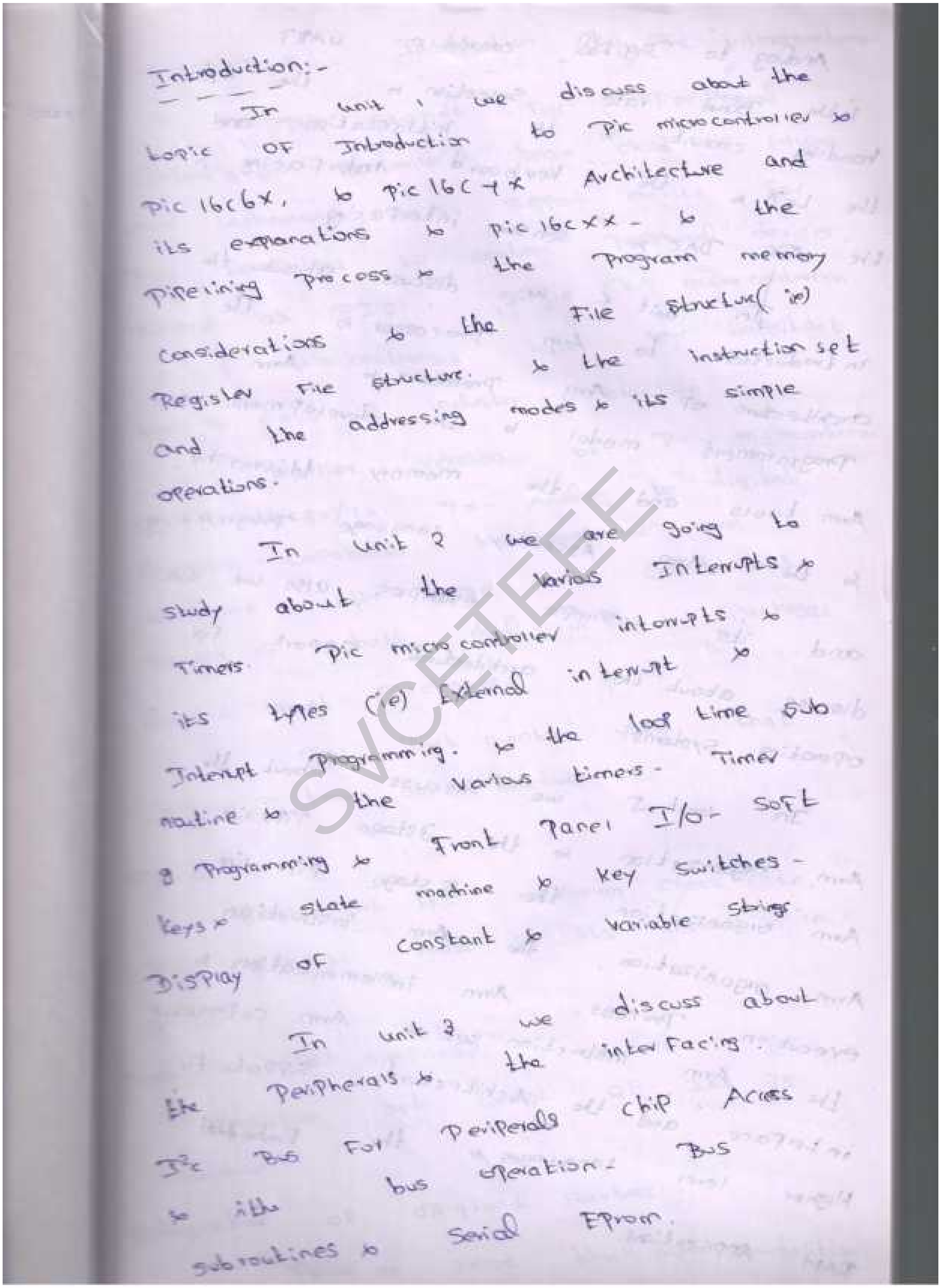
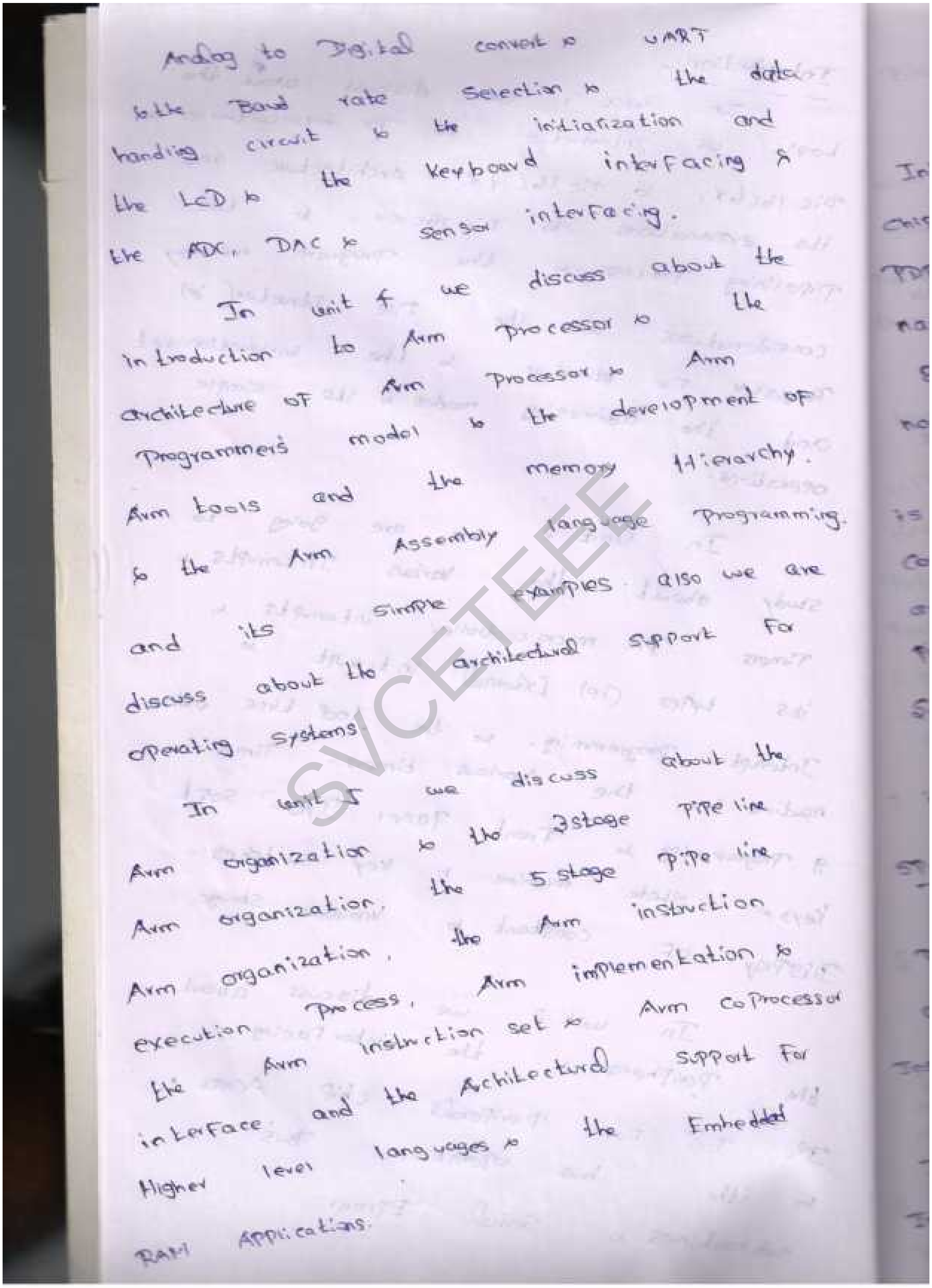


Introduction:-

In unit 1 we discuss about the topic of Introduction to pic microcontroller & pic 16c6x, & pic 16c7x Architecture and its explanations & pic 16cxx - & the pipelining process & the program memory considerations & the File structure (ie) Register File structure. & the instruction set and the addressing modes & its simple operations.

In unit 2 we are going to study about the various Interrupts & Timers pic micro controller interrupts & its types (ie) External interrupt & Interrupt programming. & the loop time sub routine & the various timers - Timer 2 programming & Front panel I/O - soft keys & state machine & key switches - Display of constant & variable strings

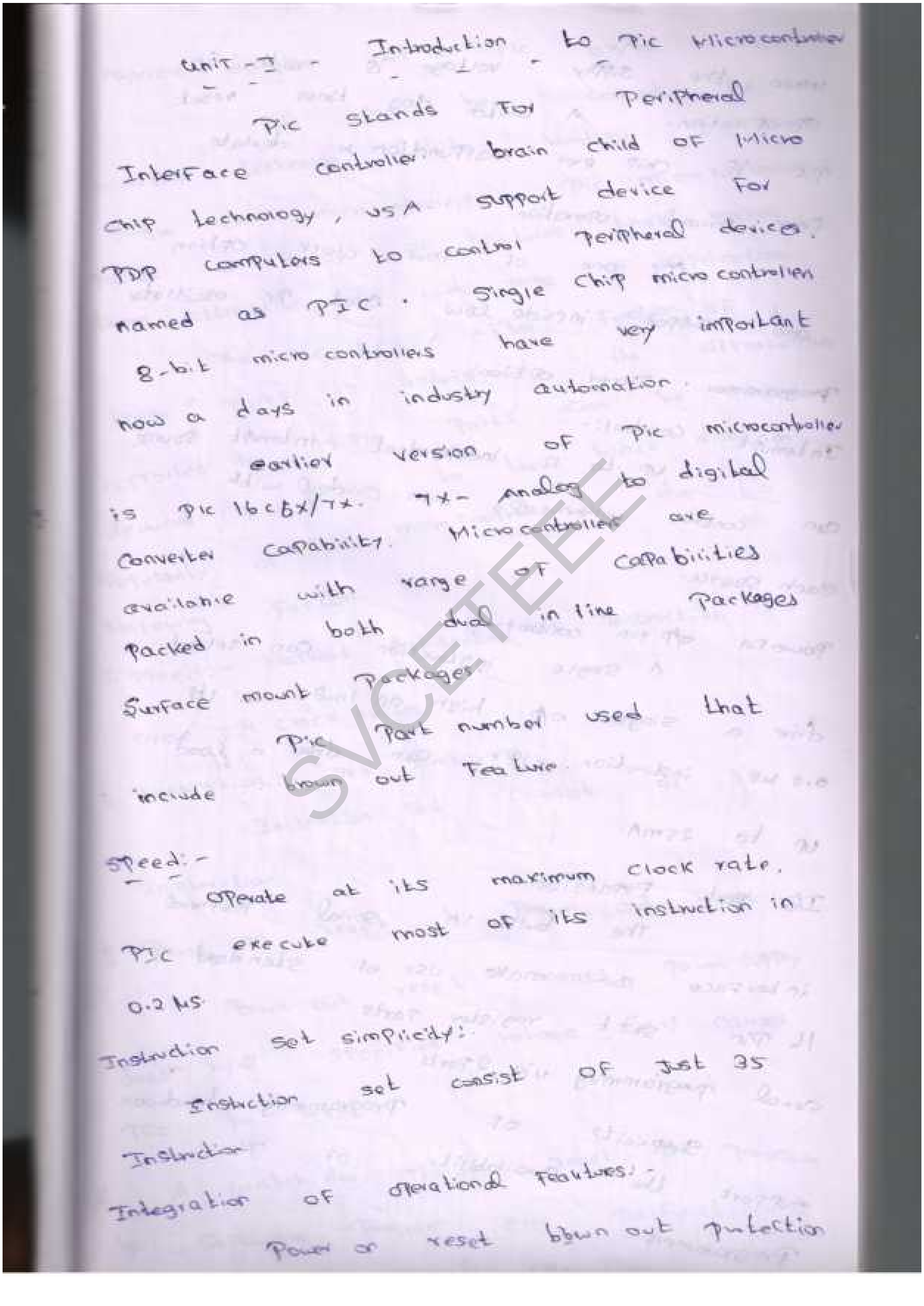
In unit 3 we discuss about the peripherals & the interfacing. I²c Bus For peripherals chip Acces & its bus operation - Bus subroutines & serial Eprom.



Analog to Digital convert & UART & the Baud rate Selection & the data handing circuit & the initiarization and the keyboard interfacing & the LCD & the ADC, DAC & sensor interfacing.

In unit 4 we discuss about the introduction to Arm processor & the architecture of Arm processor & Arm Programmer's model & the development of Arm tools and the memory Hierarchy. & the Arm Assembly language Programming. and its simple examples also we are discuss about the architectural support for operating systems.

In unit 5 we discuss about the Arm organization & the 3stage pipe line Arm organization, the 5 stage pipe line Arm organization, the Arm instruction execution process, Arm implementation & the Arm instruction set & Arm coprocessor interface and the Architectural support for Higher level languages & the Embedded RAM Applications.



unit - I - Introduction to pic Microcontroller

Pic stands for Peripheral Interface controller brain child of Micro chip technology, USA support device for PDP computors to control peripheral devices. Single chip micro controller named as PIC. 8-bit micro controllers have very important now a days in industry automation.

earlier version of Pic microcontroller is pic 16c6x/7x. 7x - analog to digital converter capability. Micro controller are available with range of capabilities packed in both dual in line packages surface mount packages.

Pic part number used that include brown out feature.

Speed:-

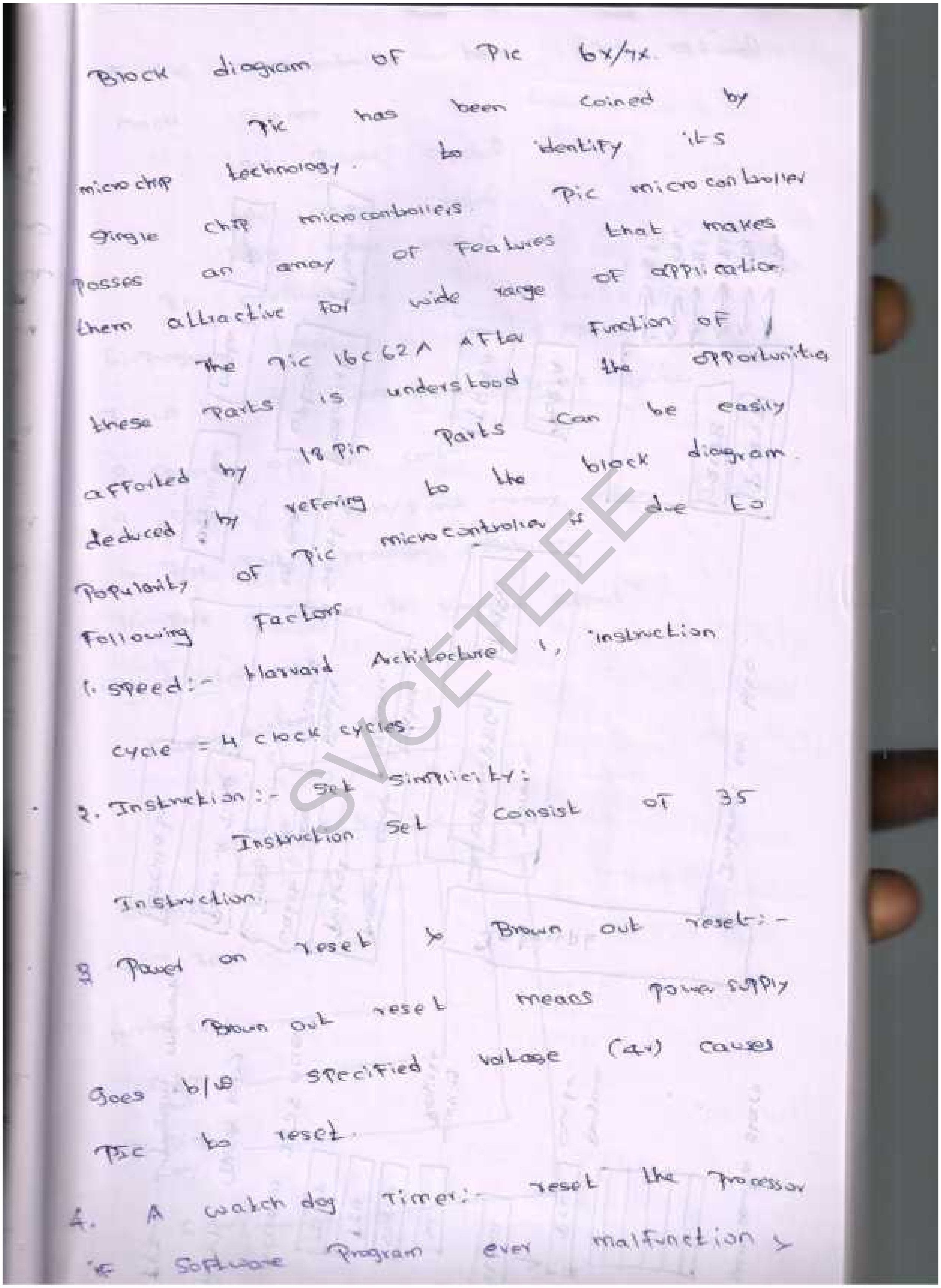
Operate at its maximum clock rate. Pic execute most of its instruction in 0.2 μs.

Instruction set simplicity:

Instruction set consist of just 35 Instruction

Integration of operational features:-

Power on reset brown out protection



Block diagram of Pic 6x/7x.

Pic has been coined by microchip technology. to identify its single chip microcontrollers. Pic microcontroller posses an array of Features that makes them attractive for wide range of application Function of

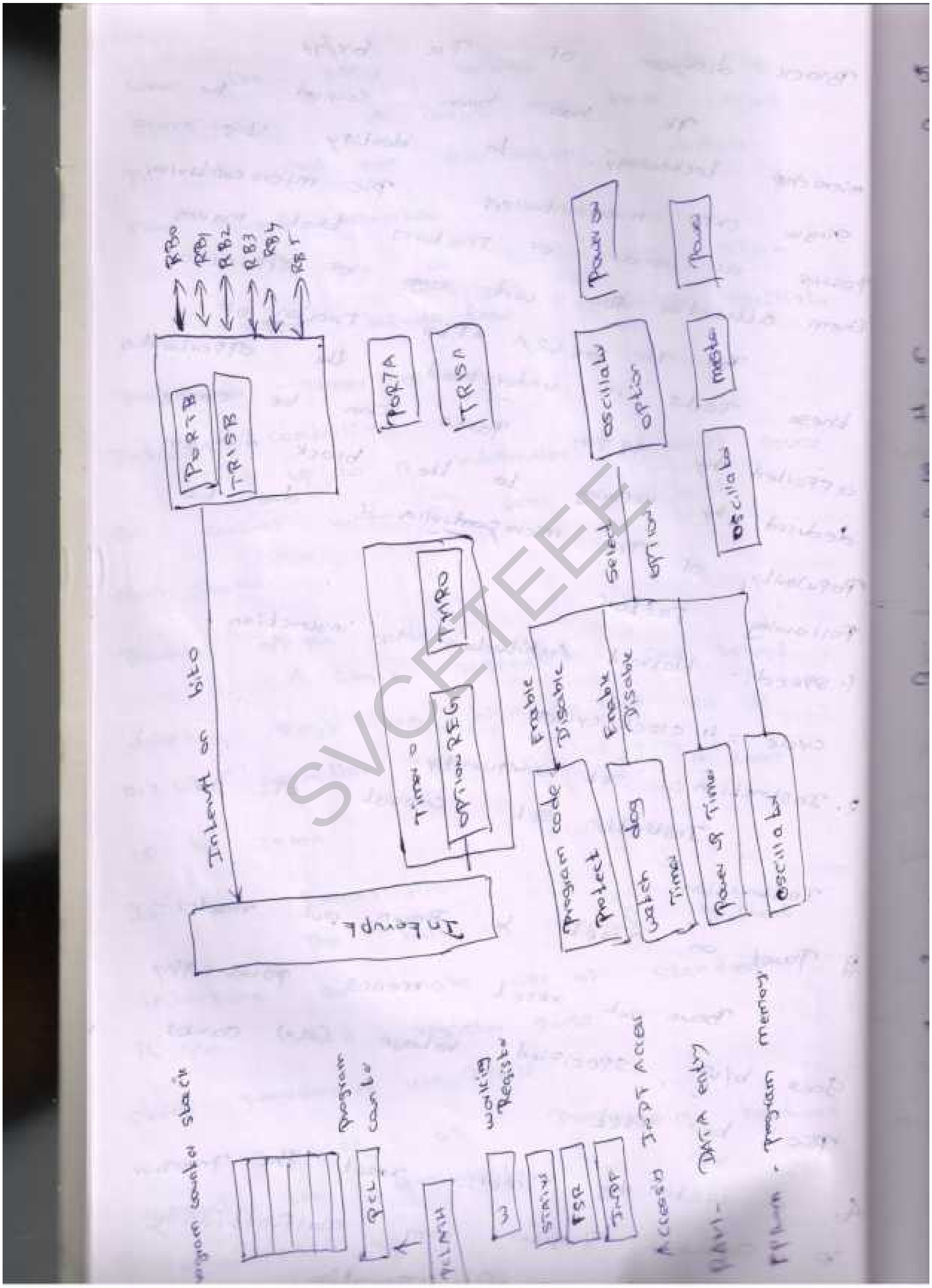
The Pic 16C62A After the opportunities these parts is understood can be easily afforded by 18 pin Parts deduced by refering to the block diagram. Popularity of Pic microcontroller is due to Following Factors

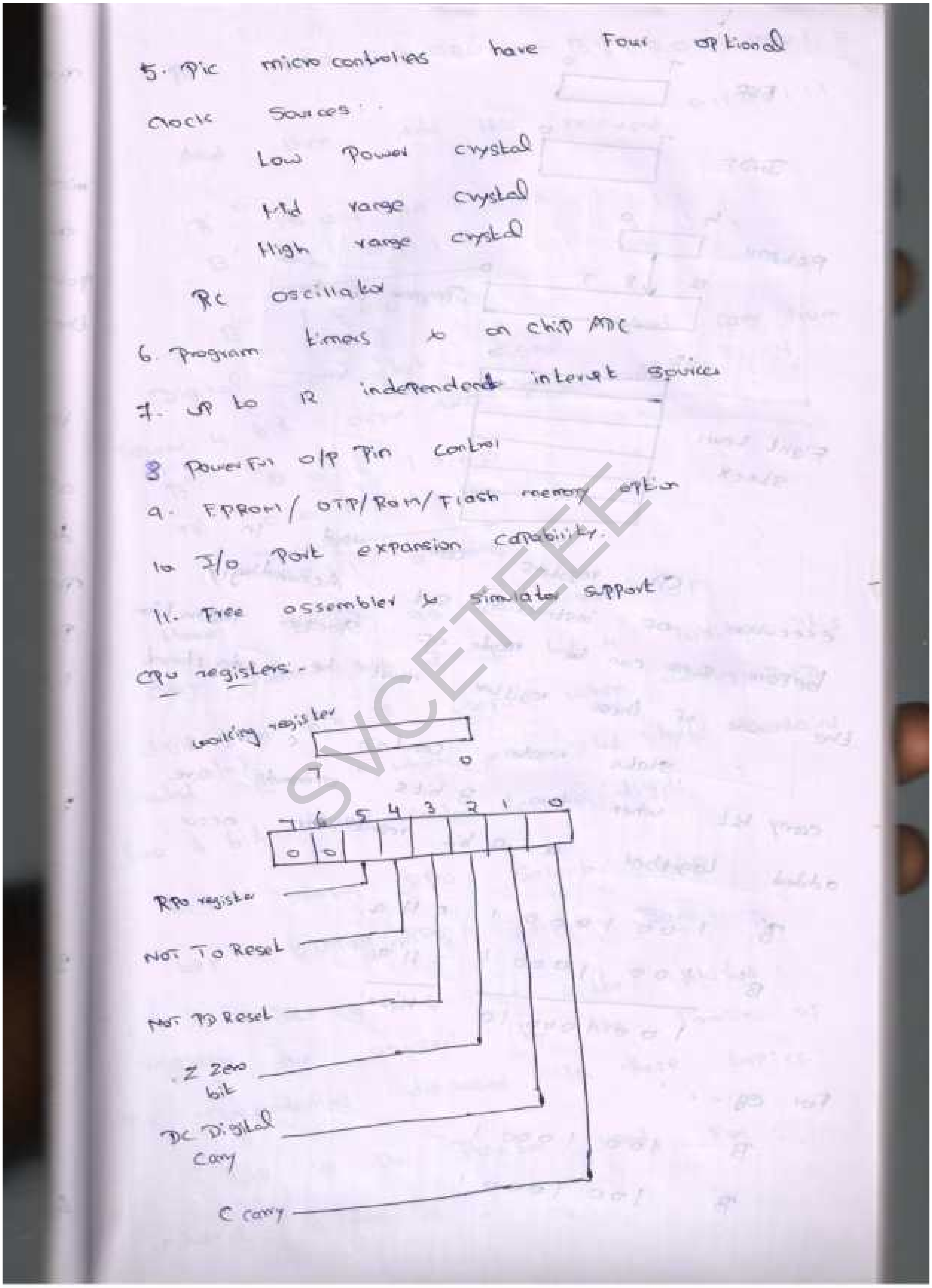
1. speed:- Harvard Architecture 1, instruction cycle = 4 clock cycles.

2. Instruction:- set simplicity:
Instruction set consist of 35 Instruction.

3. Power on reset & Brown out reset:-
Brown out reset means power supply Goes b/w specified voltage (or) causes PIC to reset.

4. A watch dog Timer:- reset the processor if Software Program ever malfunction &


Program counter stack
PCL
Program counter
PCLATH
W
working Register
STATUS
FSR
INDF
Access INDF Accer
RAM - DATA entry
FLASH - Program memory
Interrupt
Interrupt on bits
PORTB
TRISB
RB0
RB1
RB2
RB3
RB4
RB7
Timer 0
TMR0
OPTION REG
PORTA
TRISA
Program code Protect
Enable
Disable
watch dog Timer
Enable
Disable
Power up Timer
oscillator
Select option
oscillator option
oscillator
Power on
mode
Power



5. Pic micro controlles have Four optional clock Sources:

- Low Power crystal
- Mid range crystal
- High range crystal
- RC oscillator

6. Program timers & on chip ADC

7. up to 12 independent interrupt Sources

8 Powerful o/p Pin control

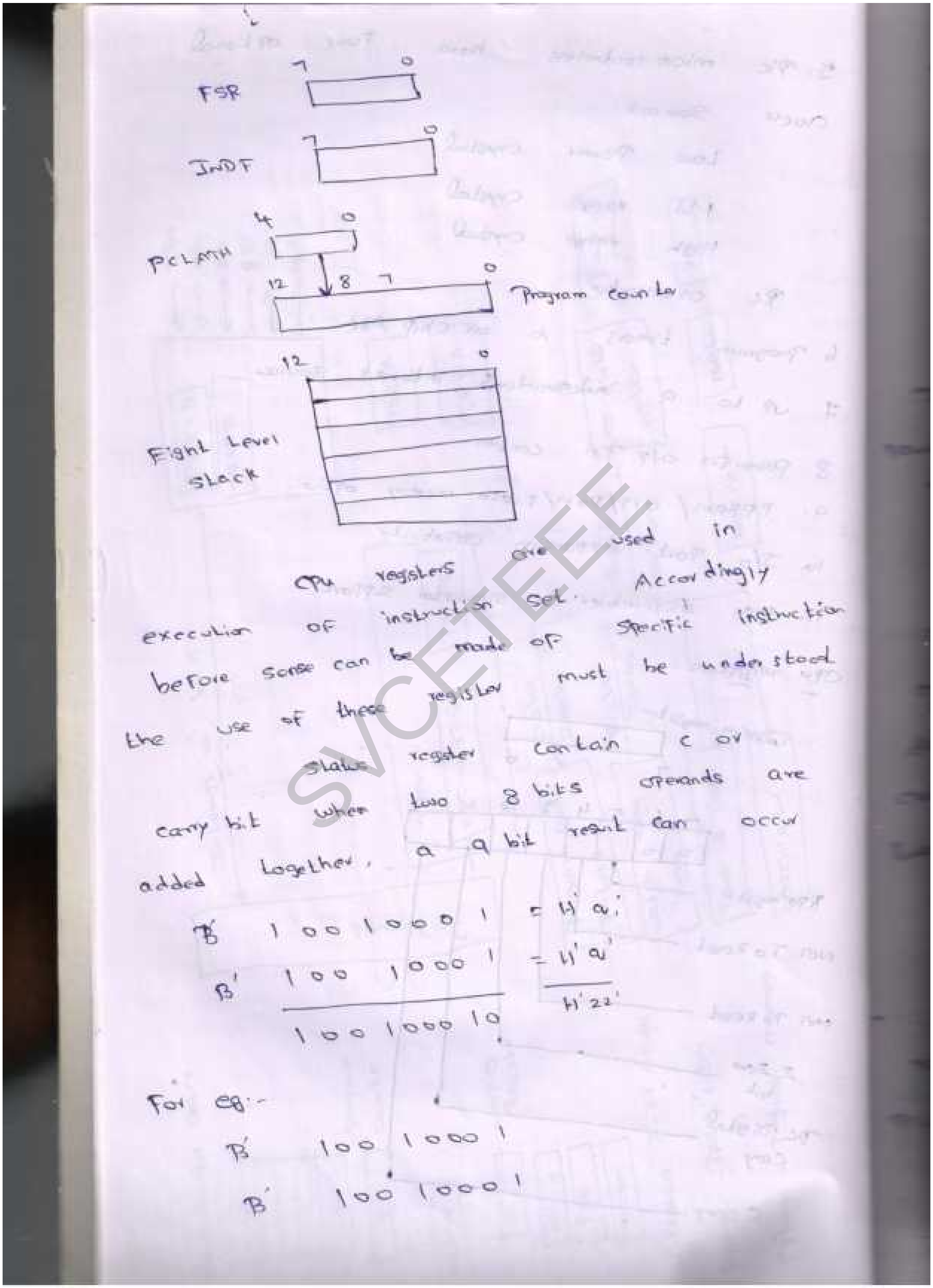
9. EPROM/ OTP/ Rom/ Flash memory option

10 I/O Port expansion capability.

11. Free assembler & simulator support

CPU registers -

working register
7
0
7 6 5 4 3 2 1 0
0 0
RP0 register
Not TO Reset
Not PD Reset
Z zero bit
DC Digital carry
C carry


7
0
FSR
7
0
INDF
4
0
PCLATH
12
8
7
0
Program counter
12
0
Eight Level
Stack

CPU registers are used in instruction set. Accordingly execution of specific instruction before sense can be made of the use of these register must be understood

Status register contain C or carry bit when two 8 bits operands are added together, a 9 bit result can occur

B' 1 00 1000 1 = H' a'

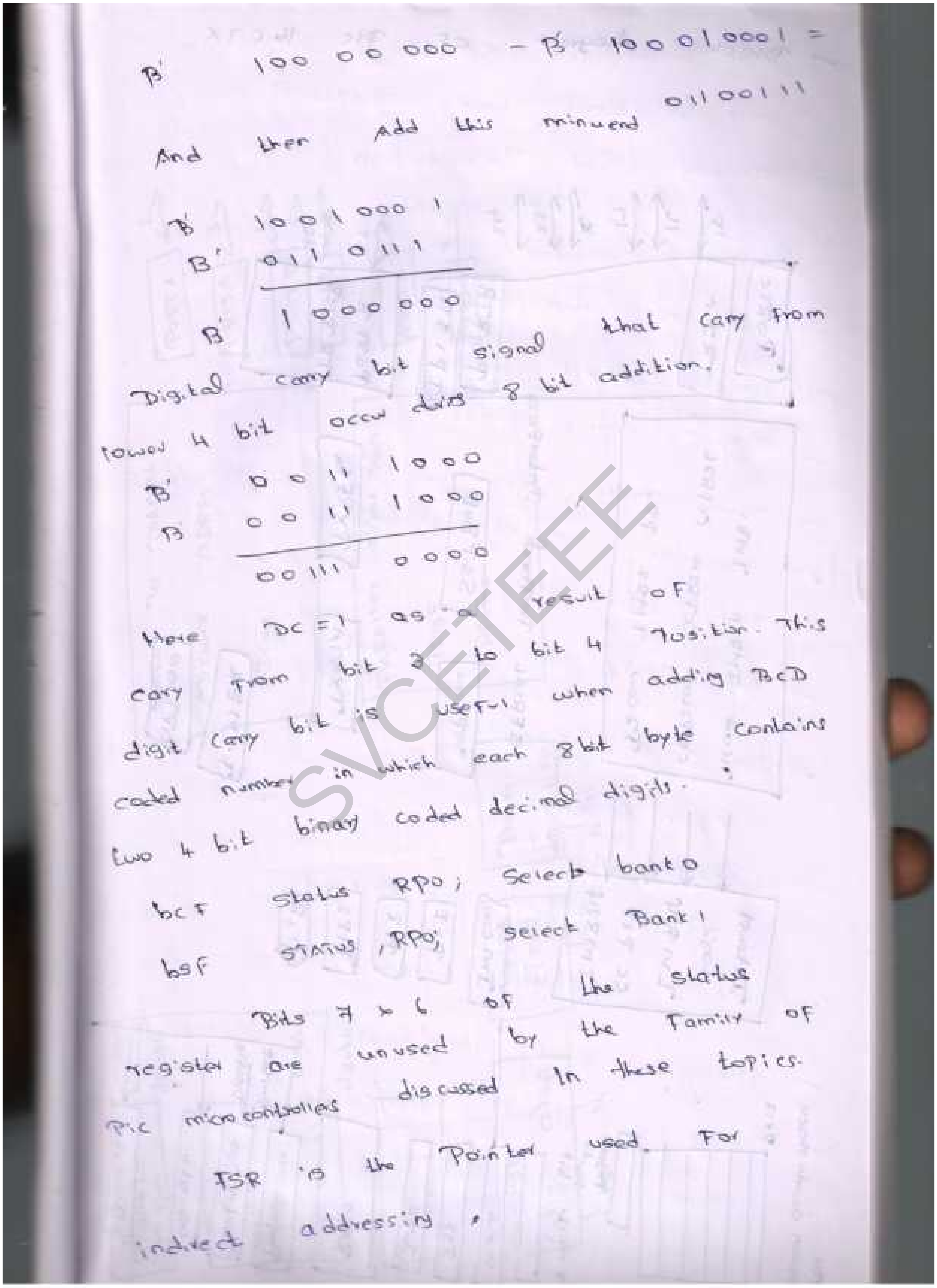
B' 1 00 1000 1 = H' a'

1 00 1000 10 H' 22'

For eg:-

B' 100 1000 1

B' 100 1000 1



B' 100 00 000 - B' 1000 1 0001 = 01100111

And then add this minuend

B' 1001 000 1
B' 011 0111

B' 1 000 0000

Digital carry bit signal that carry from lower 4 bit occur during 8 bit addition.

B' 0 0 11 1000
B 0 0 11 1000

00 111 0000

Here DC=1 as a result of carry from bit 3 to bit 4 position. This digit carry bit is useful when adding BCD coded number in which each 8 bit byte contains two 4 bit binary coded decimal digits.

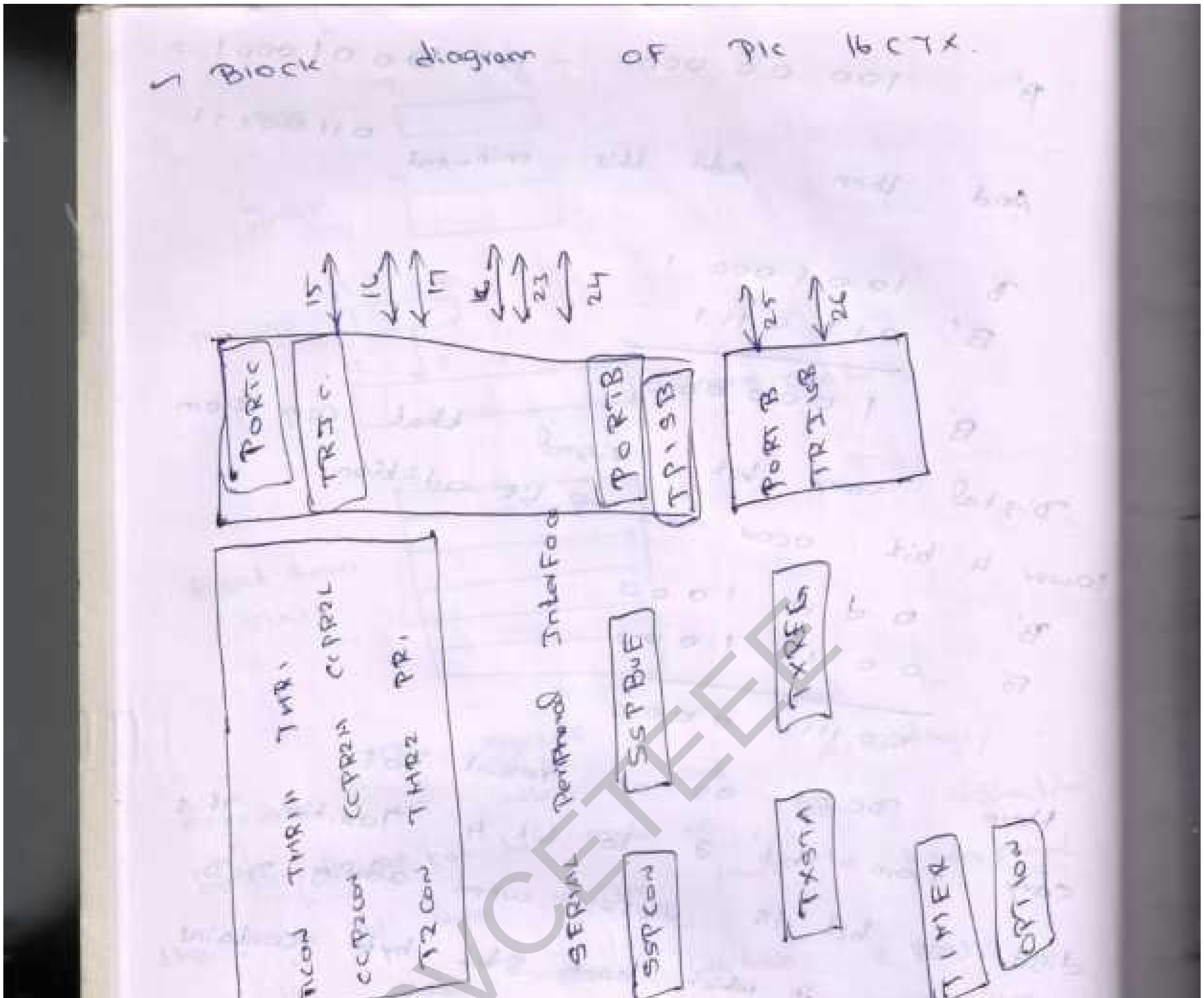
bcf Status RP0; Select bank 0

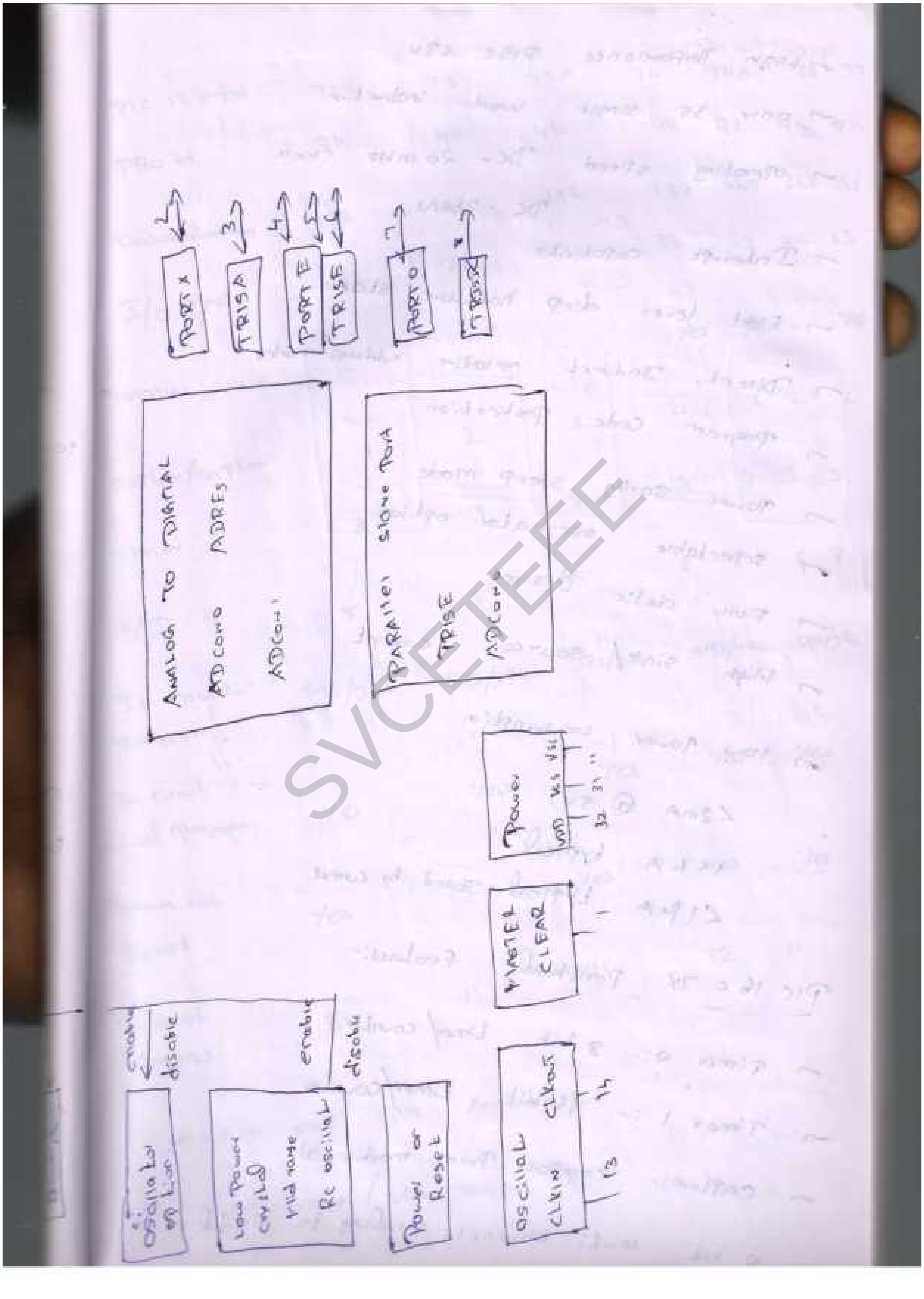
bsf STATUS, RP0; select Bank 1

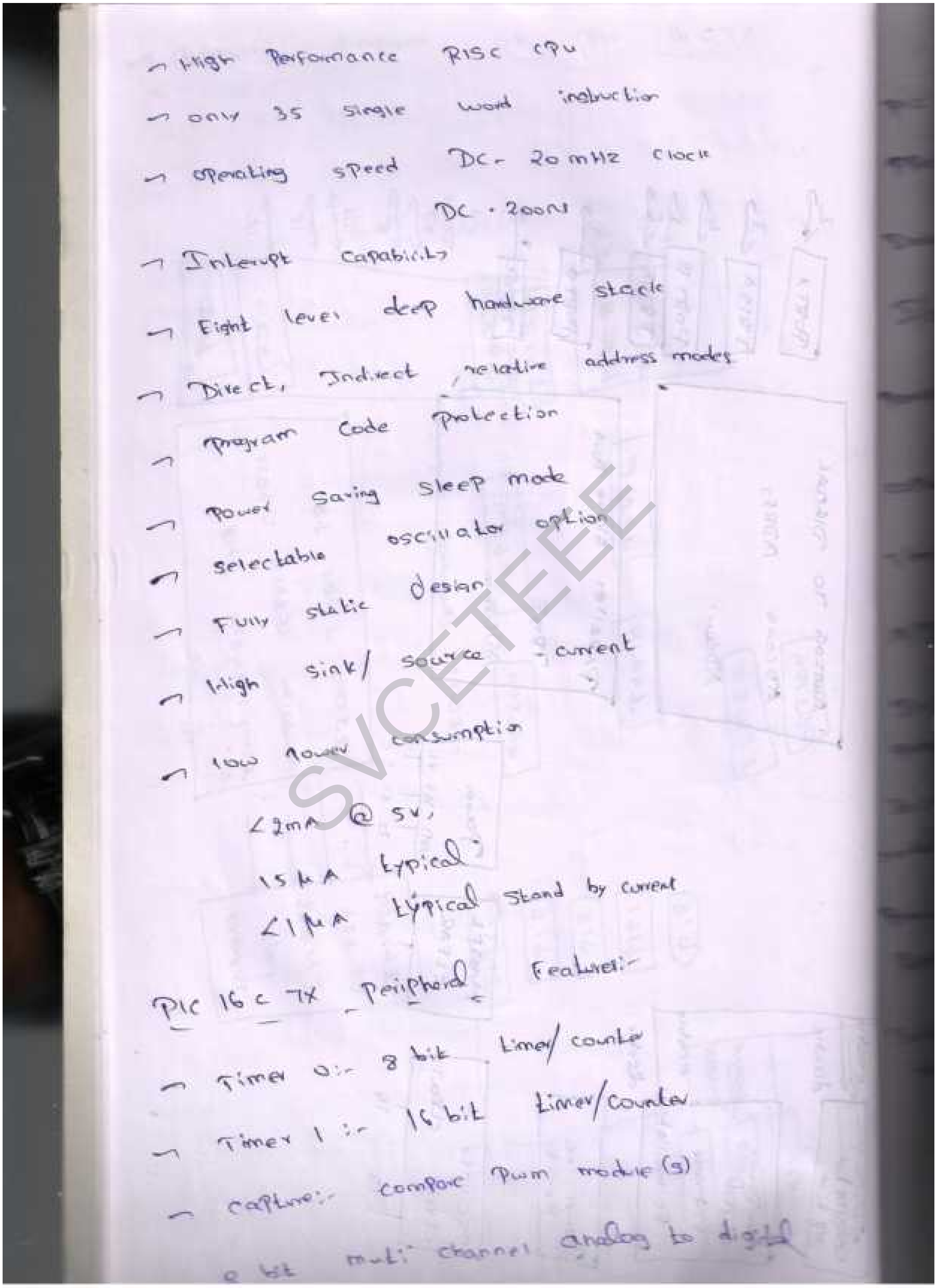
Bits 7 & 6 of the status register are unused by the Family of Pic microcontrollers discussed In these topics.

FSR is the Pointer used For indirect addressing.

→ Block diagram of PIC 16C7X.




PORT X
TRISA
PORT E
TRISE
PORT D
TRISD
2
3
4
5
6
7
Analog to Digital
ADCON0
ADRES
ADCON1
Parallel Slave Port
TRISE
ADCON1
Power
VDD
VSS
VSS
32
31
11
MASTER CLEAR
1
oscillator option
enable
disable
Low Power circuit
Mid range
RC oscillaL
enable
disable
Power on Reset
oscillaL
CLKIN
CLKOUT
13
14

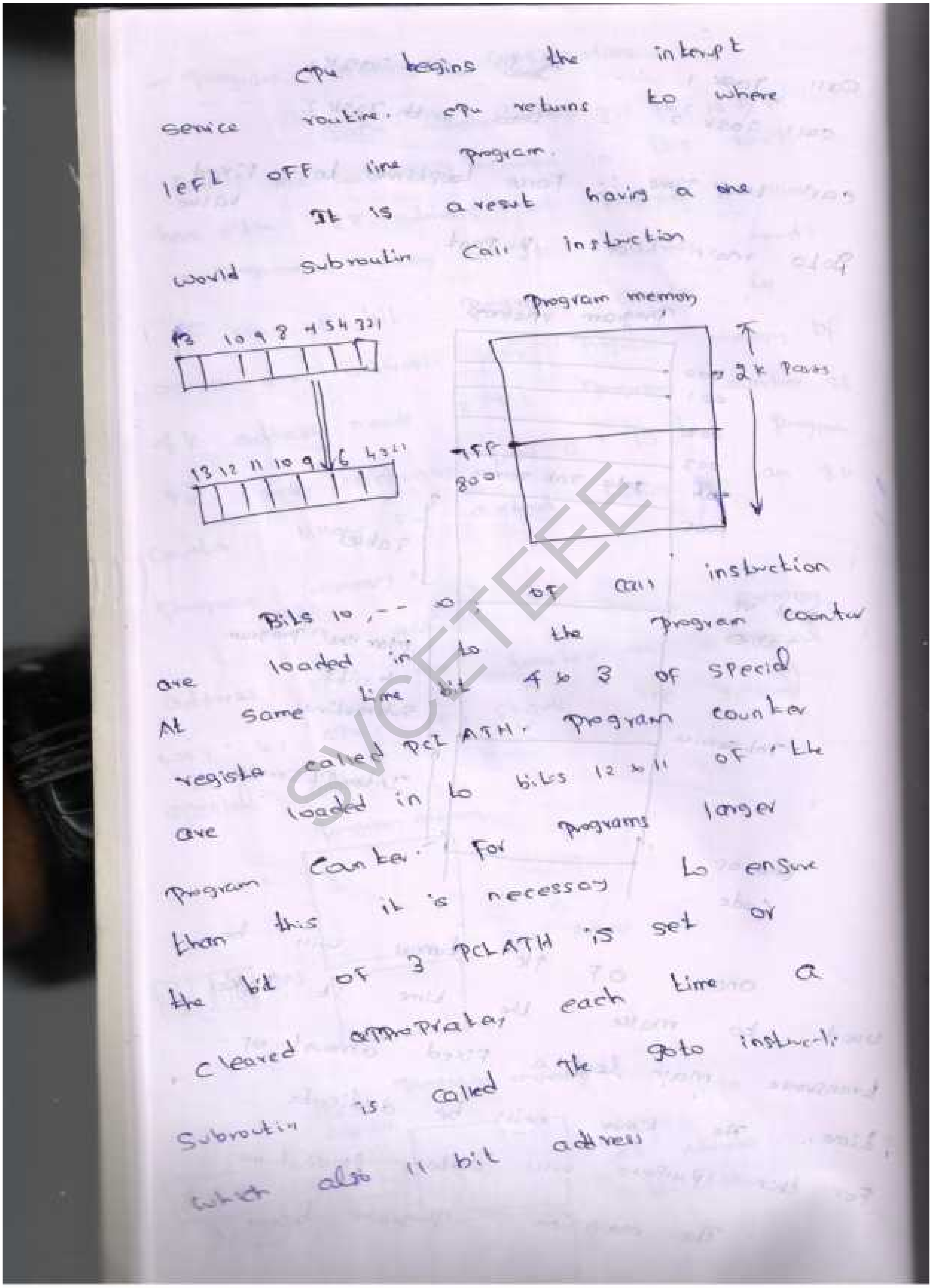


- High Performance RISC CPU
- only 35 single word instruction
- operating speed DC - 20 mHz clock
  DC - 200ns
- Interupt capability
- Eight level deep hardware stack
- Direct, Indirect, relative address modes
- Program code protection
- power saving sleep mode
- selectable oscillator option
- Fully static design
- High sink/ source current
- low power consumption
  <2mA @ 5v,
  15 μA typical
  <1 μA typical stand by current

PIC 16 c 7X peripheral Features:-

- Timer 0:- 8 bit timer/ counter
- Timer 1 :- 16 bit timer/counter
- capture:- compare pwm module (s)

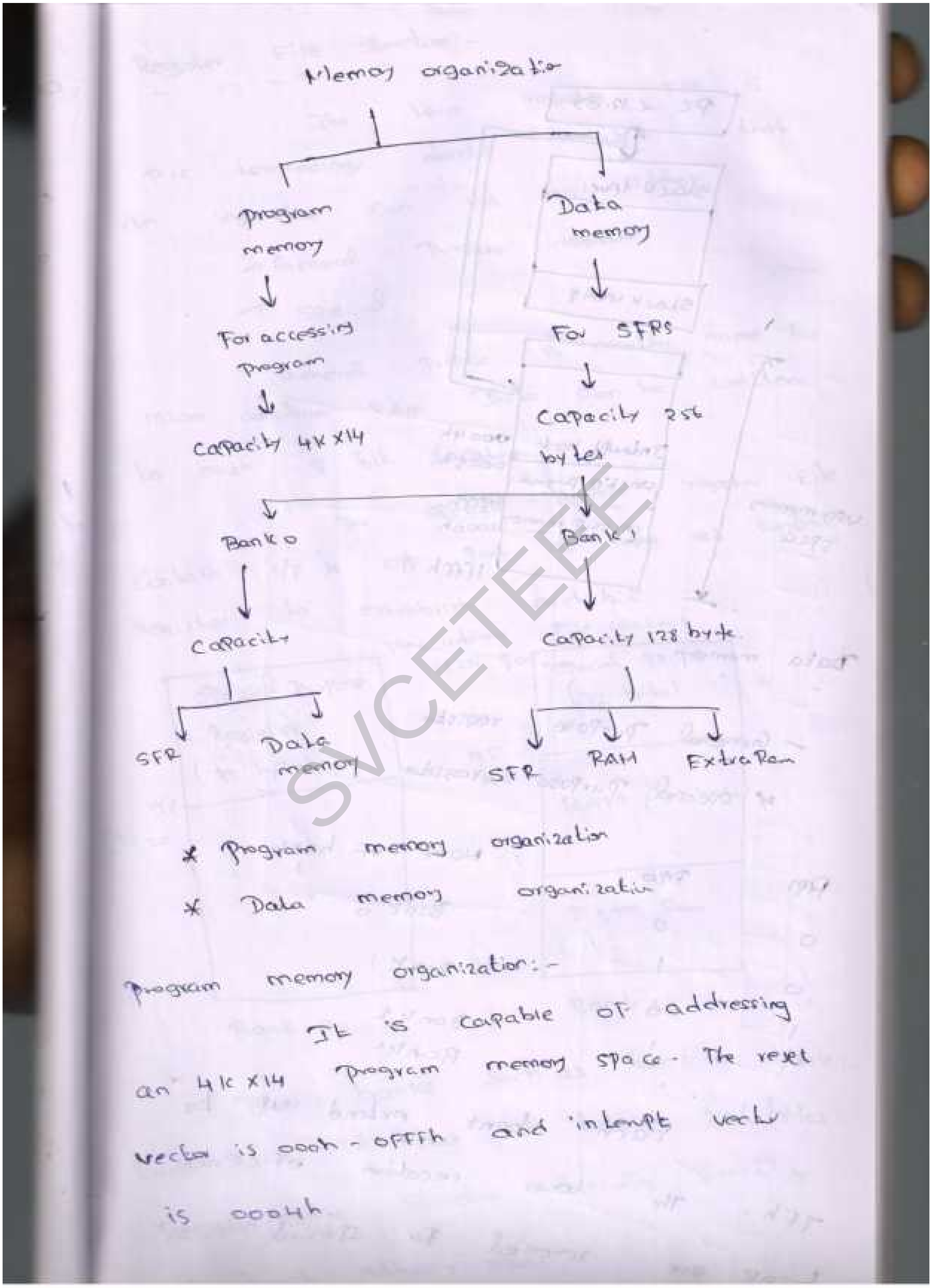
8 bit multi channel analog to digital


13 10 9 8 7 5 4 3 2 1
13 12 11 10 9 6 4 3 2 1
Program memory
7FF
800
2k pages

cpu begins the interupt service routine. cpu returns to where left off the program.

It is a resut having a one world subroutin call instruction

Bits 10, - - 0 of call instruction are loaded in to the program counter At same time bit 4 & 3 of special register called PCLATH. program counter are loaded in to bits 12 & 11 of the program counter. For programs longer than this it is necessary to ensure the bit of 3 PCLATH is set or cleared appropriately each time a subroutin is called the goto instructi which also 11 bit address

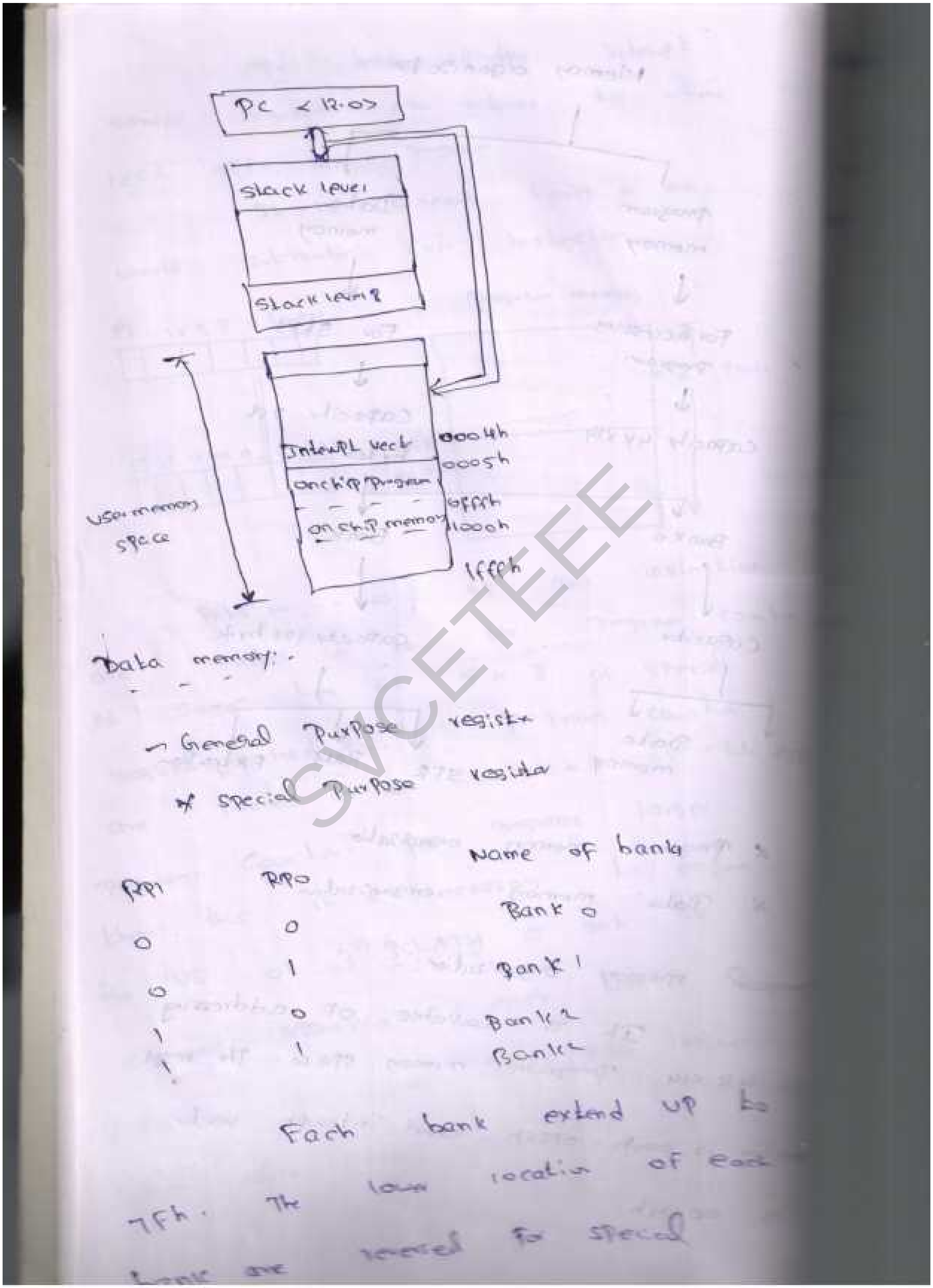

Memory organization
Program memory
For accessing program
Capacity 4K X14
Data memory
For SFRS
Capacity 256 bytes
Bank 0
Capacity
SFR
Data memory
Bank 1
Capacity 128 byte
SFR
RAM
Extra Ram

* Program memory organization

* Data memory organization

Program memory organization:-

It is capable of addressing an 4K X14 Program memory space. The reset vector is 000h - 0FFFh and interrupt vector is 0004h.



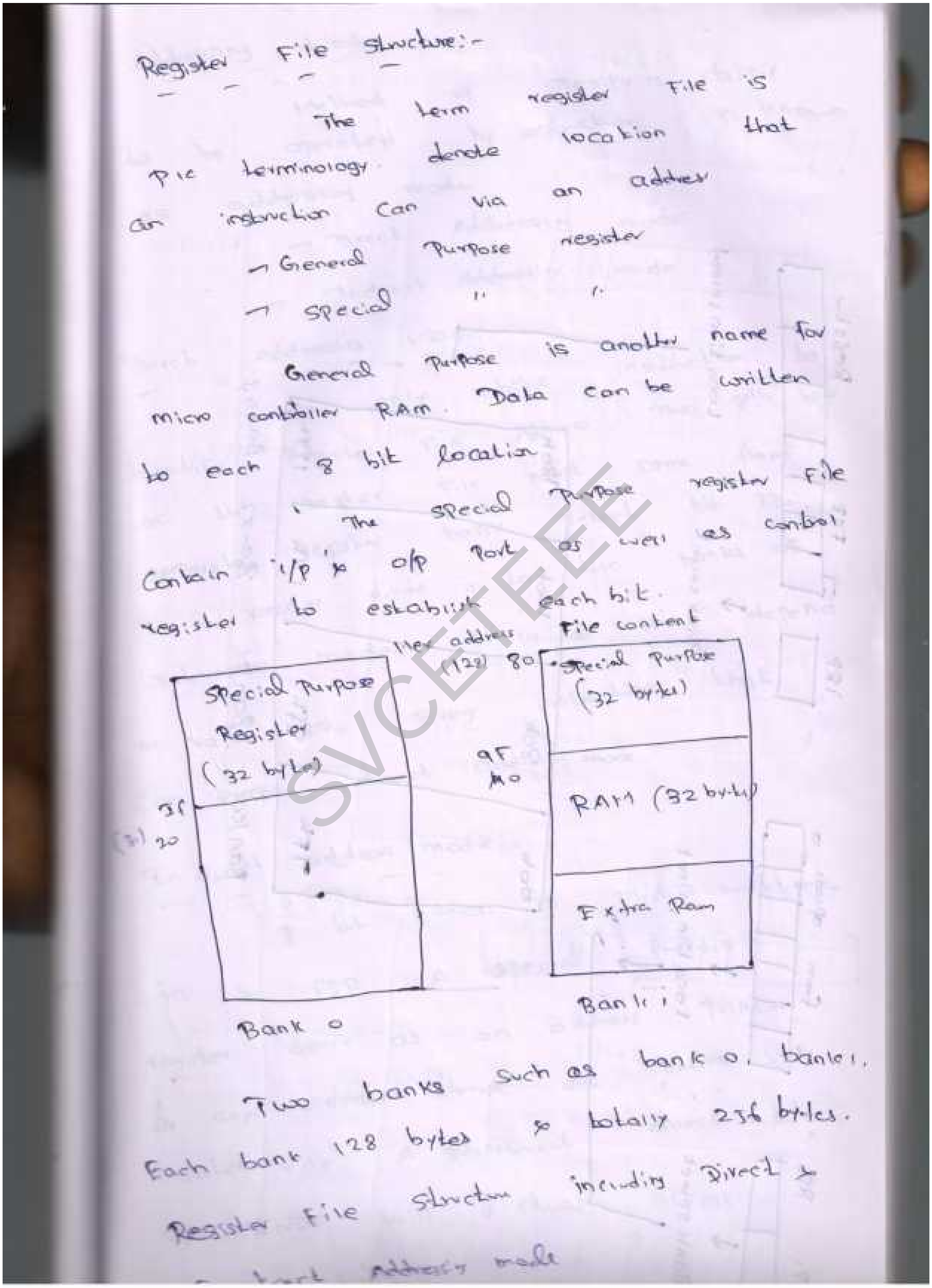
PC <12:0>
Stack level 1
Stack level 8
Interupt vect
0004h
onchip program
0005h
07FFh
on chip memory
1000h
1FFFh
user memory space

Data memory:-

- General purpose register
- * special purpose register

RP1	RP0	Name of bank
0	0	Bank 0
0	1	Bank 1
1	0	Bank 2
1	1	Bank3

Each bank extend up to 7Fh. The lower location of each bank are reserved for special



Register File structure:-

The term register File is Pic terminology denote location that an instruction can via an addres

- General Purpose register
- Special " "

General Purpose is another name for micro controller RAM. Data can be written to each 8 bit location

. The special Purpose register File Contain i/p & o/p Port as well as control register to establish each bit.

Bank 0:

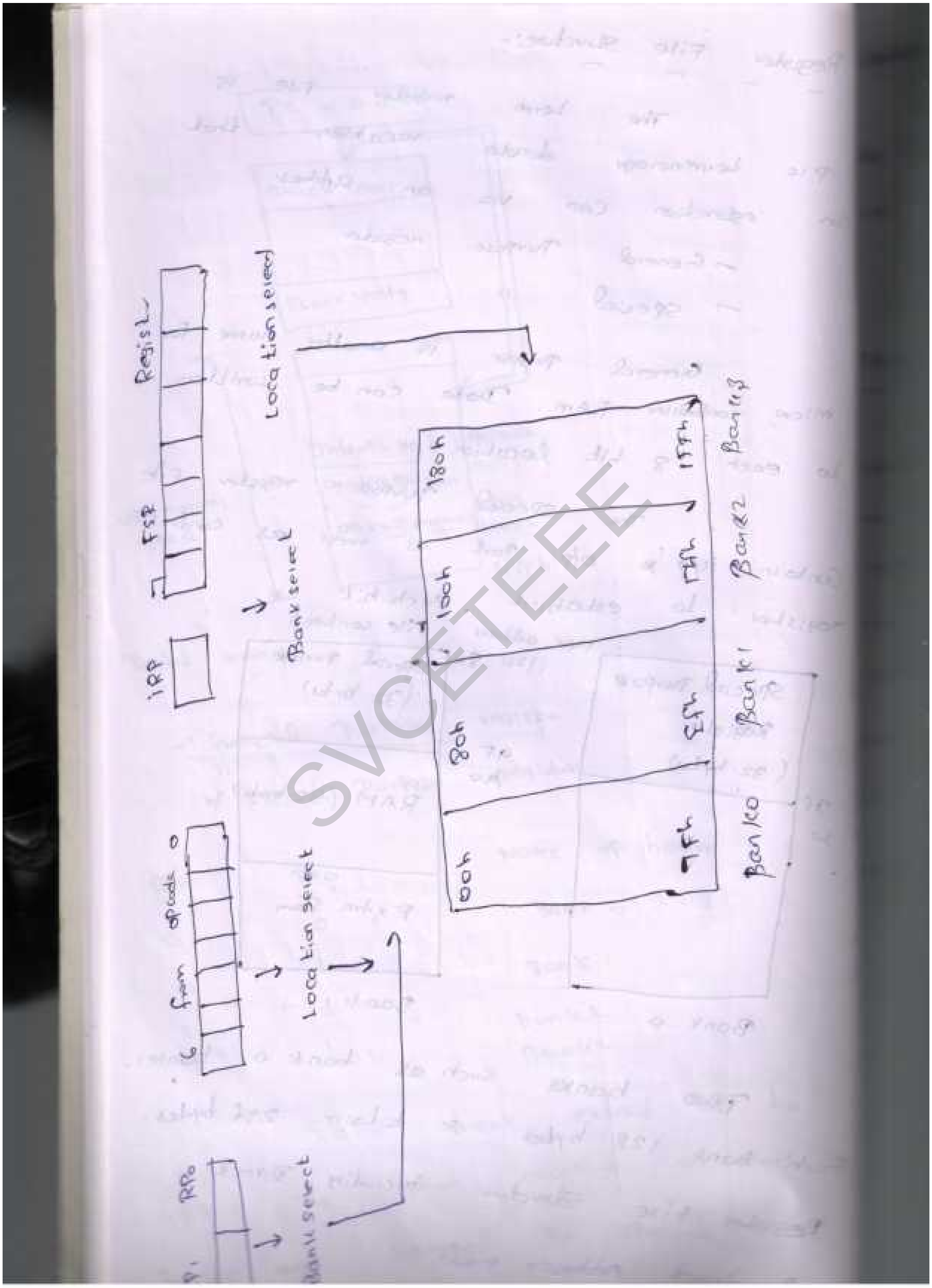
- Special Purpose Register (32 bytes)
- 3F
- 20
- (31)

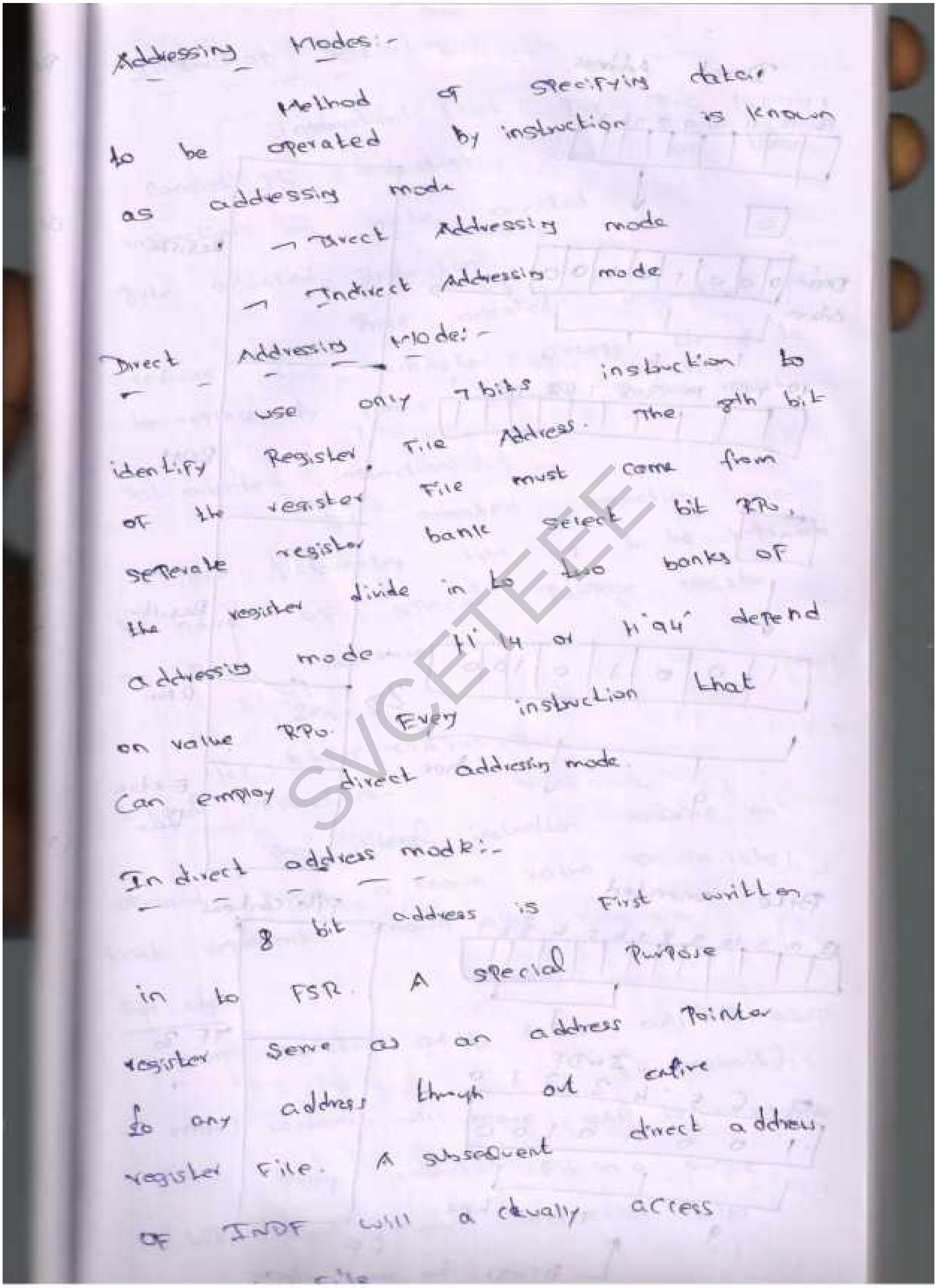
Bank 1:

- Hex address (128)
- File content
- 80
- Special Purpose (32 bytes)
- 9F
- A0
- RAM (32 bytes)
- Extra Ram

Two banks such as bank 0, bank 1. Each bank 128 bytes & totally 256 bytes. Including Direct & Register File structure

~ bank address mode


RP0
Bank select
6 from opcode 0
Location select
IRP
Bank select
FSR
Register
Location select
00h
7Fh
80h
FFh
100h
17Fh
180h
1FFh
Bank0
Bank1
Bank2
Bank3



Addressing Modes:-

Method of specifying data to be operated by instruction is known as addressing mode

→ Direct Addressing mode

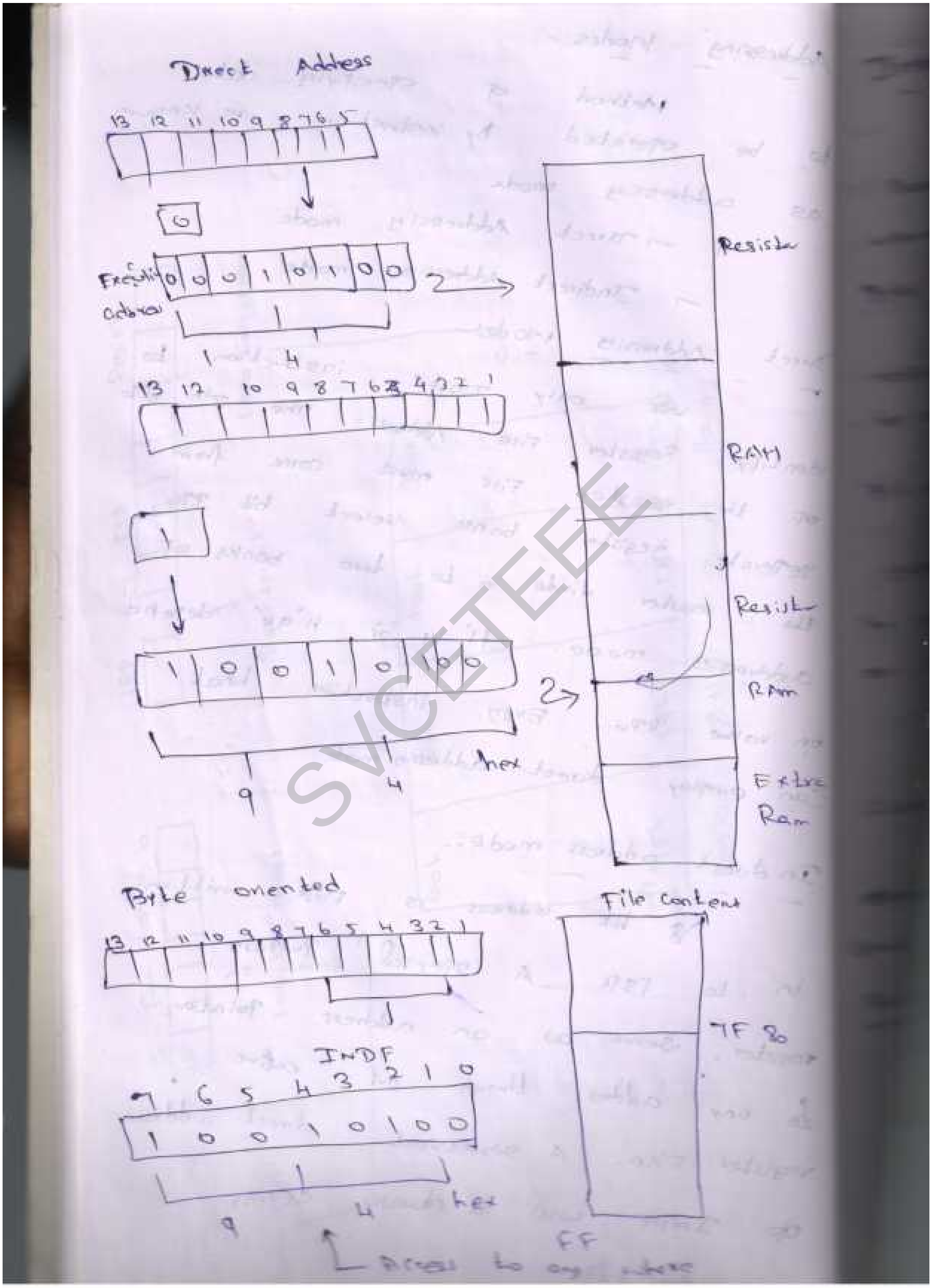
→ Indirect Addressing mode

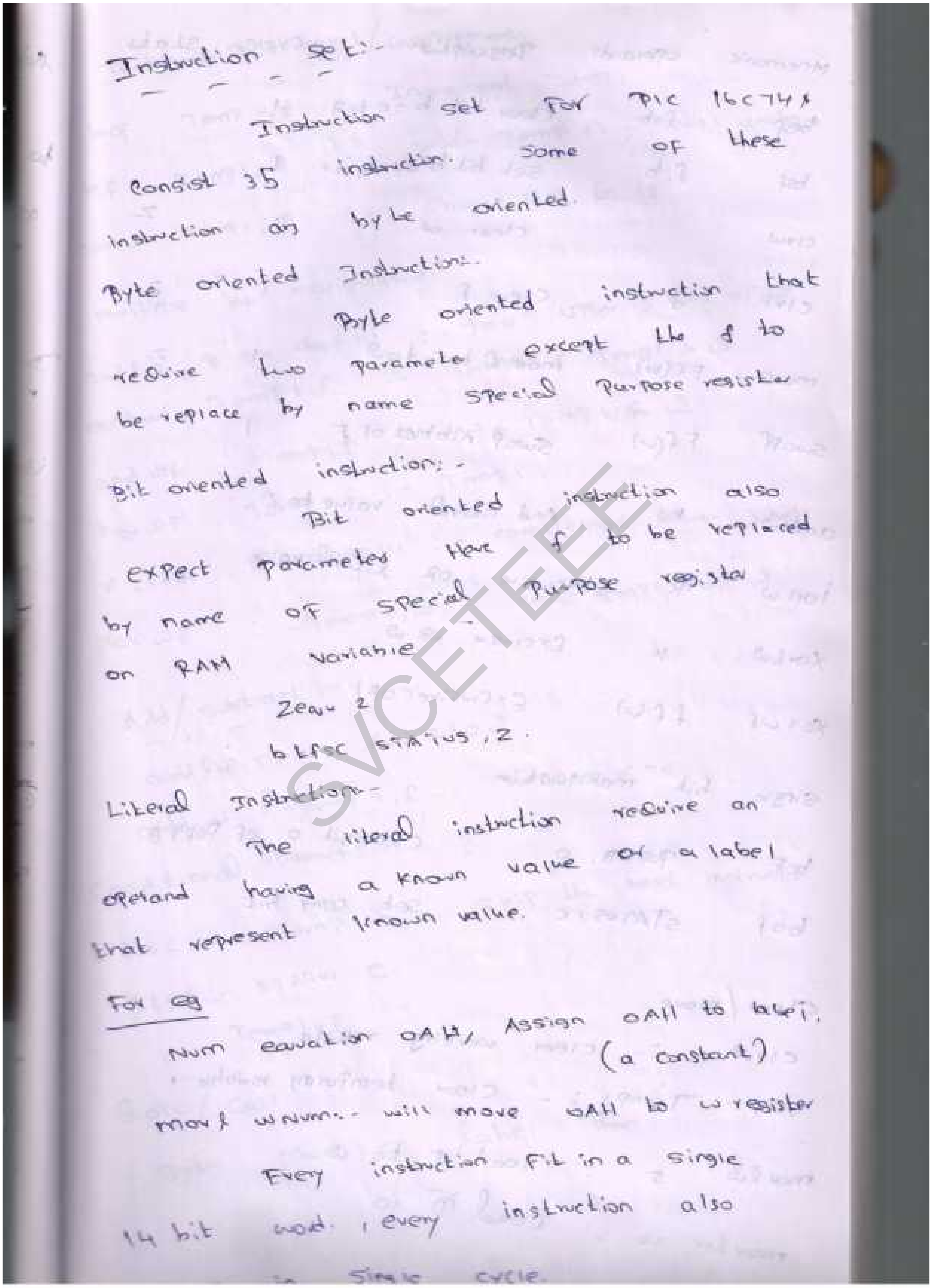
Direct Addressing Mode:-

use only 7 bits instruction to identify Register File Address. The 8th bit of the register file must come from seperate register bank select bit RP0, the register divide in to two banks of addressing mode. h'14 or h'94' depend on value RP0. Every instruction that can employ direct addressing mode.

Indirect address mode:-

8 bit address is first written in to FSR. A special purpose register serve as an address pointer to any address through out entire register File. A subsequent direct address of INDF will actually access


Direct Address
13 12 11 10 9 8 7 6 5
0
Effective address
0 0 0 1 0 1 0 0
1
4
13 12 10 9 8 7 6 5 4 3 2 1
1
1 0 0 1 0 1 0 0
9
4
hex
Resiste
RAM
Resist
RAm
Extra
Ram
Byte oriented
13 12 11 10 9 8 7 6 5 4 3 2 1
INDF
7 6 5 4 3 2 1 0
1 0 0 1 0 1 0 0
9
4
hex
Access to any where
File content
7F 80
FF



Instruction set:-

Instruction set for PIC 16C74A consist 35 instruction. some of these instruction are byte oriented.

Byte oriented Instruction:-

Byte oriented instruction that require two parameter except the f to be replace by name special purpose register

Bit oriented instruction:-

Bit oriented instruction also expect parameter Here f to be replaced by name of special purpose register on RAM variable.

Zeau 2

btfsc STATUS, Z.

Literal Instruction:-

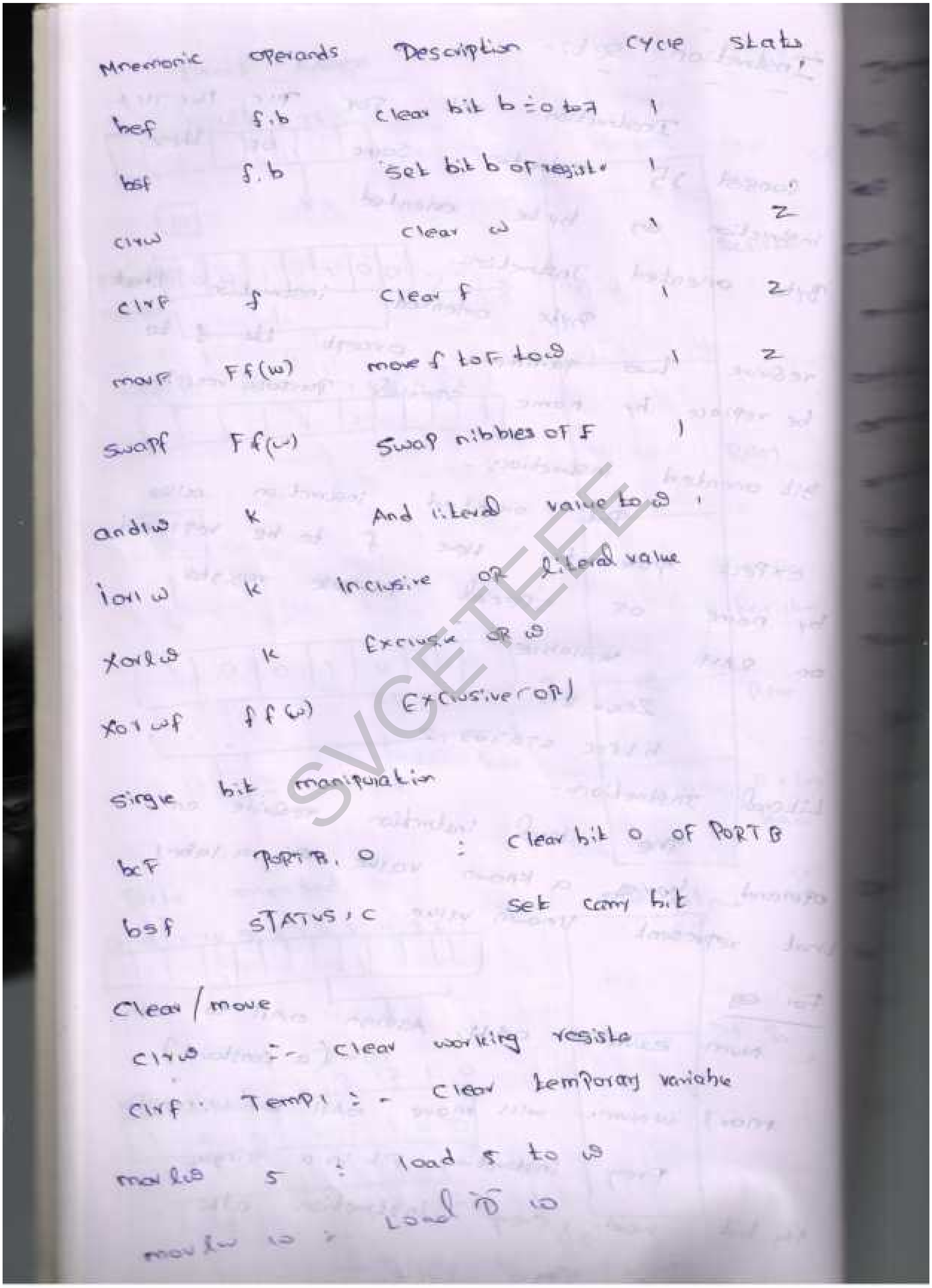
The literal instruction require an operand having a known value or a label that represent known value.

For eg

Num equation 0AH, Assign 0AH to label, (a constant)

movlw Num:- will move 0AH to w register

Every instruction fit in a single 14 bit word, every instruction also in single cycle.



Mnemonic	operands	Description	cycle	Stats
bcf	f.b	clear bit b = 0 to 7	1	
bsf	f.b	set bit b of registe	1	
clrw		clear w	1	Z
clrf	f	Clear f	1	Z
movf	Ff(w)	move f to F to w	1	Z
swapf	Ff(w)	swap nibbles of F	1	
andlw	k	And literal value to w	1	
iorlw	k	Inclusive OR literal value		
xorlw	k	Exclusive OR w		
xorwf	f f(w)	Exclusive OR)		

single bit manipulation

bcF PORTB, 0 : clear bit 0 of PORTB

bsf STATUS, C set carry bit

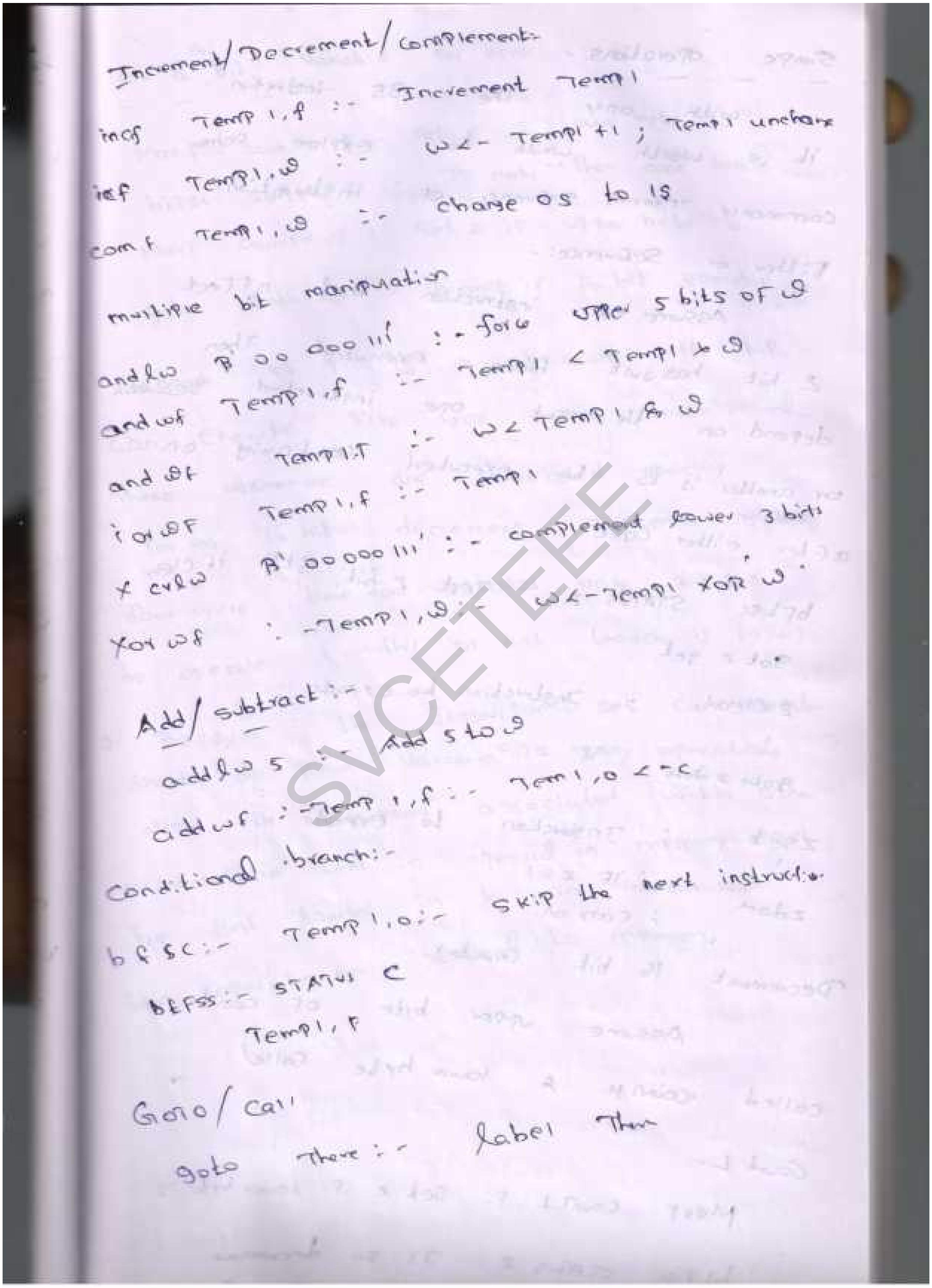
Clear / move

clrw :- clear working registe

clrf Temp1 :- clear temporary variable

movlw 5 : load 5 to w

movlw 10 : Load 10 to 10



Increment/ Decrement/ complement:-

incf Temp 1,f :- Increment Temp1

ief Temp1,w :- w<- Temp1 +1 ; Temp1 unchange

com f Temp1,w :- change 0s to 1s.

multiple bit manipulation

andlw B 00 000 111' :- force upper 5 bits of w

andwf Temp1,f :- Temp1 < Temp1 & w

and wf Temp1,T :- w< Temp1 & w

and wf Temp1,f :- Temp1

iorwF Temp1,f :- complement lower 3 bits

x orlw B' 00000 111' :-

xorwf :- Temp1,w :- w<-Temp1 XOR w.

Add/ subtract:-

addlw 5 :- Add 5 to w

addwf :- Temp 1,f :- Tem1,0 <-c

Conditional branch:-

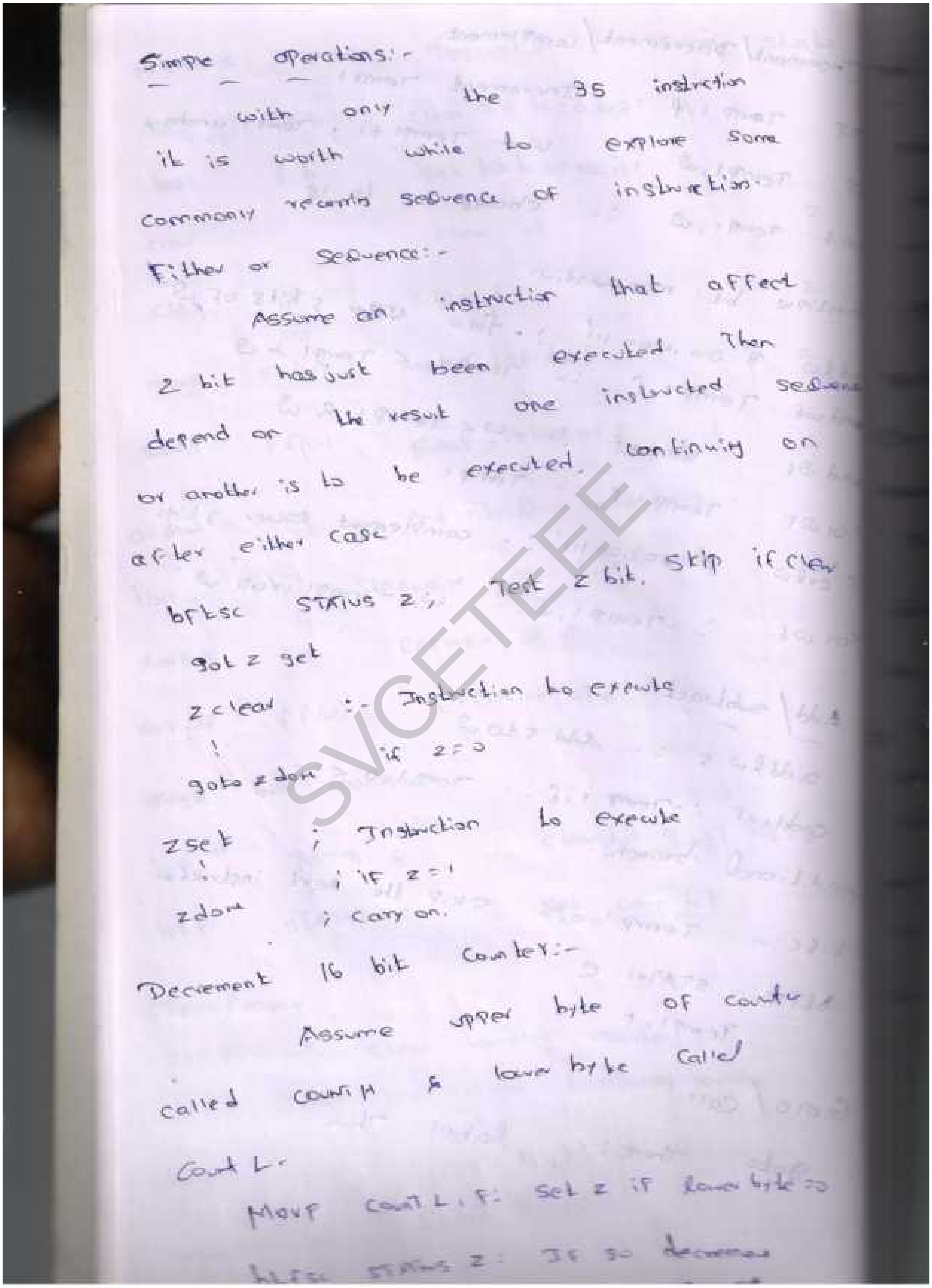
bfsc:- Temp1,0:- skip the next instruction

bfss:- STATUS C

Temp1, F

Goto/ Call

goto There :- label Then



Simple operations:-

with only the 35 instruction it is worth while to explore some commonly recurring sequence of instruction.

Either or sequence:-

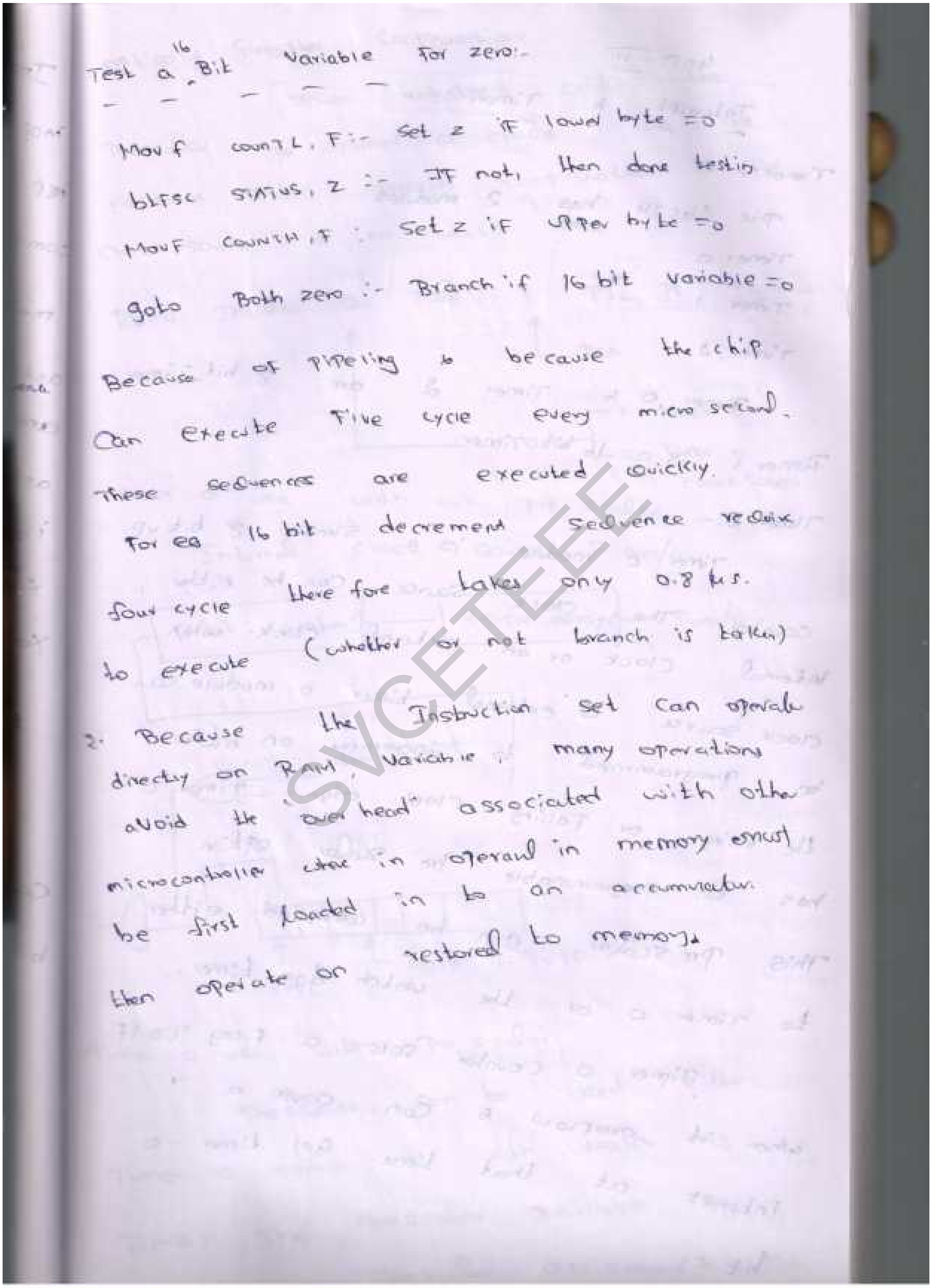
Assume an instruction that affect z bit has just been executed. Then depend on the result one instructed sequence or another is to be executed. continuing on after either case

```
btfsc   STATUS z ;  Test z bit, skip if clear
  goto z set
  z clear   :- Instruction to execute
    !             if  z=0
  goto z don
zset      ;  Instruction to execute
  !        ; IF z=1
zdon       ; carry on.
```

Decrement 16 bit counter:-

Assume upper byte of counter called count H & lower byte called count L.

```
Movf   count L, F:  set z if lower byte =0
btfsc  STATUS z:   IF so decrement
```



Test a ^{16} Bit variable for zero:-

movf countL, F :- set z if lower byte = 0

btfsc STATUS, Z :- If not, then done testin

movF countH, F :- set z if upper byte = 0

goto Both zero :- Branch if 16 bit variable = 0

Because of pipeling & because the chip can execute five cycle every micro second. These sequences are executed quickly. For eg 16 bit decrement sequence require four cycle there fore takes only 0.8 μs. to execute (whether or not branch is taken)

2. Because the Instruction set can operate directly on RAM variable, many operations avoid the "over head" associated with other microcontroller where in operand in memory must be first loaded in to an accumulator restored to memory. then operate on