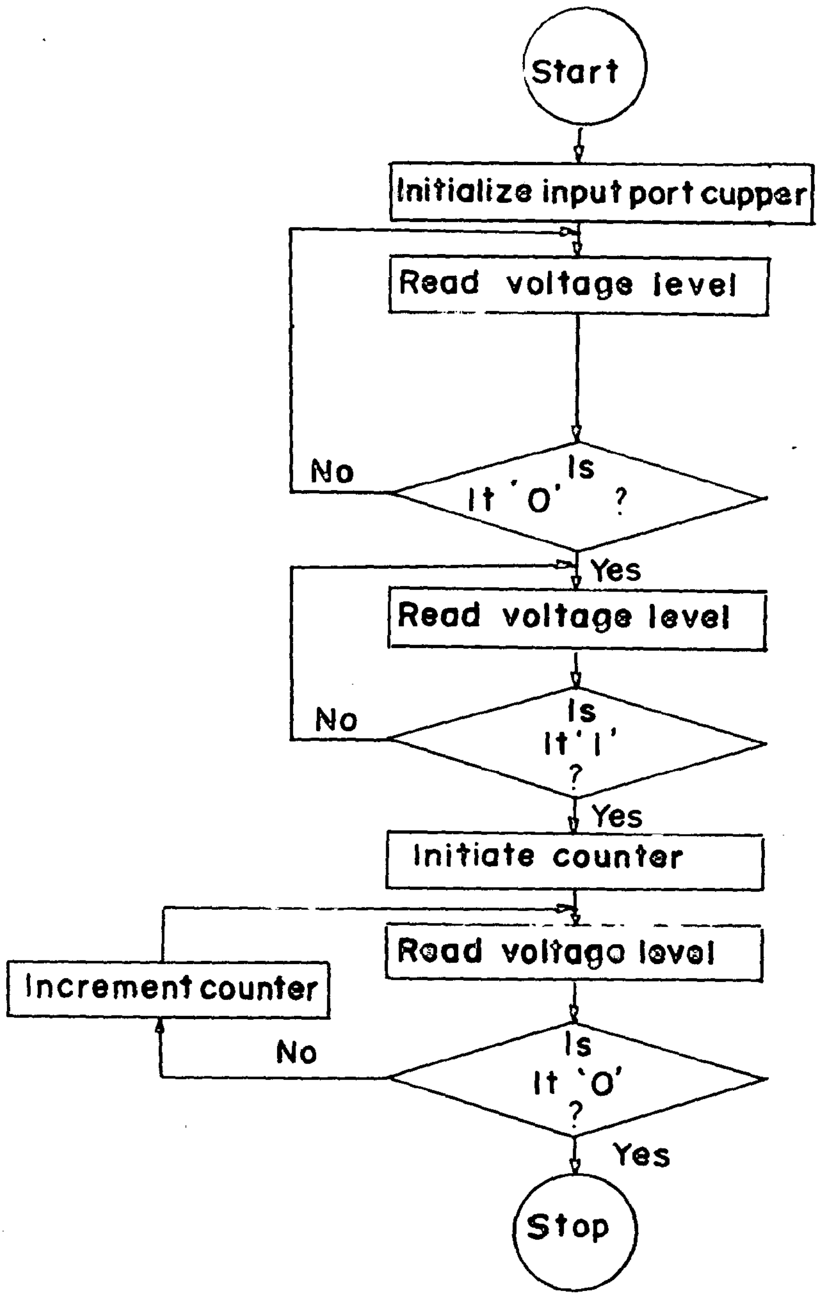


FIG.2.3.3 FLOW CHART FOR FREQUENCY MEASUREMENT.

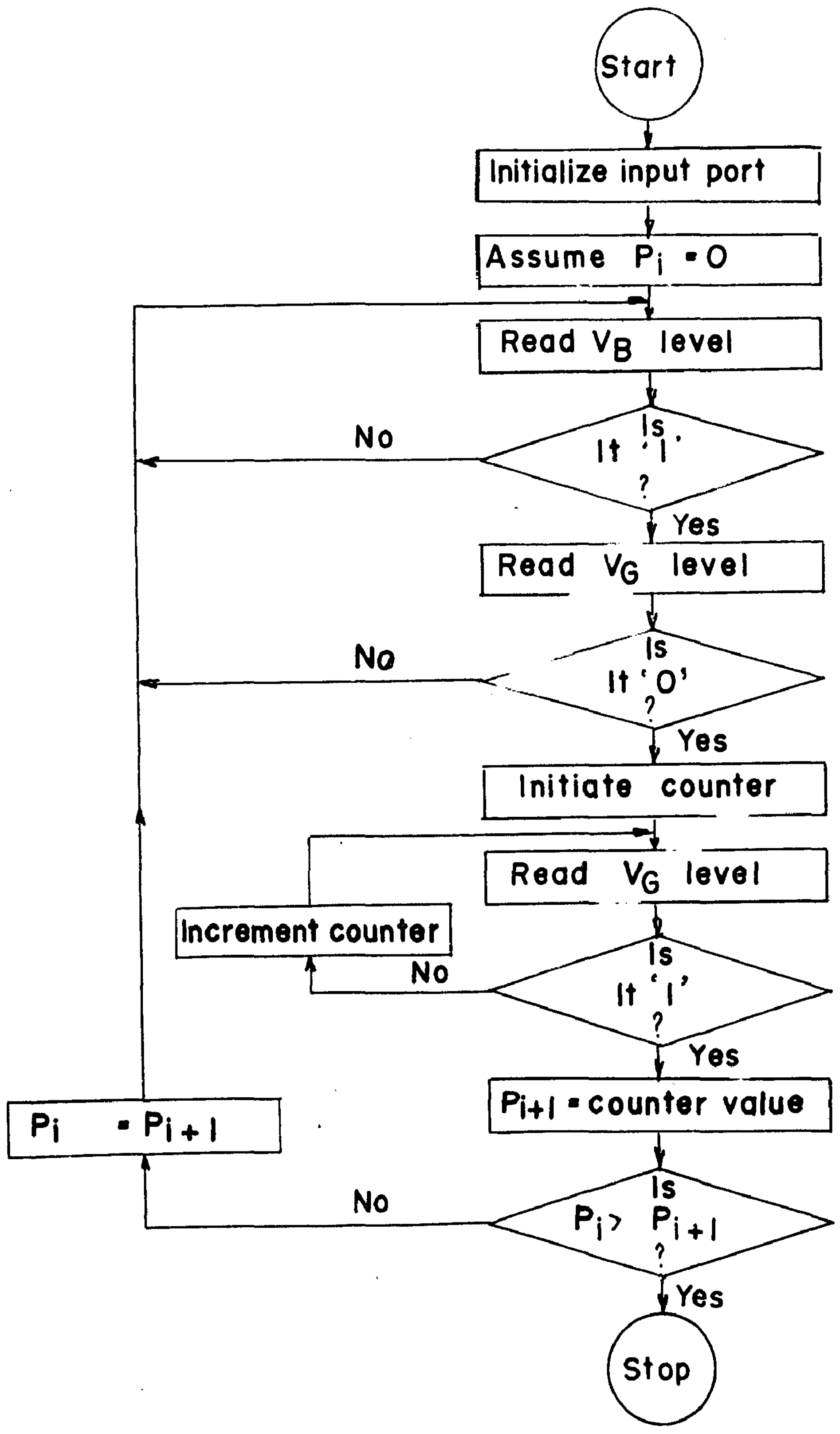


FIG.2.3.4 FLOW CHART FOR PHASE MEASUREMENT.

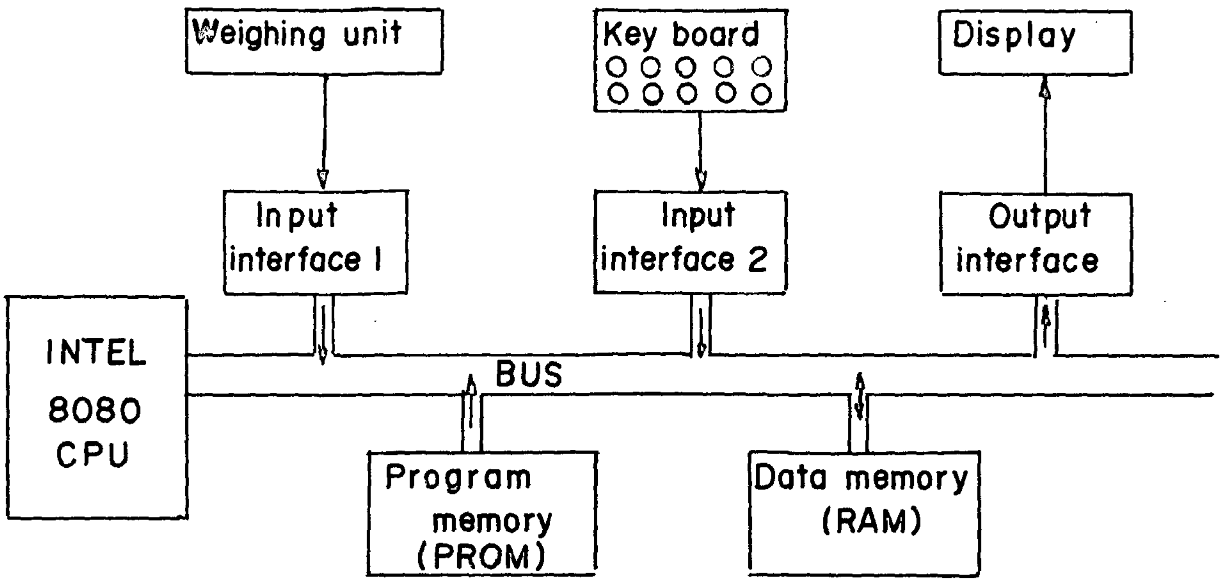


FIG.3.1 SYSTEM BLOCK DIAGRAM.

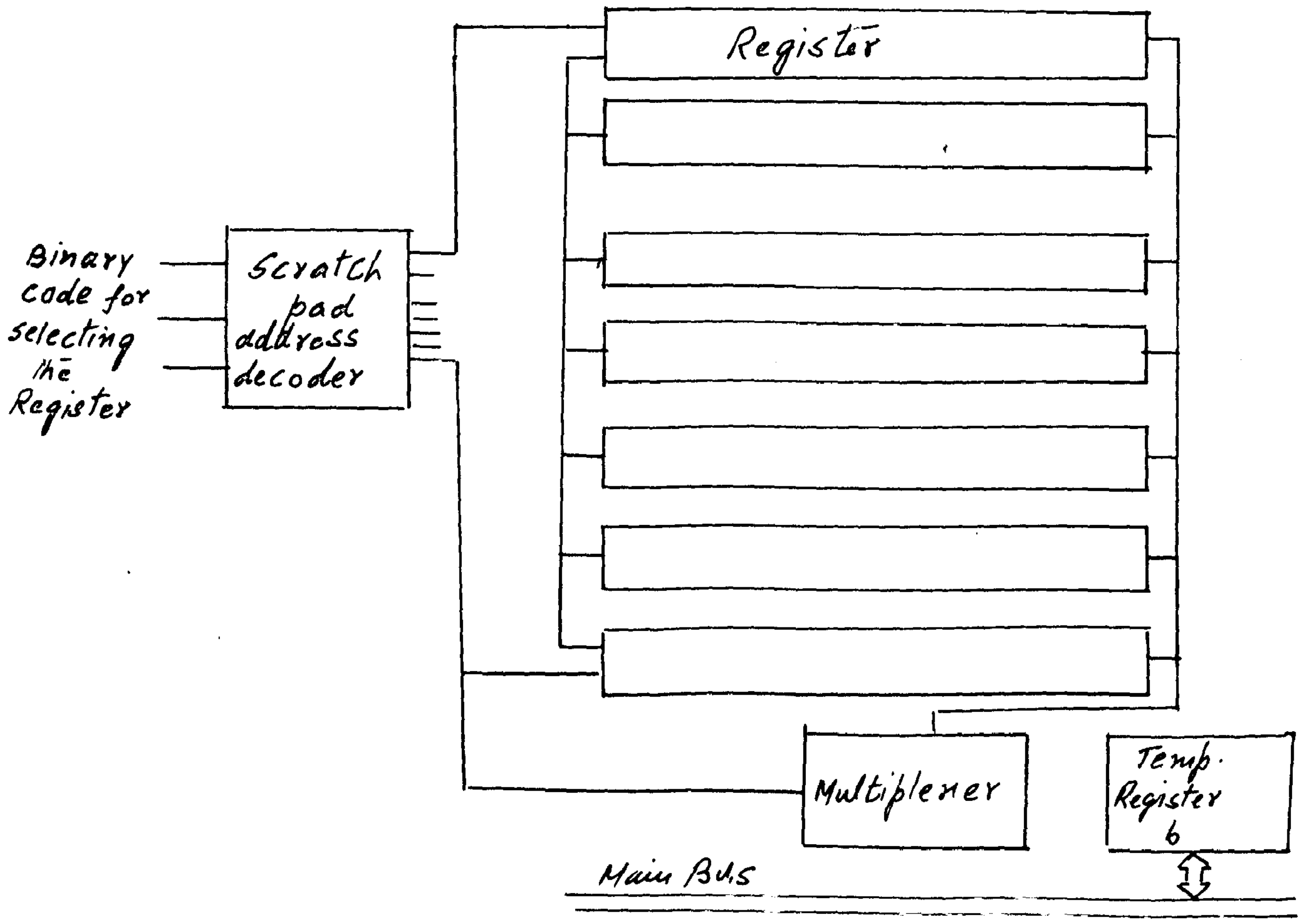

Register
Binary code for selecting the Register
Scratch pad address decoder
Multiplexer
Temp. Register 6
Main Bus

FIG.3.2

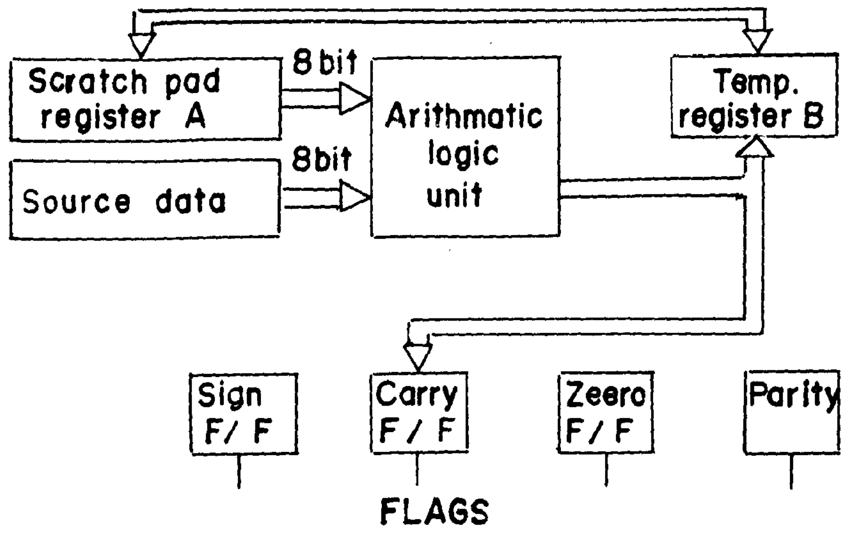


FIG.3.3

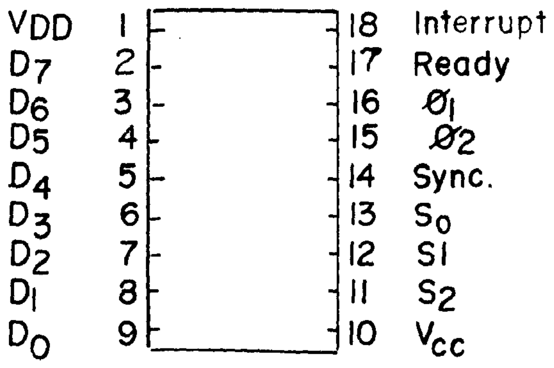


FIG.3.4 8080 CPU

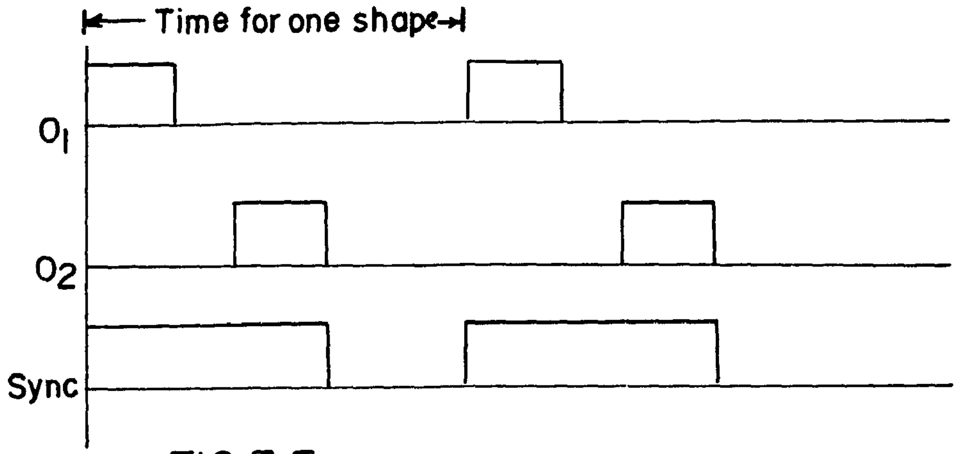


FIG.3.5

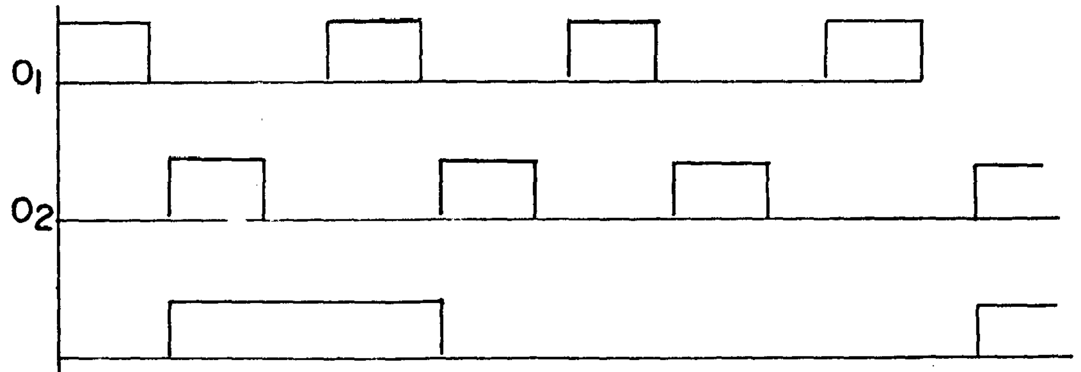


FIG. 3.6

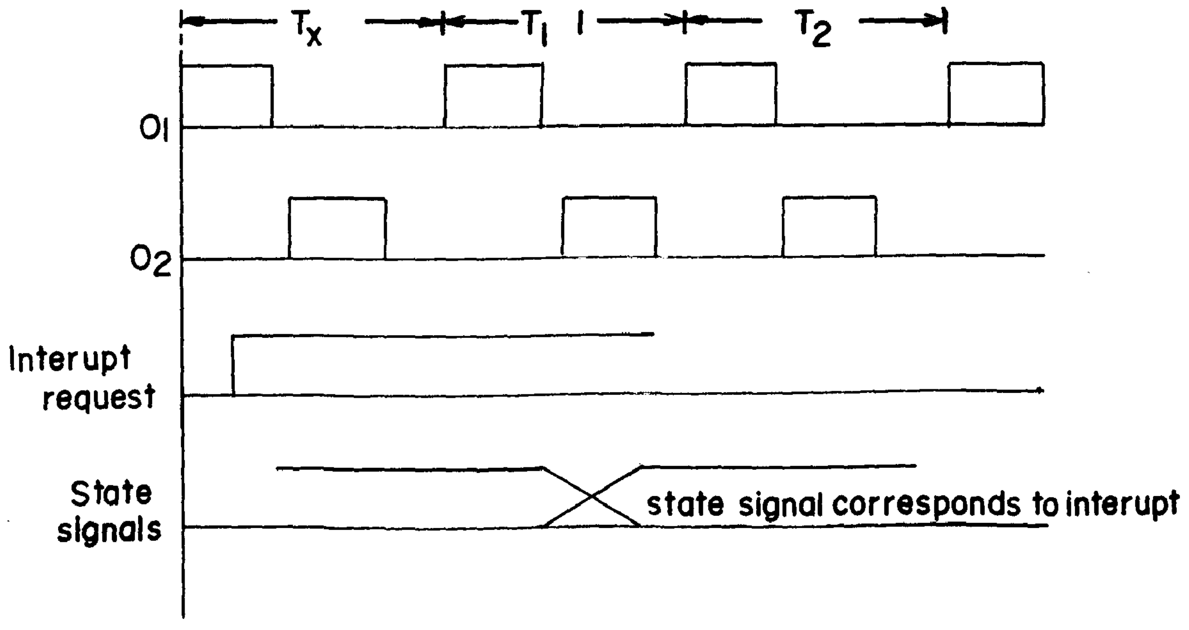


FIG. 3.7

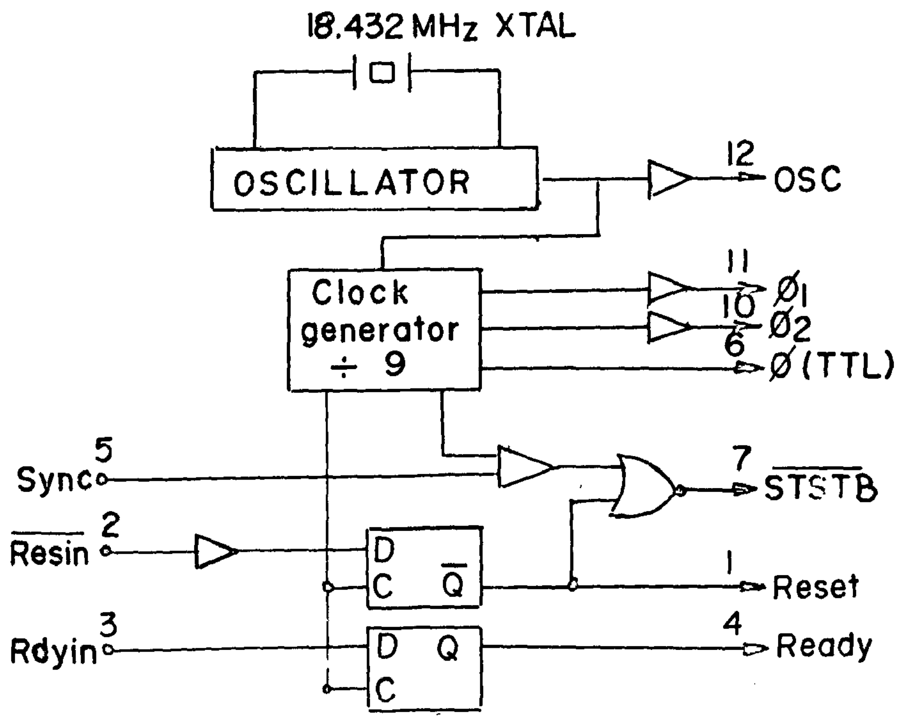


FIG.4.1.2

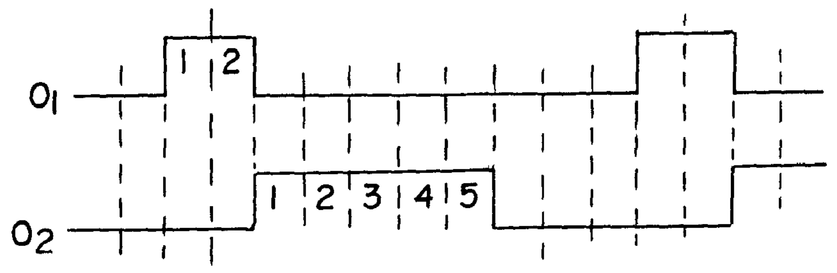


FIG.4.1.3

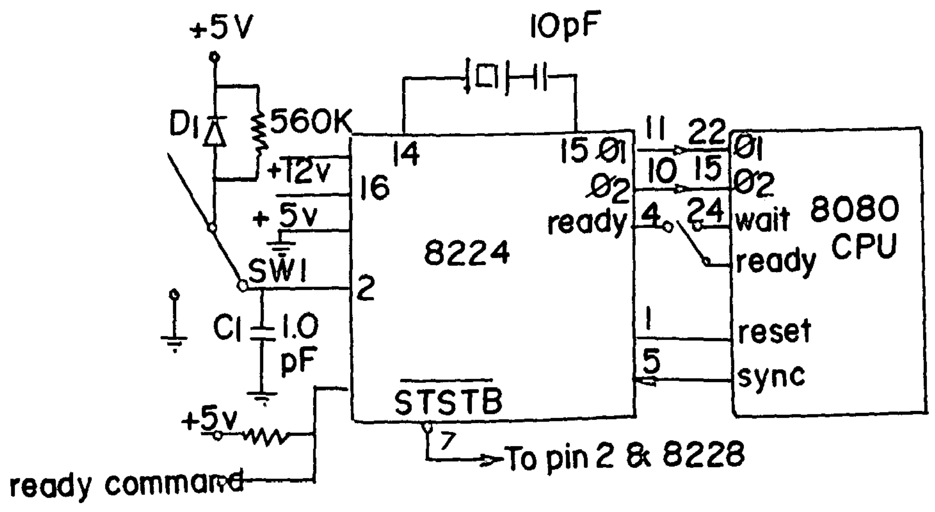


FIG.4.1.4

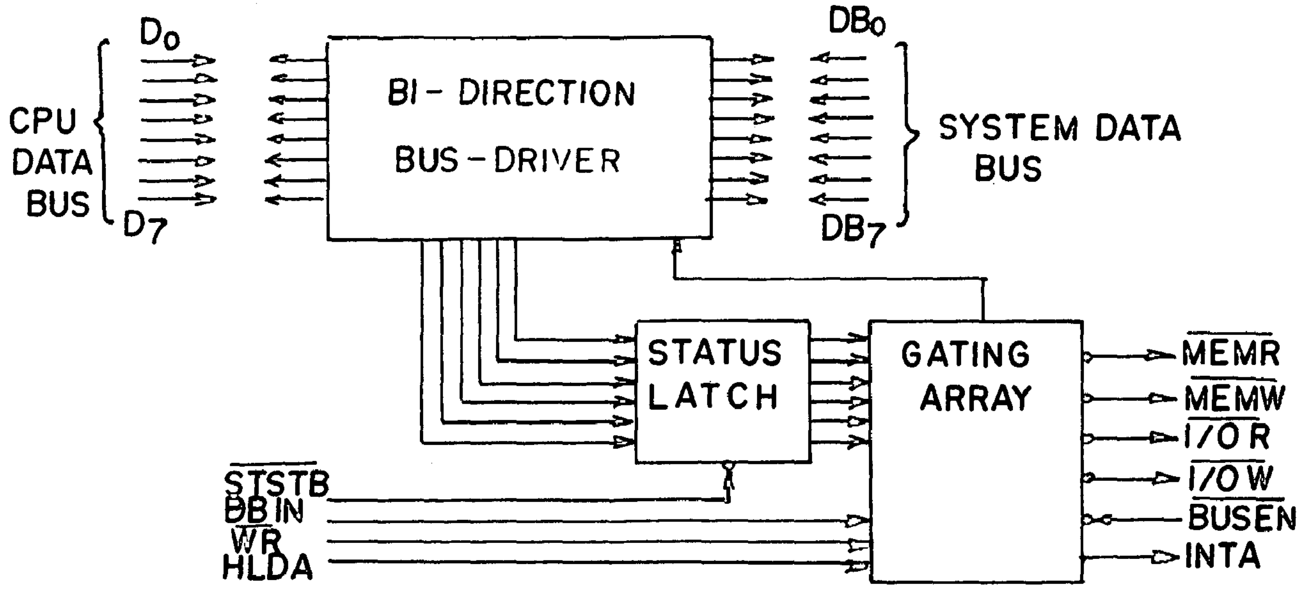


FIG.4.1.5

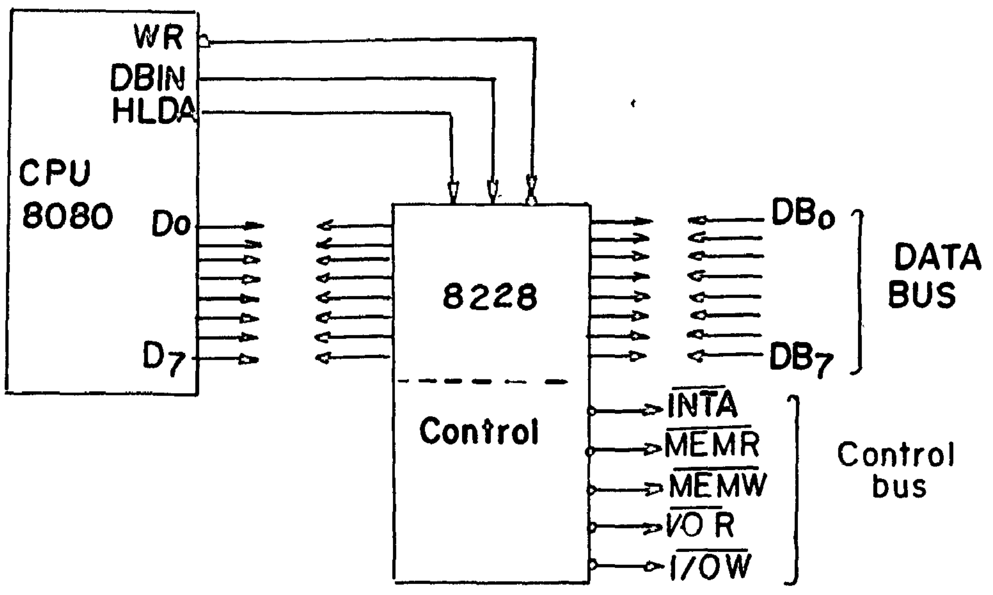


FIG.4.1.6

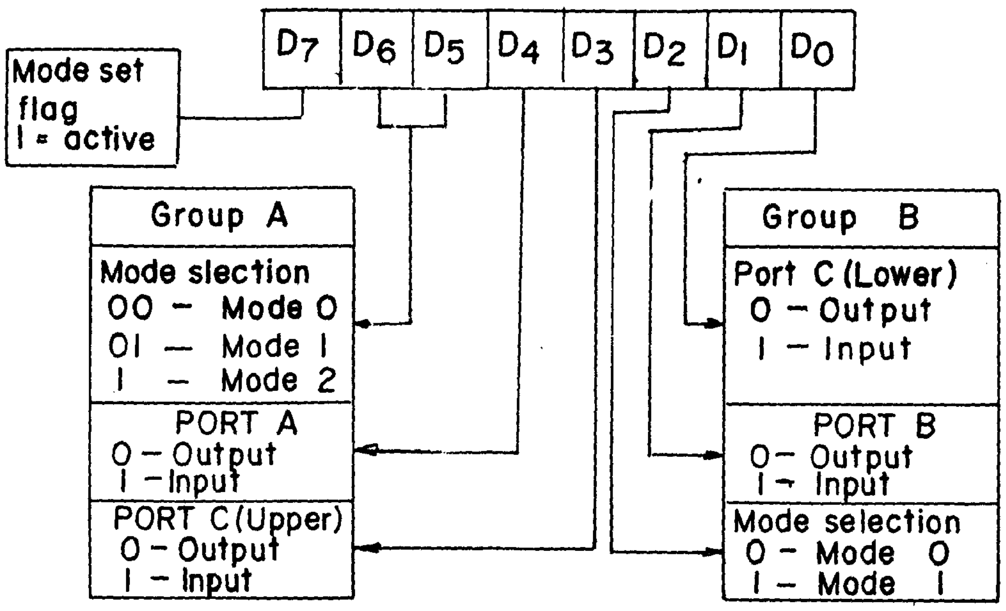

D7
D6
D5
D4
D3
D2
D1
D0
Mode set flag 1 = active
Group A
Mode slection
00 – Mode 0
01 – Mode 1
1 – Mode 2
PORT A
0 – Output
1 – Input
PORT C (Upper)
0 – Output
1 – Input
Group B
Port C (Lower)
0 – Output
1 – Input
PORT B
0 – Output
1 – Input
Mode selection
0 – Mode 0
1 – Mode 1

FIG. 4.1.8

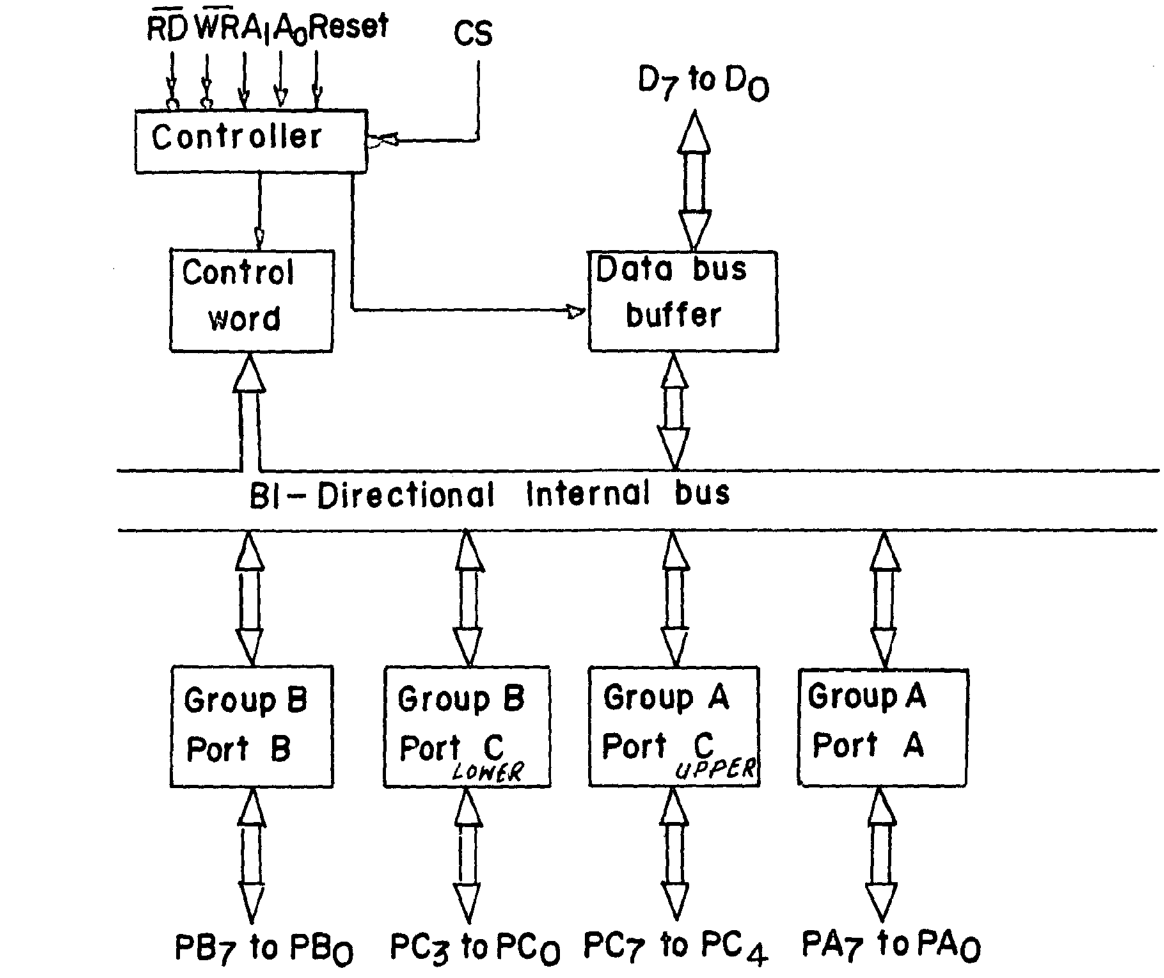


FIG. 4.1.7

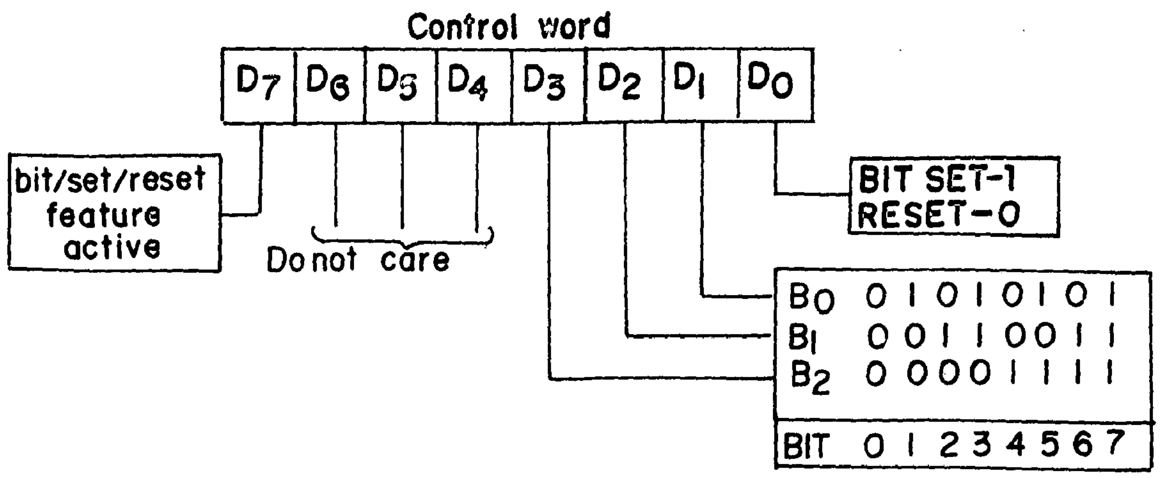


FIG. 4.1.9

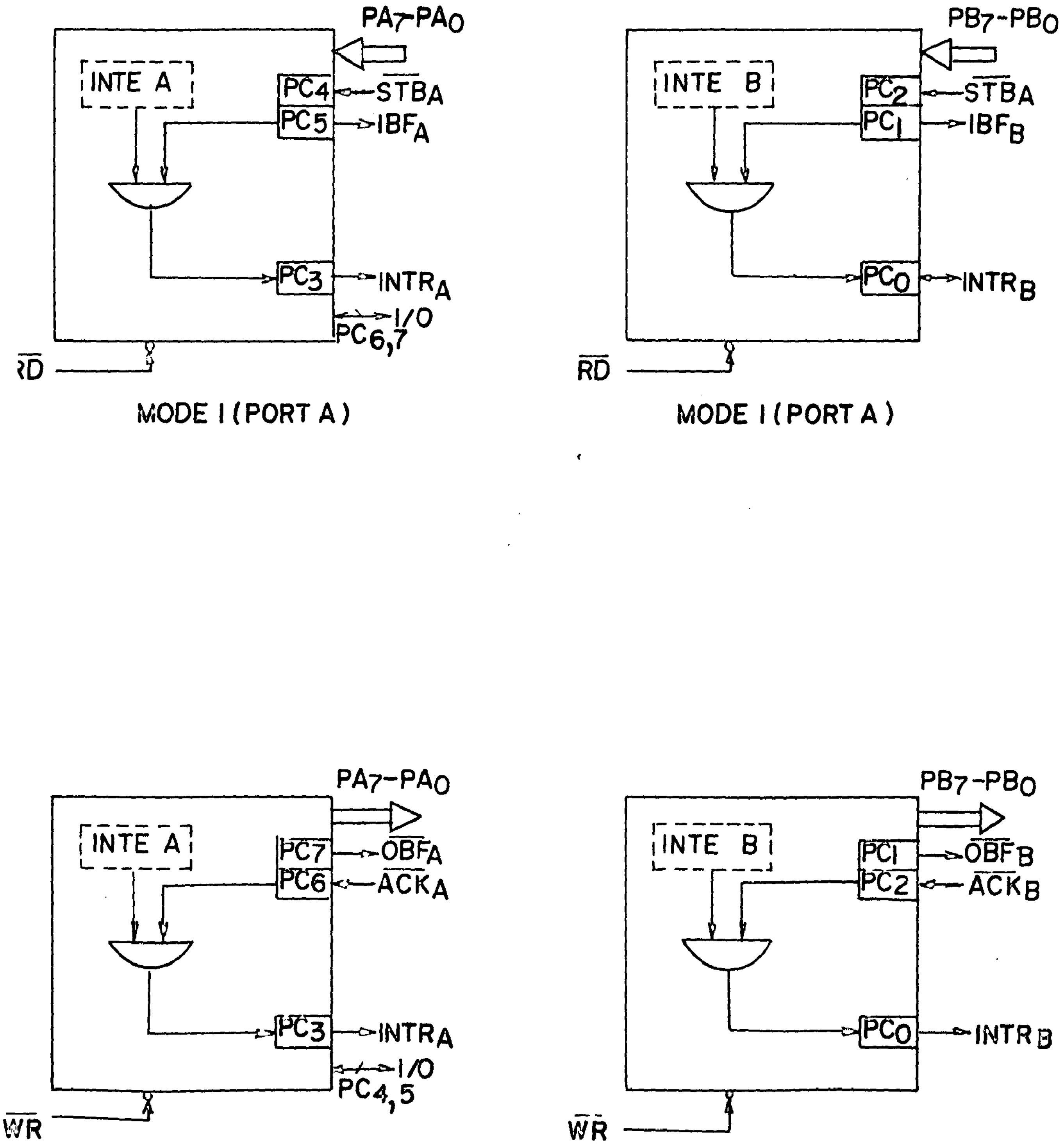

PA7-PA0
INTE A
PC4
$\overline{STB}_A$
PC5
IBF_A
PC3
$INTR_A$
I/O
PC6,7
$\overline{RD}$
MODE I (PORT A)
PB7-PB0
INTE B
PC2
$\overline{STB}_A$
PC1
IBF_B
PC0
$INTR_B$
$\overline{RD}$
MODE I (PORT A)
PA7-PA0
INTE A
PC7
$\overline{OBF}_A$
PC6
$\overline{ACK}_A$
PC3
$INTR_A$
I/O
PC4,5
$\overline{WR}$
PB7-PB0
INTE B
PC1
$\overline{OBF}_B$
PC2
$\overline{ACK}_B$
PC0
$INTR_B$
$\overline{WR}$

FIG. 4.1.10

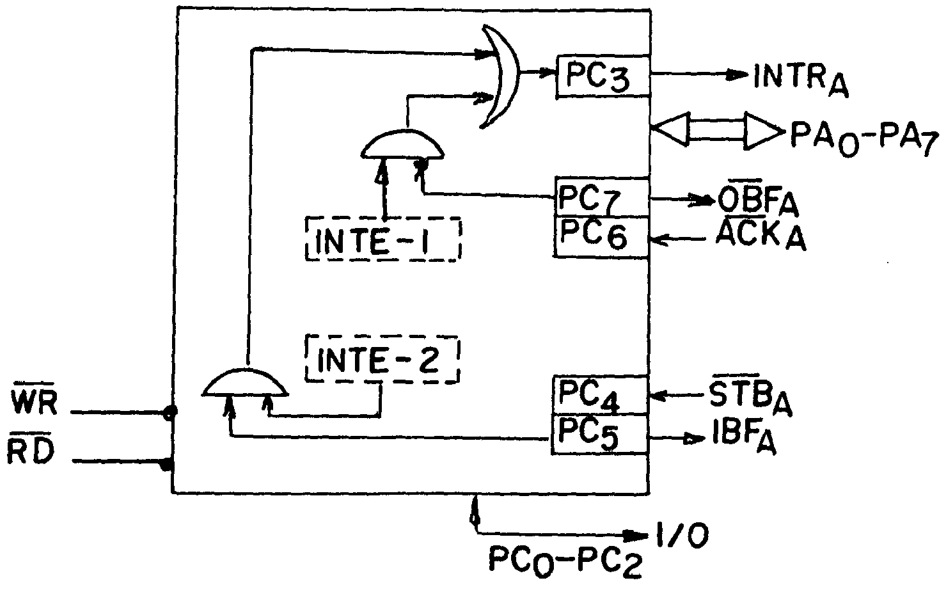


FIG.4.1.11 MODE 2

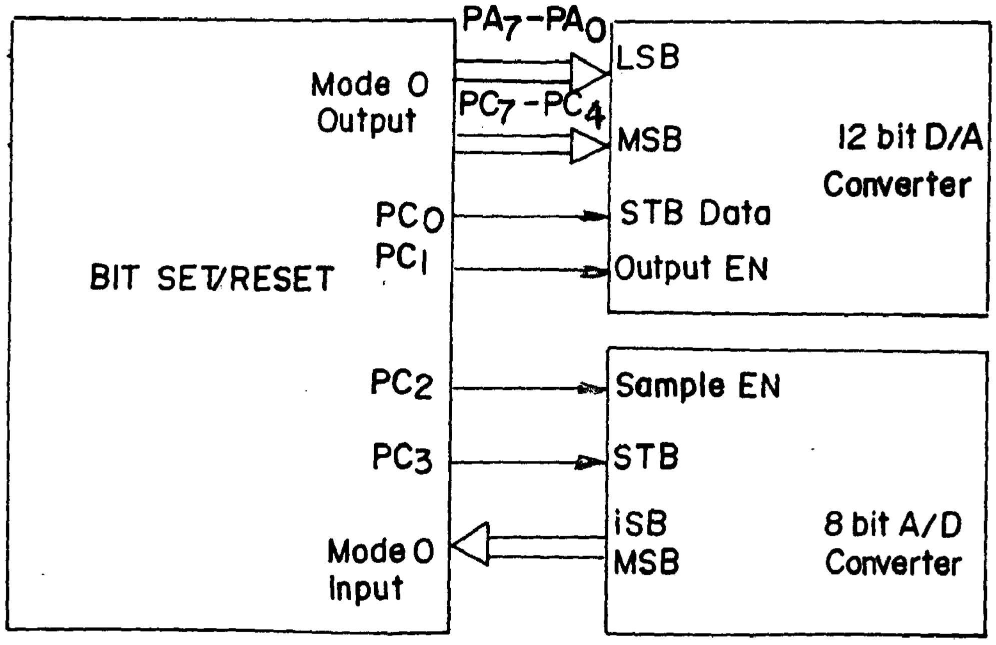


FIG.4.1.12

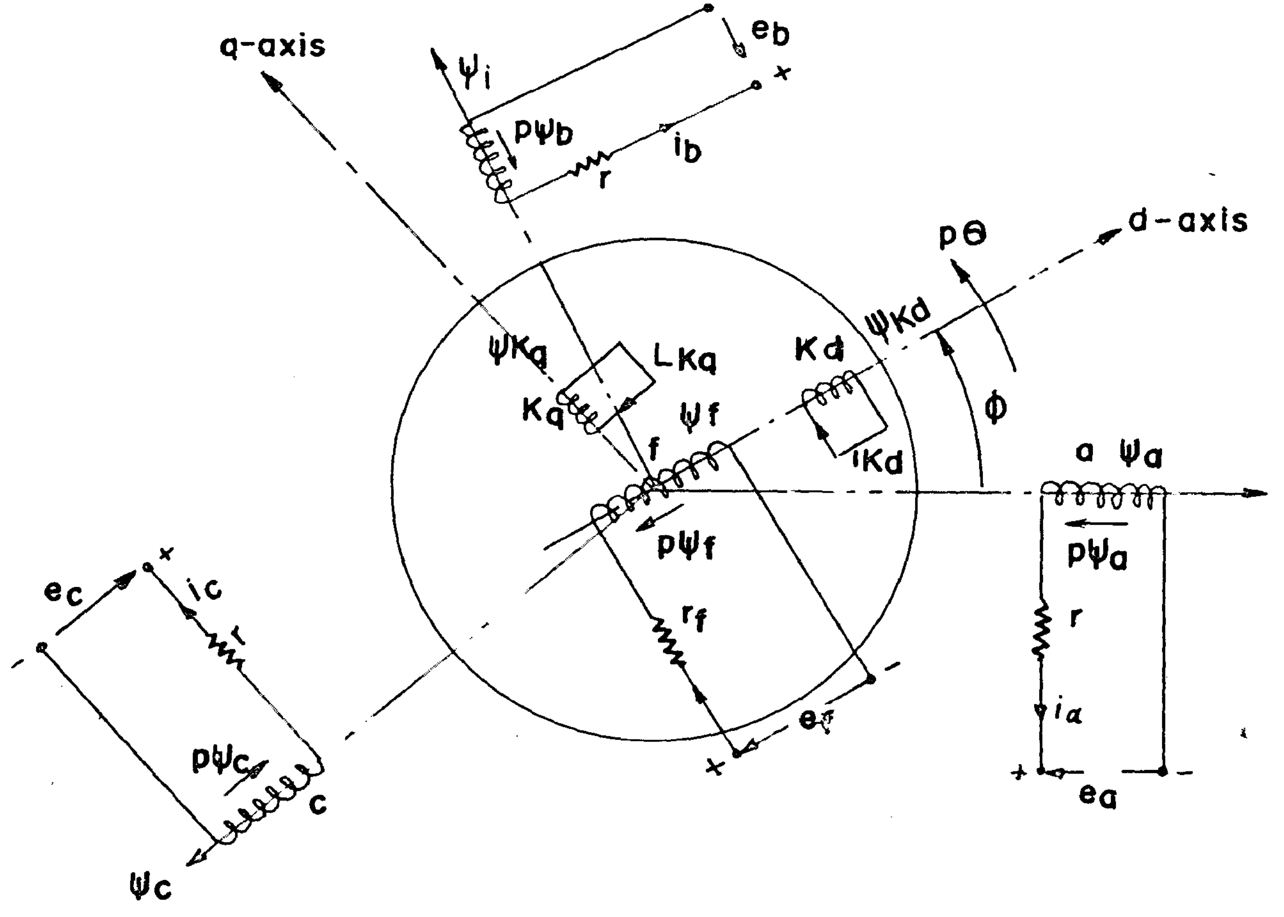


FIG.5.2.1 SCHEMATIC REPRESENTATION OF SYNCHRONOUS MACHINE.

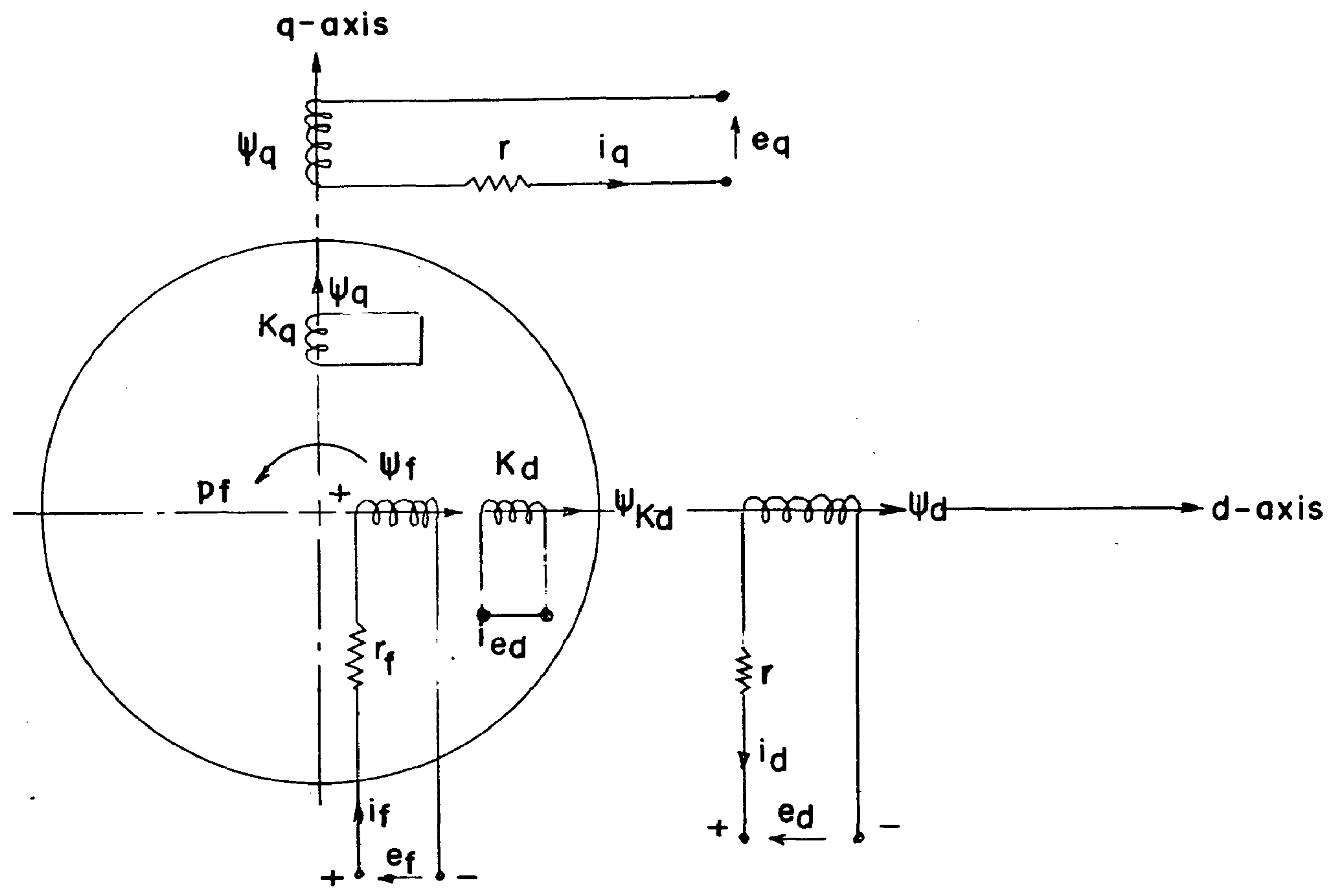


FIG.5.2.2AXIS MODEL OF SALIENT POLE SYNCHRONOUS MACHINE.

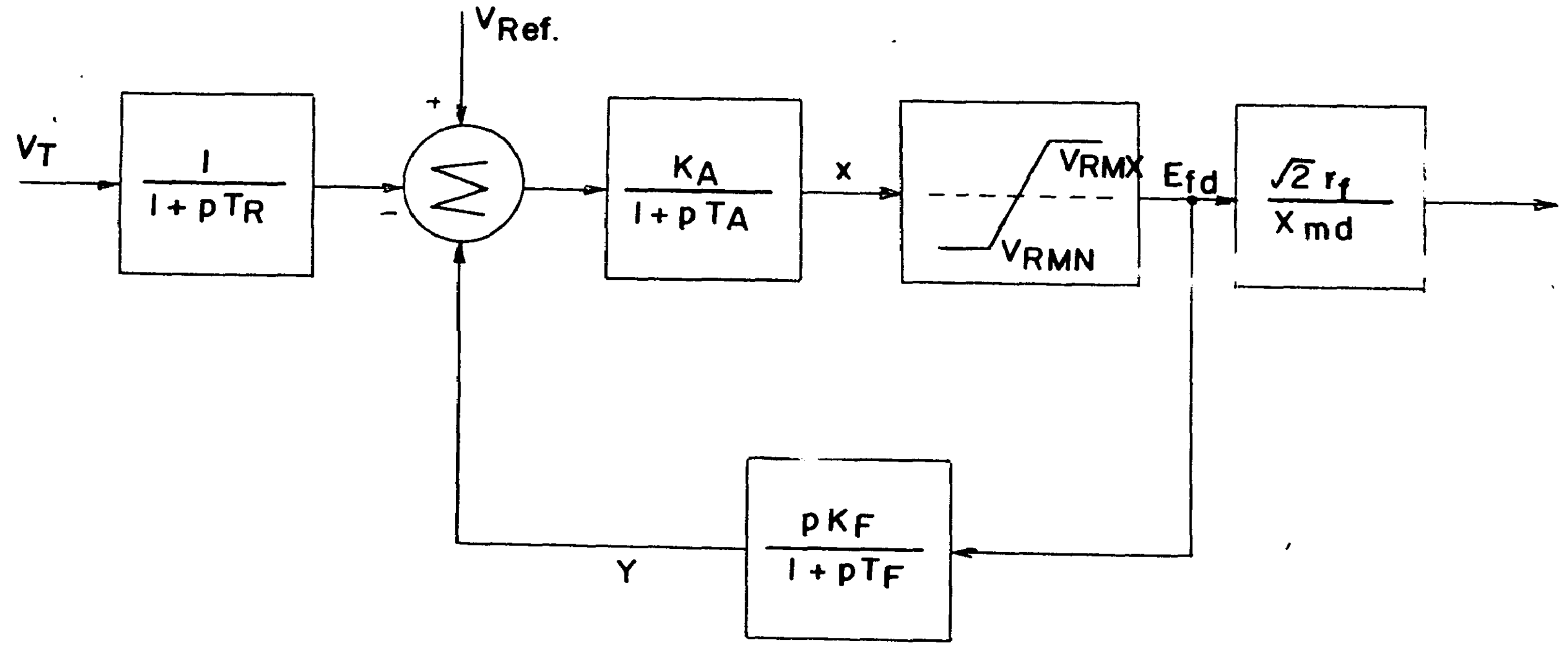
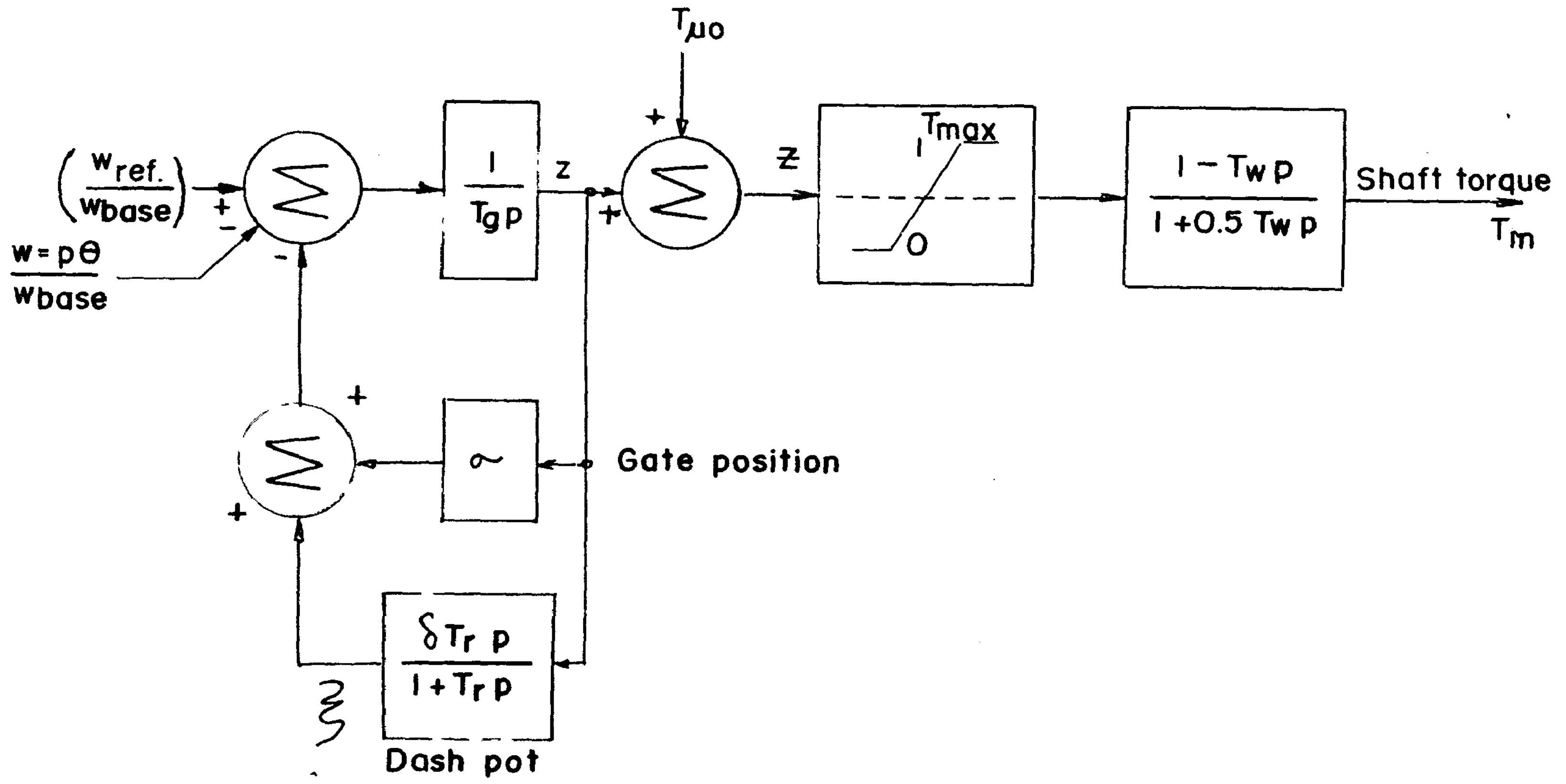
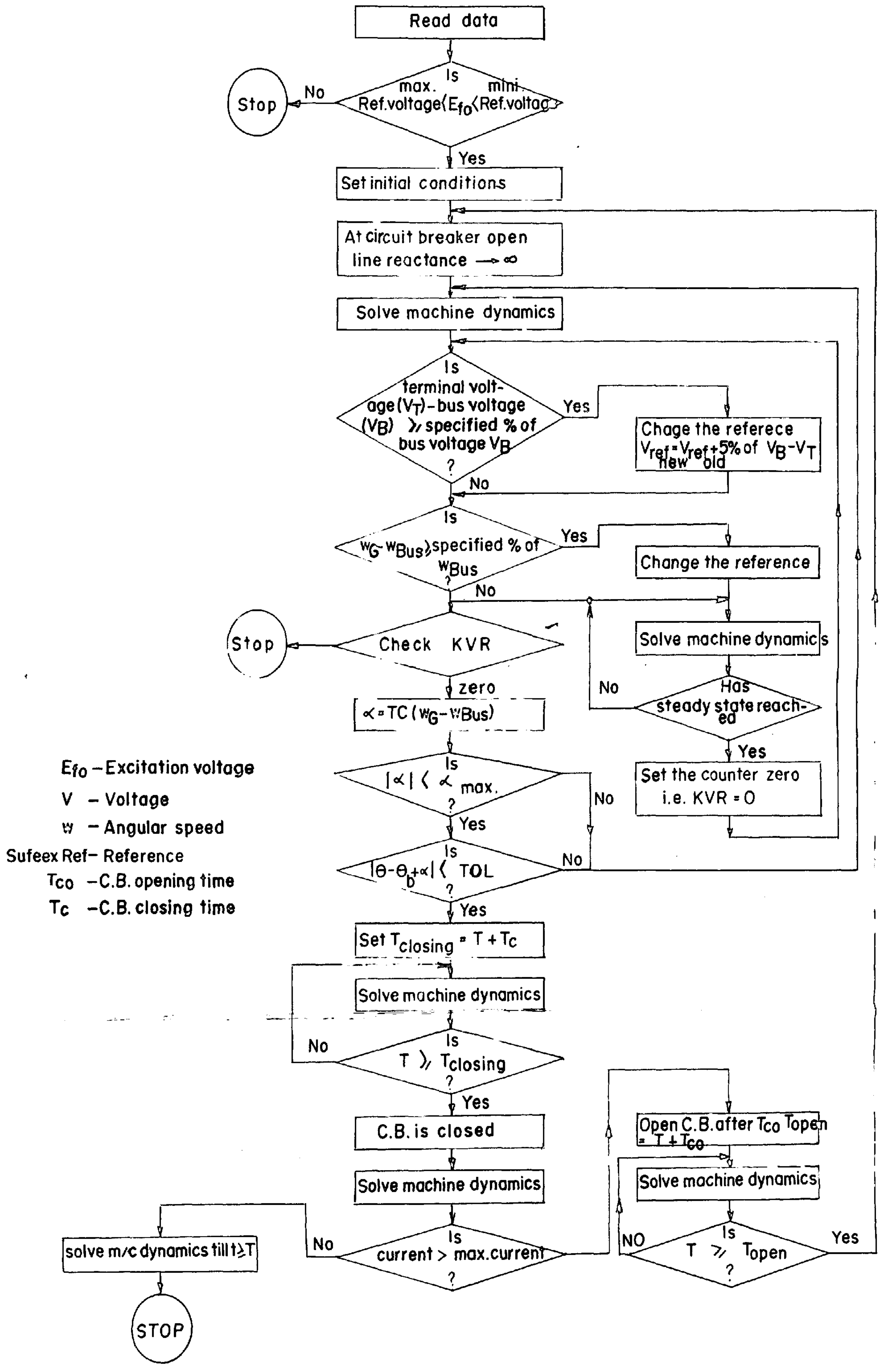


FIG.5.3.1 COMPUTER REPERSENTATION OF EXCITATION SYSTEM.



$\left(\frac{w_{ref.}}{w_{base}}\right)$
+
−
$\frac{w = p\theta}{w_{base}}$
−
$\frac{1}{T_g p}$
z
$T_{\mu o}$
+
+
ż
T_{max}
1
0
$\frac{1 - T_w p}{1 + 0.5 T_w p}$
Shaft torque
T_m
+
+
σ
Gate position
$\frac{\delta T_r p}{1 + T_r p}$
ξ
Dash pot

FIG.5.4.1 COMPUTER REPERSENTATION OF GOVERNOR SCHEME.



Read data
Is max. Ref.voltage < E_{fo} < mini. Ref.voltage ?
No
Stop
Yes
Set initial conditions
At circuit breaker open line reactance → ∞
Solve machine dynamics
Is terminal voltage (V_T) - bus voltage (V_B) ≥ specified % of bus voltage V_B ?
Yes
Chage the referece $V_{ref_{new}} = V_{ref_{old}} + 5\%$ of $V_B - V_T$
No
Is ($w_G - w_{Bus}$) > specified % of w_{Bus} ?
Yes
Change the reference
No
Solve machine dynamics
Has steady state reached ?
No
Yes
Set the counter zero i.e. KVR = 0
Check KVR
Stop
zero
α = TC ($w_G - w_{Bus}$)
Is $|\alpha| < \alpha_{max}$?
No
Yes
Is $|\theta - \theta_b + \alpha| <$ TOL ?
No
Yes
Set $T_{closing} = T + T_c$
Solve machine dynamics
Is T ≥ $T_{closing}$?
No
Yes
C.B. is closed
Solve machine dynamics
Is current > max.current ?
No
solve m/c dynamics till t≥T
STOP
Open C.B. after T_{co} $T_{open} = T + T_{co}$
Solve machine dynamics
Is T ≥ T_{open} ?
NO
Yes
E_{fo} – Excitation voltage
V – Voltage
w – Angular speed
Sufeex Ref– Reference
T_{co} – C.B. opening time
T_c – C.B. closing time

FIG.5.6.1 FLOW CHART.

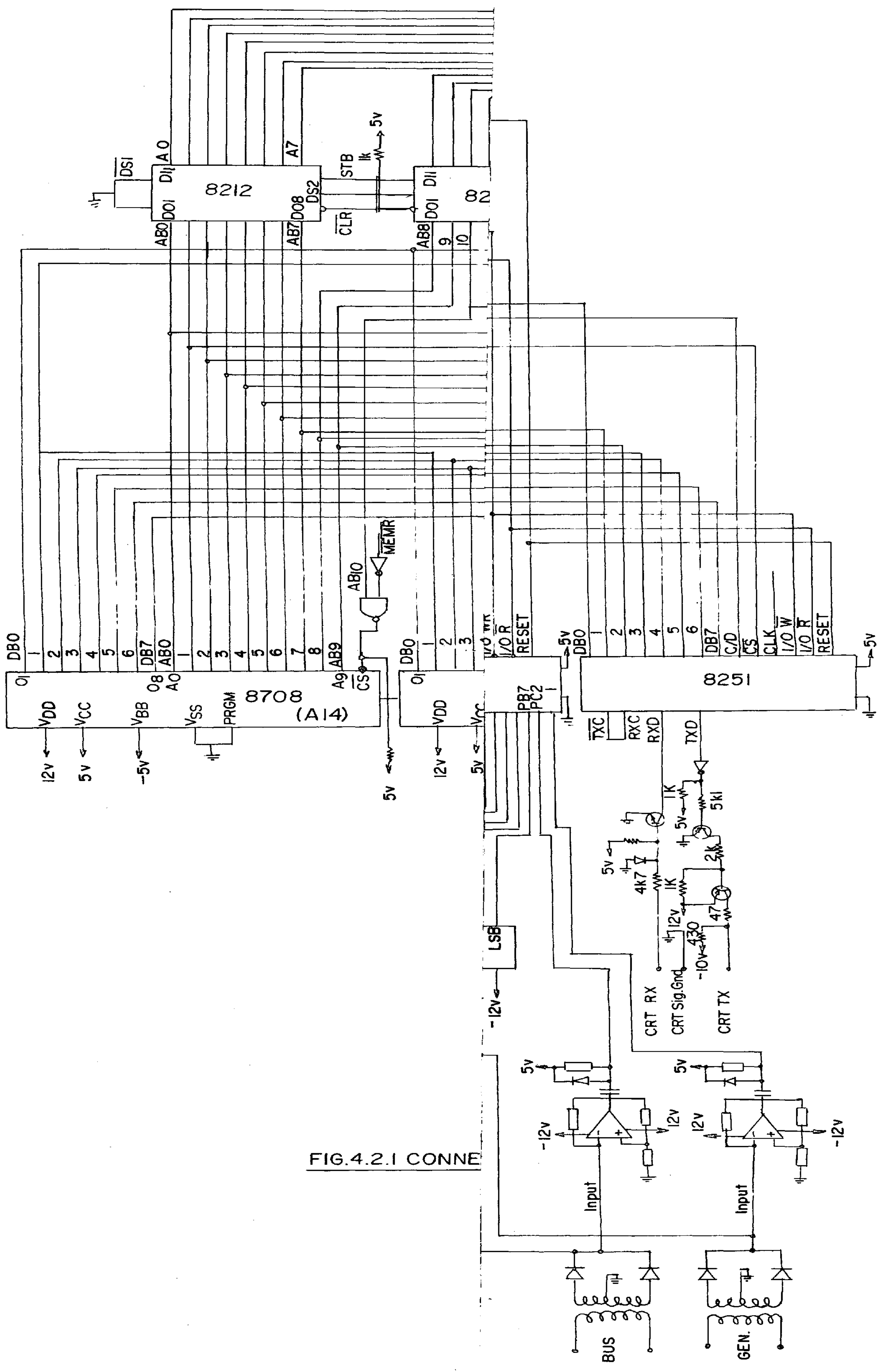


FIG.4.2.1 CONNE