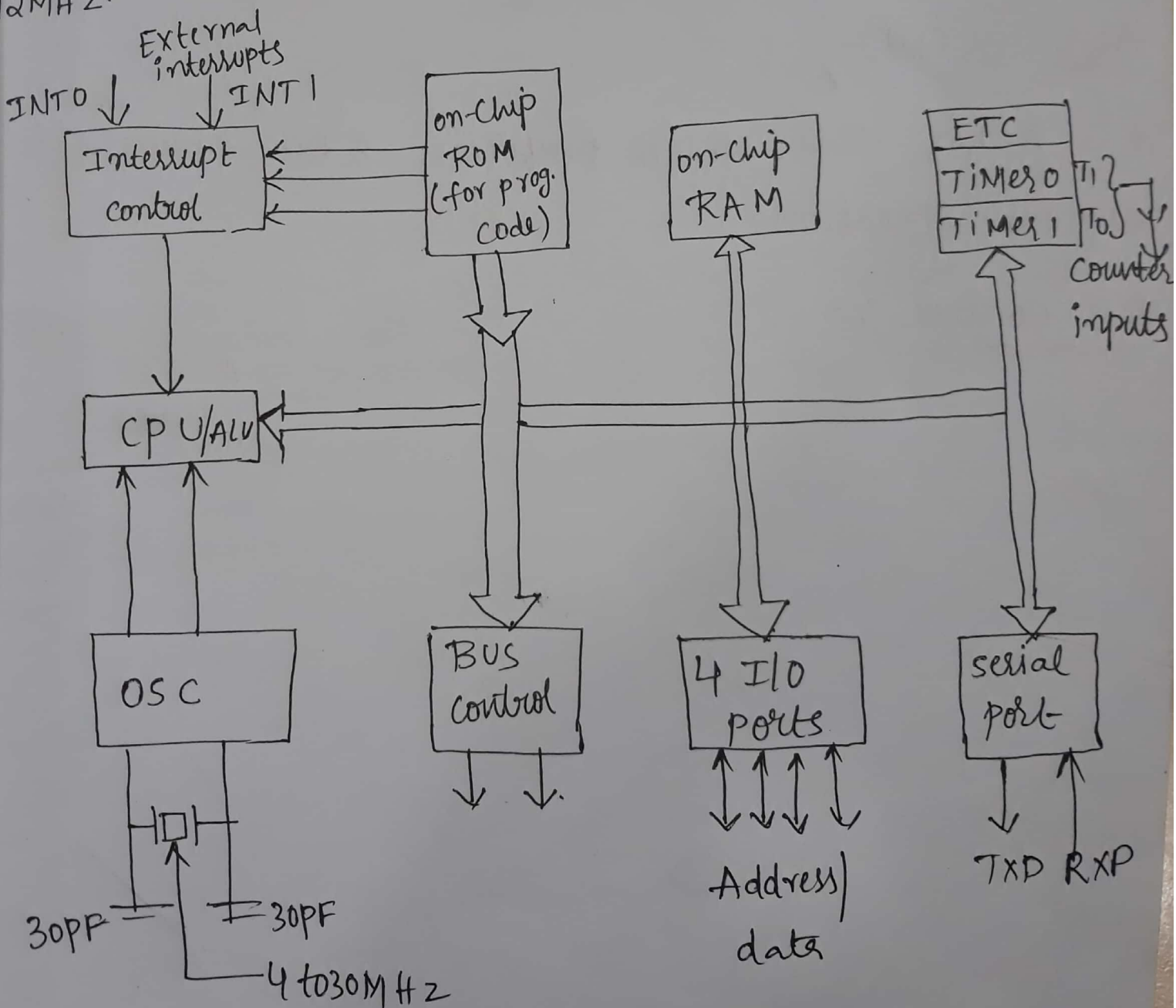
8051 Architecture :-

It is an 8-bit Microcontroller. It is built with 40 pins 4Kb ROM, 128 bytes of RAM storage, 2 16-bit timers. It consists of four parallel 8-bit ports. which can be programmable as well as addressable. An onchip crystal oscillator is integrated in the µc having frequency of 12MHz.



Oscillator :-

A Microcontroller is a S/M of F/F's or synchronous devices. In order to operate this synchronous devices we require a clk sgl which is generated by oscillator, whose freq is from 4MHz - 30MHz

If freq ↑ (high) ⇒ processing speed of uC is high

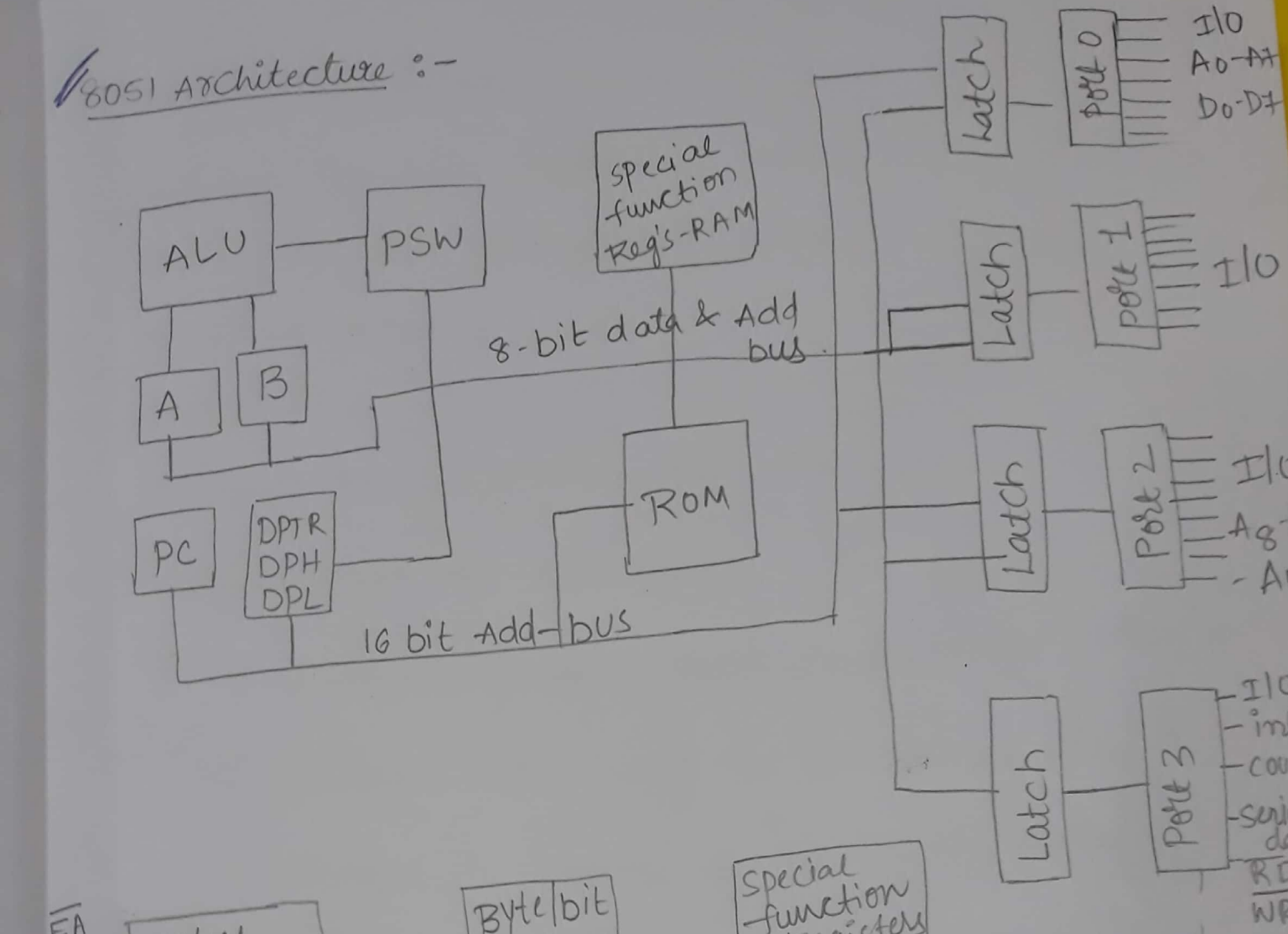
If freq is low ⇒ processing speed of uC is low

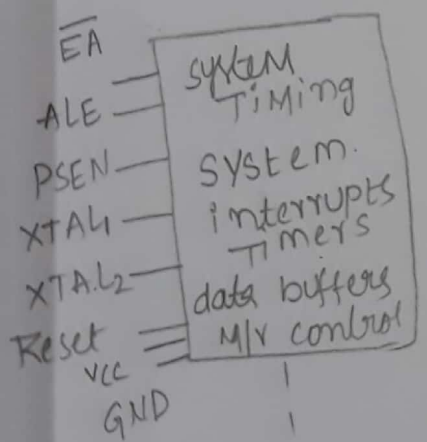
ALU :- (CPU)

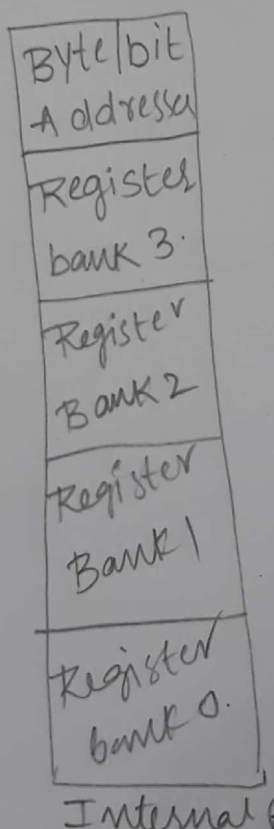
It is 8-bit ALU which performs 8 bit Arithmetic & logical operations.

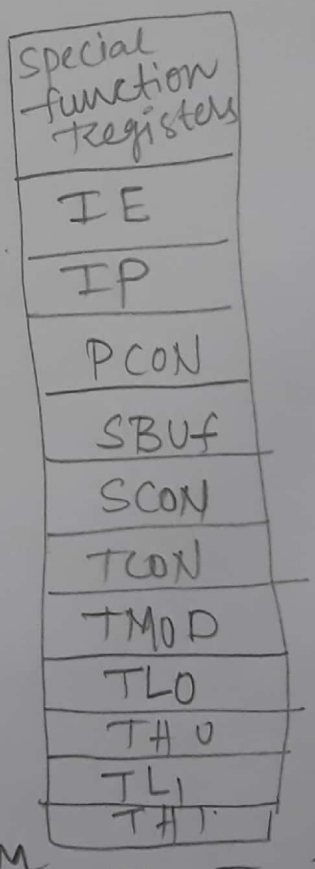
BUS control :-

8051 Architecture :-









The 8051 architecture consists of these specific features

→ Eight bit CPU with registers A and B.

→ 16 bit PC and data pointer (DPTR).

→ Eight bit PSW

→ 8-bit SP

→ Internal ROM or EPROM of 0 - 4KB

→ Internal RAM of 128 bytes.

i) 4 Register banks, each containing 8 Reg's

ii) 16 bytes which may be addressed at bit level

iii) 80 bytes of General-purpose data m/r

→ 32 I/O pins arranged as 4, 8bit ports

→ 2. 16 bit Timer/counters : T_0 & T_1

→ Full duplex serial data T_x^{er} / R_x^{er} : SBUF

→ Control registers : TCON, TMOD, SCON, PCON, IP, IE

→ 2 External & 3 internal interrupt sources

→ Oscillator & clock Ckt.

A and B CPU Reg's

The 8051 contains 34 General purpose registers. Two of these registers are A, B. The other 32 are arranged as part of internal RAM in four banks, $B_0 - B_3$, of Eight registers each named R_0 to R_7.

The Accumulator register is used for many operations, including addition, subtraction, integer Multiplication and division and boolean bit manipulation. The A-reg is also used for all data T/F between 8051 and any external m/r. The B-reg is used with the A-reg for Multiplication and division operation.

program Status word (psw) :-

7	6	5	4	3	2	1	0
CY	AC	F_0	RS_1	RS_0	OV	-	P

RS_1 - Register bank select bit 1

RS_0 - Register bank select bit 0

RS_1	RS_0	
0	0	select Reg-bank 0
0	1	select Reg-bank 1
1	0	select Reg-bank 2
1	1	select Reg-bank 3

CY → It is set when there is a carry while Add^{n} opn or borrow while Subtr^{n} opn, while JUMP, ROTATE and boolean. inst^{n}

AC → used/set for BCD Arithmetic

F_0 → User flag 0.

OV → used in arithmetic inst^{n}.

– → for future use.

P → even parity/odd parity is shown for contents of Reg-A.

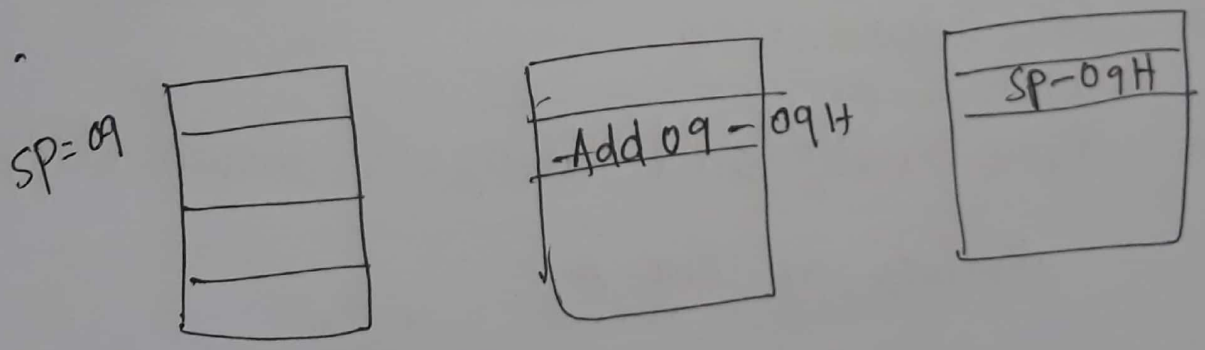
Program counter and data pointer :-

The 8051 contains two 16-bit registers ; the PC and data pointer. These are used to hold the address of a byte in M/r.

Program inst^{n} bytes are fetched from locations in Memory that are addressed by the PC. The DPTR register is made up of two 8-bit registers, named DPH and DPL

8-bit stack pointer :-

It points the data which is most recent entry in stack. This SP is used by the 8051 to hold an address that is top of stack



The address held in SP registes is the location is the location in internal RAM where the last byte of data was stored by a stack operation

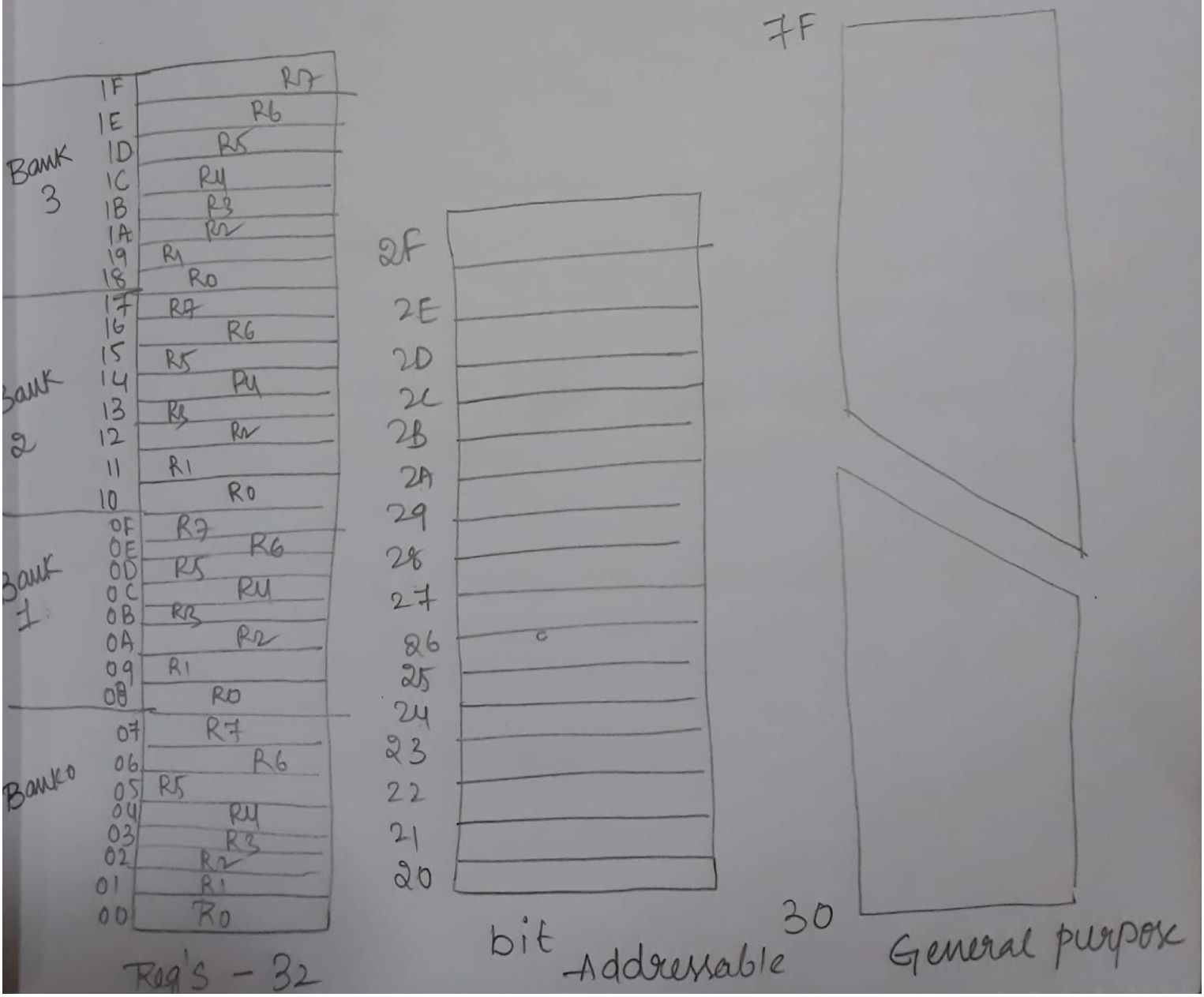
Internal RAM

128-byte of RAM is organized in 3 distinct areas.

i) 32 bytes from 00h to 1Fh that Make Up 32 working registers organized as 4 banks of eight reg's each

2) A bit-addressable area of 16 bytes occupies RAM byte addresses 20h to 2Fh

3) A General purpose RAM area above bit area from 30h to 7Fh. addressable as bytes.



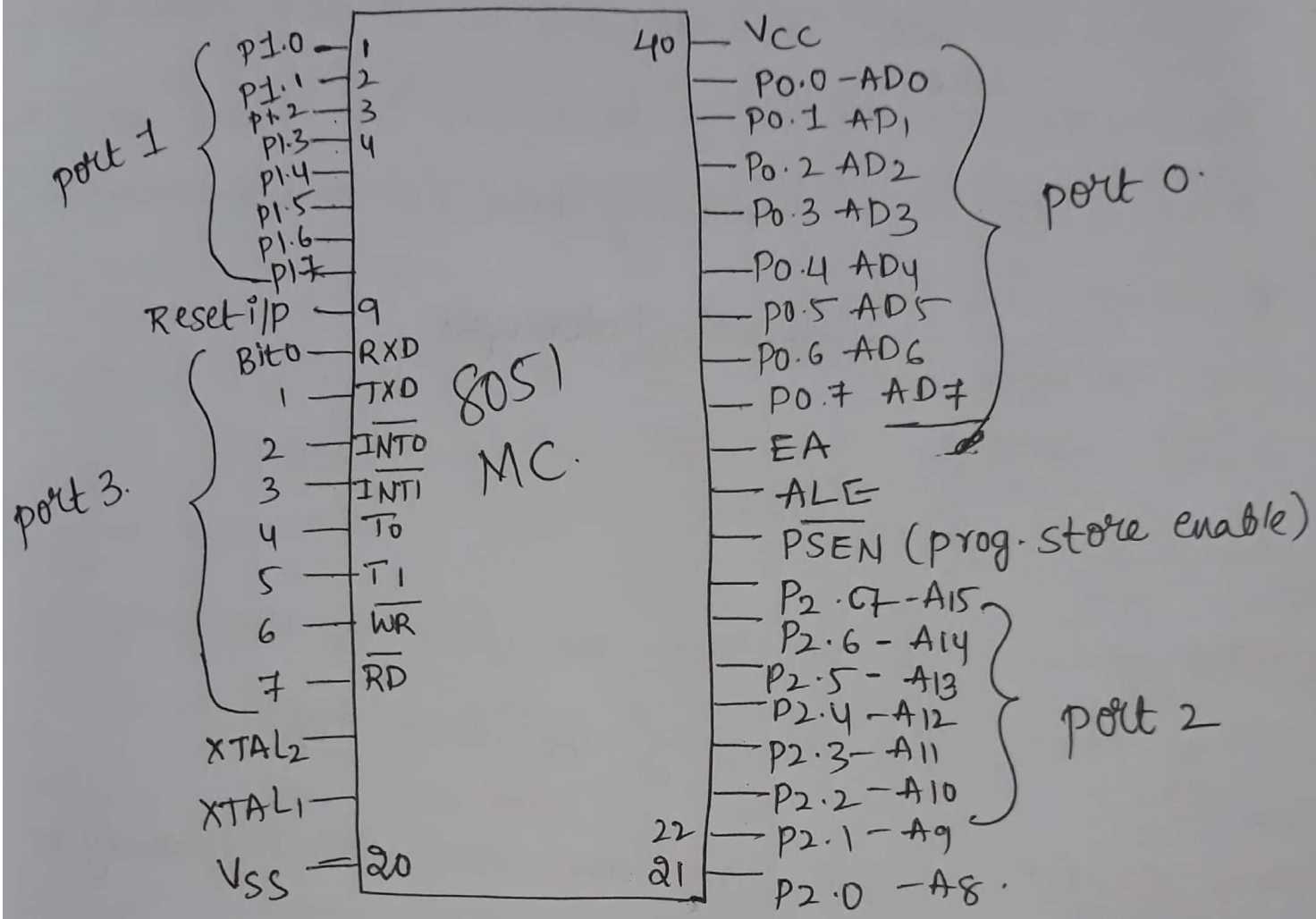
Special function Reg's :-

Name	function
A	Accumulator
B	Arithmetic
DPH	Addressing external M/r
DPL	Addressing external M/r
IE	interrupt enable control
IP	interrupt priority
PCON	power control
SCON	Serial port control
SBUF	Serial port data buffer

- All are 8-bit Reg's

  2 Timers are 16-bit Reg's

Pin diagram :-



EA - External enable ACCESS

PSEN - prog. store enable

Pins 1-8 :- These pins are known as p-①. This port does'nt serve any other functions. It is internally pulled up, bi-directional I/o port.

Pin 9:- It is used to reset the µC to its initial values i.e. Reset pin

Pin 10 - 17 :- These are port-③ pins. It also serves func^{n} like

P3.0 → RxD i.e. Receiving data pin in serial Comm^{n}

P3.1 → TXD i.e. Transmitting data in serial comm^{n}

P3.2 → $\overline{INT0}$
P3.3 → $\overline{INT1}$ } External interrupts

P3.4 → T_0
P3.5 → T_1 } Timers

P3.6 → $\overline{WR}$ i.e. writting the data by µC

P3.7 → $\overline{RD}$ i.e. Reading the data by µC

Pins 18-19 :- These pins are used for interfacing an External crystal to get S/M clock

Pin 20 :- Vss for Ground references

Pins 21-28 :- These are port-② pins. Higher order address bus signals are also multiplexed using this port.

Pin 29 :- PSEN - prog. store enable

This is used to read a signal from external prog. M/r

pin-30 :- EA - External Access.

It is used to enable / disable the external m/r interfacing

pin-31 :- ALE - Address latch Enable

It is used to deMultiplex the address - data signal of port-0

pins 32-39 :- port-0

Lower order Address & data bus signals are Multiplexed using this port

pin 40 :- Vcc → +5V

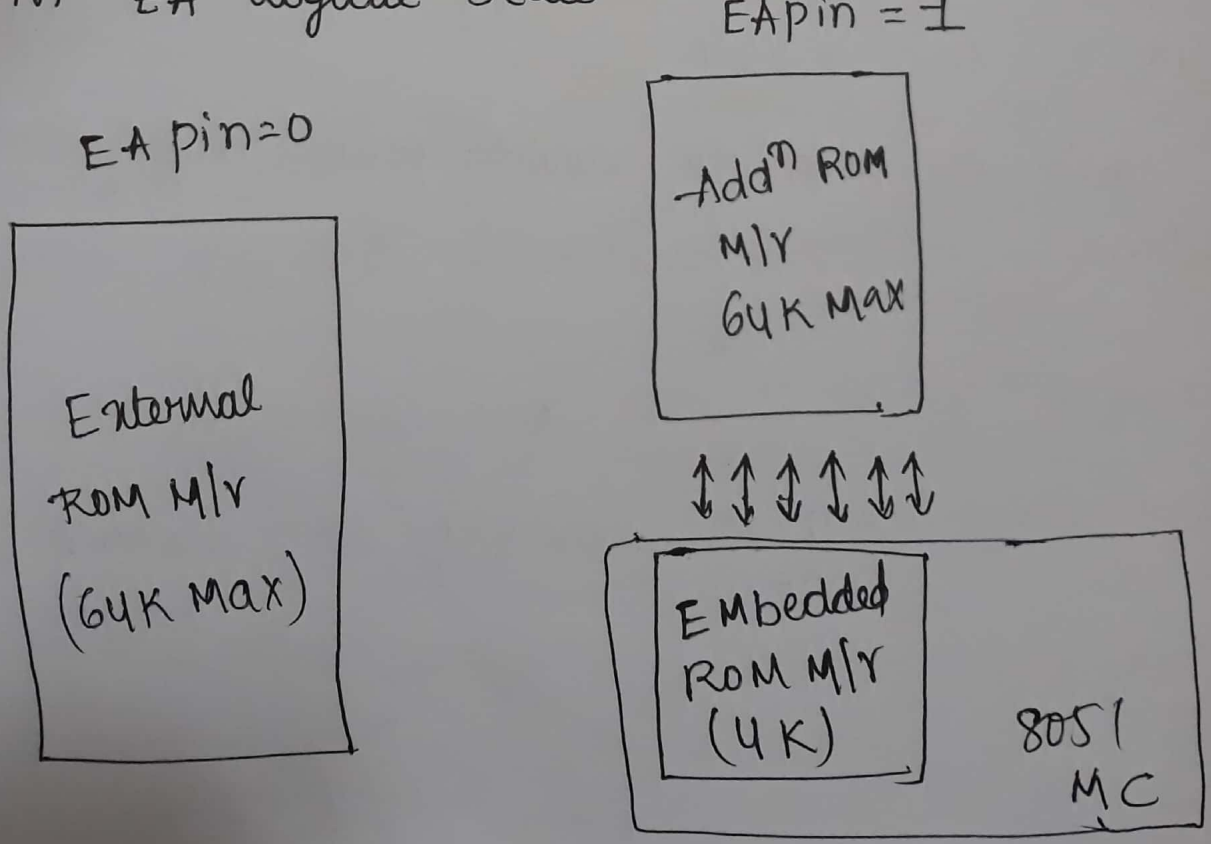
This is used to provide power supply to ckt.

8051 M/r organization :-

The 8051 MC M/r is divided into prog. M/r and data M/r. program Memory (ROM) is used for permanent saving program being executed, while data M/r (RAm) is used for temporarily storing and keeping intermediate results and variables

program M/r (ROM)

Prog. M/r is used for permanent saving prog. The 8051 executes programs stored in prog. M/r only 8051 M/r organization allows external prog. M/r to be added. This Addn of external prog M/r depends on pin EA logical state.



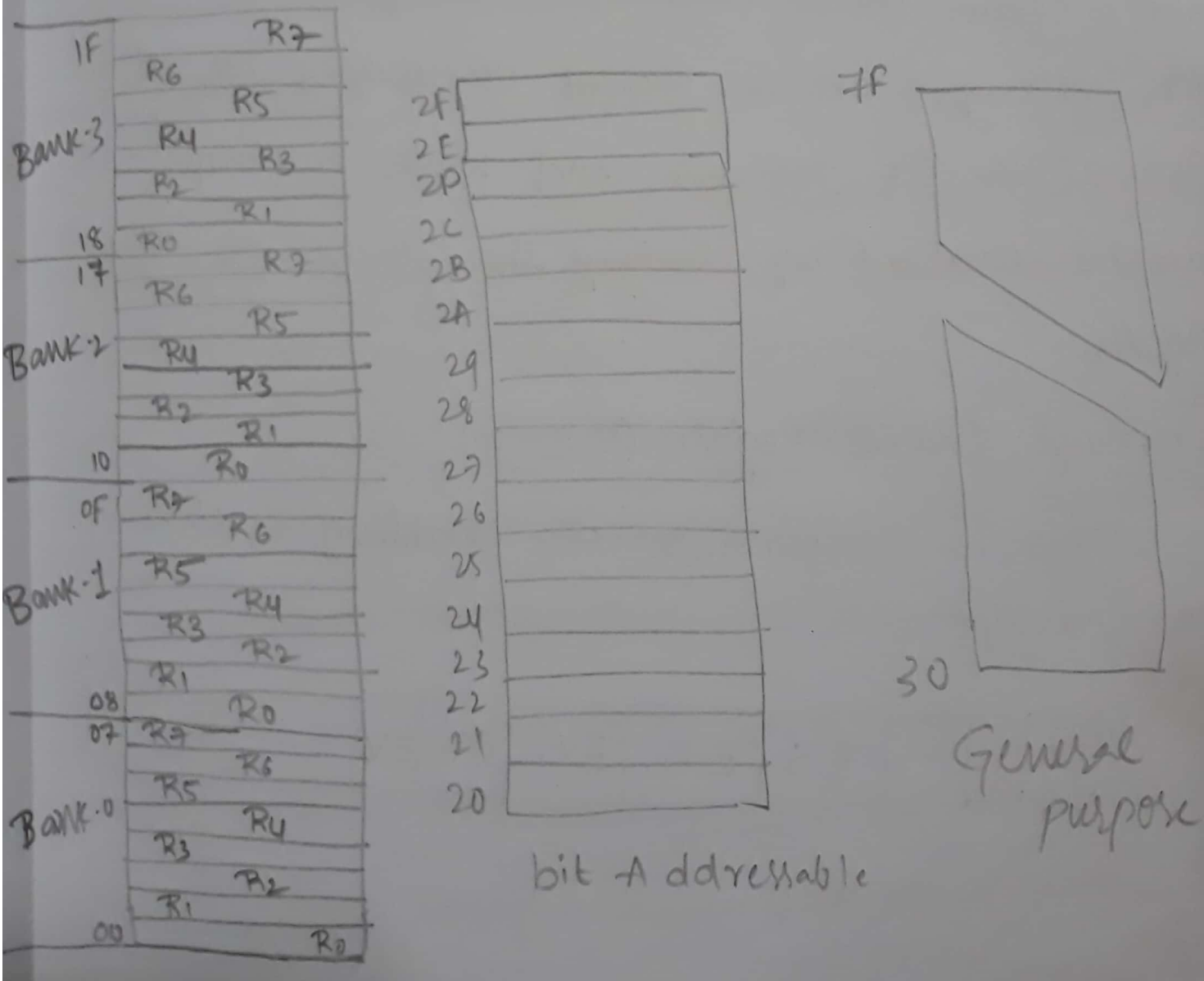
Internal data M|r (RAM)

128 byte of RAM is organized in 3 distinct areas

i) 32 bytes from 00h to 1Fh that Make up 32 working Reg's organized as 4 banks containing eight registers of each bank

ii) A-bit addressable area of 16 bytes occupies RAM byte Addresses 20h to 2Fh

iii) A General purpose RAM area above bit area from 30h to 7Fh addressable as bytes



Interrupts of 8051 :- It provides 5 vectored interrupts

i) $\overline{INT0}$ - External interrupt-0

2) TF0 - Timer-0 overflow interrupt

3) TF_1 - Timer-1 overflow interrupt

4) $\overline{INT1}$ - External interrupt-1

5) RI/TI - serial port interrupt

$\overline{INT0}$ & $\overline{INT1}$ are external interrupts whereas Timer & serial port interrupts are generated internally Each interrupt can be enabled or disabled by setting bits of the IE register and whole interrupt s/m can be disabled by clearing the EA bit of same register.

Interrupt Enable Register (IE)

This is responsible for enabling & disabling the interrupt

EA	-	-	ES	ET_1	EX_1	ET_0	EX_0

EA — when EA = 0 it disables all interrupts
when EA = 1 enables the interrupts individually

ES — Enables / disables serial port interrupt

ET_1 — Enables / disables. timer 1 overflow interrupt

EX_1 — Enables / disables External interrupt - 1

ETO — Enables / disables timer 0 overflow interrupt

EXO — Enables / disables External interrupt - 0

Interrupt priority Reg :- (IP).

we can change the priority levels of the interrupts by changing the corresponding bits in IP register as shown in following figure

i) If two interrupts of different priority levels are received simultaneously the request of higher priority level is served.

ii) If the requests of the same priority levels are received simultaneously, then the internal polling sequence determines which request is to be serviced.

c

-	-	-	PS	PT_1	PX_1	PT_0	PX0

PX0 – It defines the external interrupt 0 priority

PT0 – It defines the TIMER-0 interrupt priority

PX1 – It defines the External interrupt 1 priority

PT_1 – It defines the TIMER-1 interrupt priority

PS – It defines serial port interrupt priority

clear → low priority ; set → high priority

TCON Reg :-

It specifies the type of External interrupt to the Micro-controller

Instruction set :-

The different types of Instⁿ under 8051 are

i) Arithmetic instⁿ

ii) Logical instⁿ

iii) data T/F instⁿ

iv) boolean instⁿ

v) branching instⁿ

Arithmetic instruction's :-

i) ADD - Addⁿ operation

ADD A, B

A ← A + B

ii) ADDC - Add with carry

ADDC A, B

A ← A + B + CY

iii) DA - decimal adjust the accumulator

ADD A, B

A ← A + B.

DA A

Adjust the binary result of adding two BCD no.'s in the A register to BCD and adjust carry flag

iv) SUBB - Subtract with borrow

Subtract the source byte and the carry from A and put the result in A and adjust the carry and overflow flags.

SUBB A, source

v) MUL AB

multiply the bytes in A and B ; High order byte of result is stored in B. the lower order byte is stored in A ; Set the OV flag to 1 if result exceeds OF bits (15 bits)

vi) INC Source.

Add a 1 to Source

INC Source

Source ← source + 1

VII) DEC Source

Subtract a 1 from source

DEC Source ⇒ Source ← source - 1

VIII) DIV AB

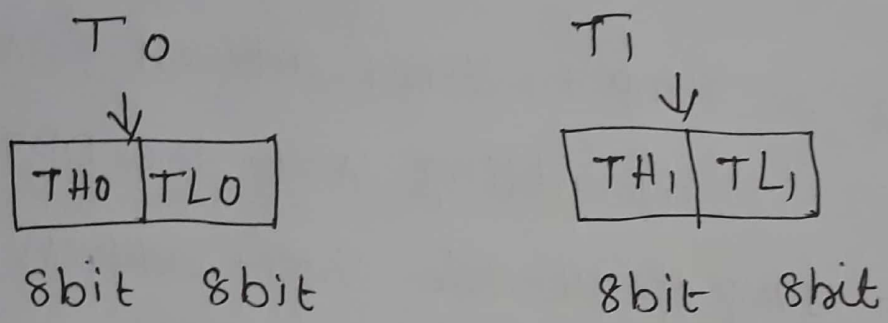
Divide the byte in A by the byte B ;

Timer / counters :-

Timer – It is used to measure time interval or No. of clock pulses

counter – stores No. of times a particular event or process occured.

Timers – T_0 & T_1



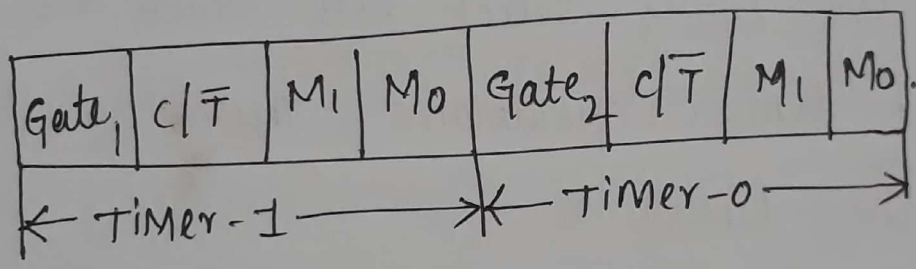
Timer 0 & Timer-1 both are 16 bit registers in which lower byte is stored in TL and higher byte is stored in TH. Timer can be used as a counter as well as for timing operation that depends on the source of clock pulses to counters

counters and Timers in 8051 MC contain two special function registers i.e. TMOD & TCON

TMOD – Timer mode Register

TCON – Timer control Register

TMOD:- TMOD is an 8-bit register used for selecting timer or counter and Mode of timers. Lower 4-bits are used for control operation of timer 0 / counter 0 and remaining 4-bits are used for control operation of timer 1 / counter-1



Gate:- OR Gate

Enables & disables Timer 1 while INT(0,1) is high.

then $Gate_1 \rightarrow 1$:- To enable the interrupt to start / stop the times i.e. when $Gate_1$ bit is set the timer will only run when INT1 is high

When $Gate_1 = 0$ the timer will run regardless of state of INT1

C/$\bar{T}$:- counter / timer op^n

if C/$\bar{T}$ bit is 1 then it is acting as counter mode

if C/$\bar{T}$ bit is 0 then it is acting as Timer mode.

Mode select bits :- M_1 & M_0.

M_1 & M_0 are Mode select bits which are used to select the timer operations. There are 4 Modes to operate the timers

M_1	M_0	Mode
0	0	Mode-0
0	1	Mode-1
1	0	Mode-2
1	1	Mode-3.

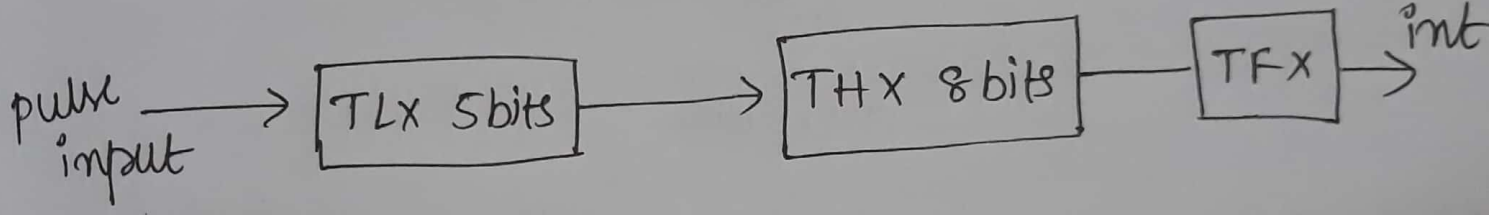
Mode-0 :- This is 13 bit Mode

$2^{\text{No. of bits}}$ – pulses required to complete timer Operation

$= 2^{13}$.

= 8192 pulses are required to complete timer operation.

TMOD register results in Using the THX register as an 8-bit counter & TLX as a 5 bit counter



Timer Mode-0

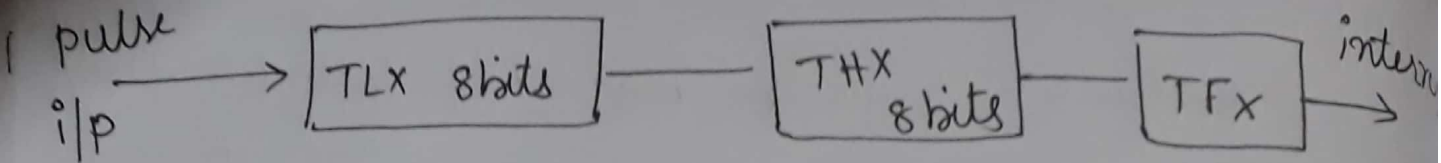
13 bit Timer / counter

Mode-1 :- This is a 16 bit Mode.

pulses required to complete Mode-1 opn = 2^{16}.

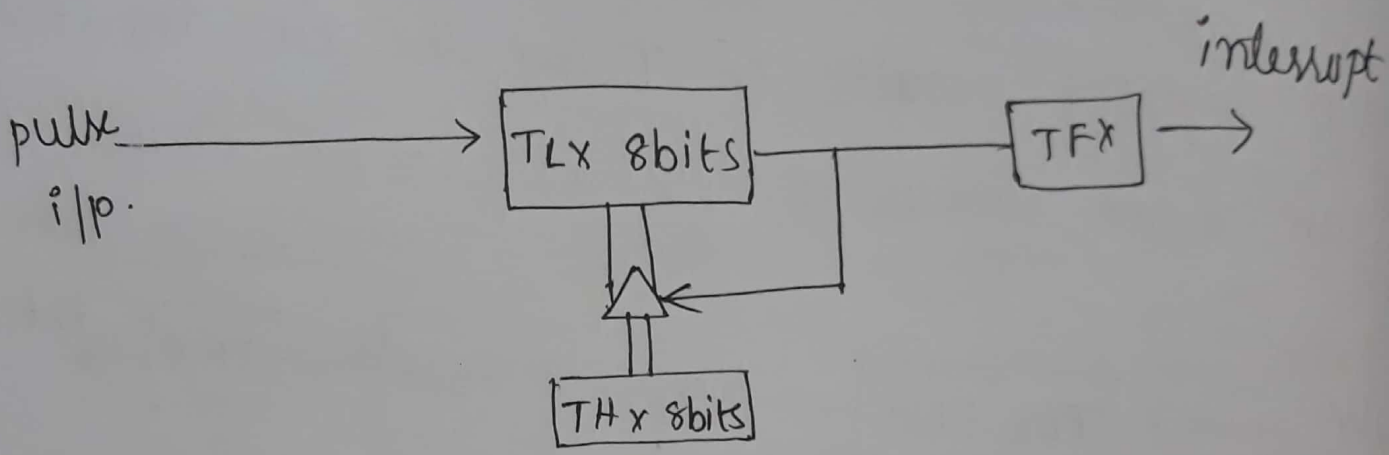
= 65,535

TMOD register results in using the THX register as an 8-bit counter & TLX as a 8 bit counter

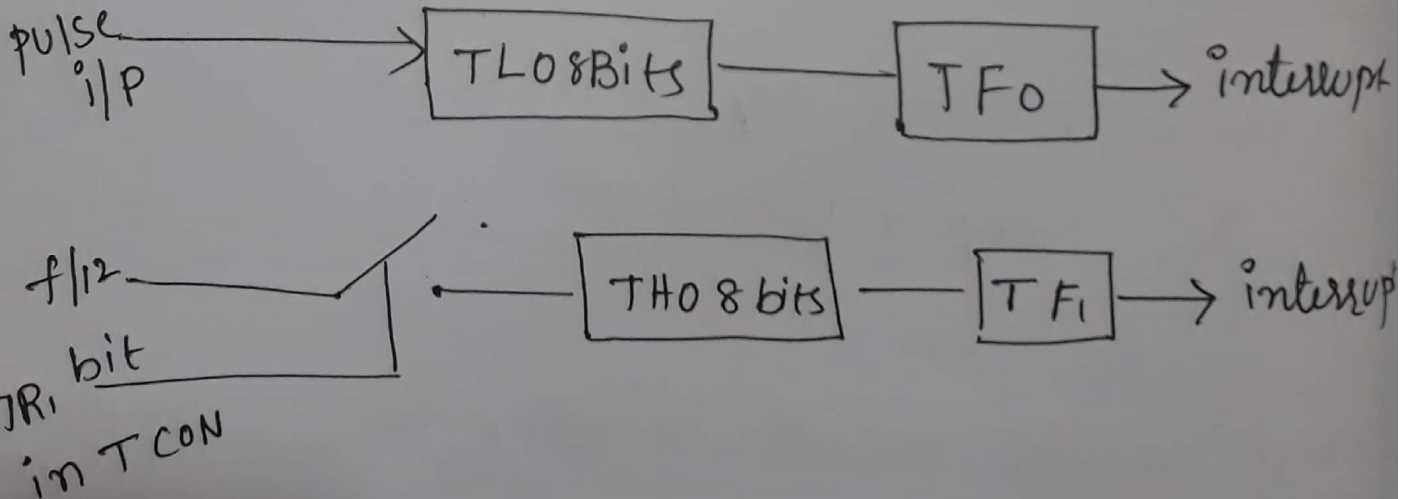


Mode-2 :- 8-bit auto reload Mode

No. of pulses required to complete Timer opn $= 2^8 = 256$.

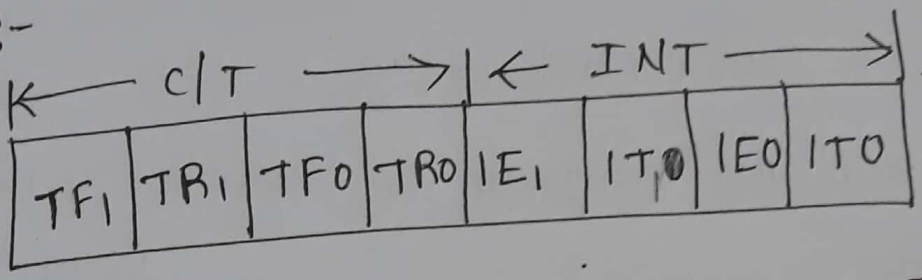


Mode-3 :- This Mode is a split-timer mode which Means the loading values in T_0 and automatically starts the T_1.



Timers 0 & 1 may be programmed to be in Mode 0, 1 & 2 independently. but T_0 & T_1 do not operate independently in Mode-3

TCON :-



TCON is another register used to control operations of counter and timers in MC. It is an 8-bit register where upper four bits are responsible for timers & counters and lower bits are responsible for interrupts

TF1 :- Timer-1 flag bit i.e. Timer 1 overflow flag

TF1 is set when timer-1 rolls all ones to zero

TF1 is cleared when ISR is executed located at address 001Bh.

TR1 :- Timer-1 Run control bit

TR1 → 1 to enable timer to count

TR1 → 0 to halt timer-1

TF_0 — Timer-0 overflow flag

TF_0 is set when timer rolls from all ones to zero

TF_0 is cleared when ISR is executed located at address 000Bh

TR_0 — Timer-0 Run control flag

TR_0 is set by program to enable timer to count

TR_0 is cleared to 0 by program to halt timer

IE_1 — External interrupt-1 edge flag

Pin 3.3 — $\overline{INT1}$

Set to 1 when a high to low edge signal is received on port 3.

cleared when processor vectors to interrupt service routine located to program address 0013h.

IT_1 — External Interrupt-1 signal flag

Set to 1 by prog. to enable $\overline{INT1}$

Cleared when $\overline{INT1}$ is disabled by program.

IE0 :- Ext. int. 0 edge flag.

Set to 1 when a high to low edge sgl is received on P-③.

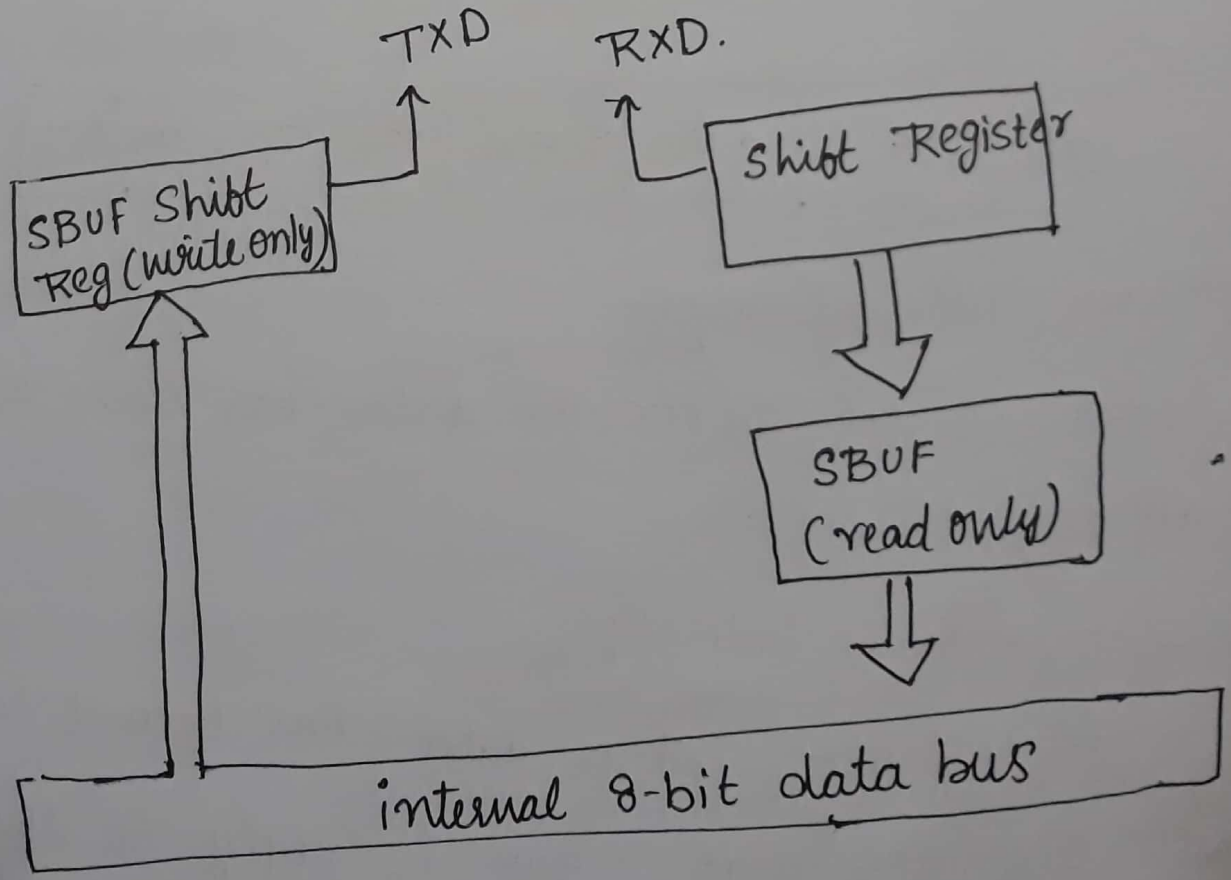
Cleared when processor vectors to ISR located to prog. Add. 0003h.

IT0 :- External interrupt 0 signal flag.

Set to 1 by prog. to enable INT0.

Cleared when $\overline{INT0}$ is disabled by prog.

Serial COMMUNICATION :-



The serial port of 8051 is full duplex i.e., it can transmit and receive simultaneously. The register SBUF is used to hold the data...

i) write-only – used to hold data to be Tx^ed out of 8051 via TxD

ii) Read only – used to holds the received data from external sources via RxD.

Hence there are 2 SBUF registers to hold data

SCON :- Serial port control register

It controls serial data comm^n.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

RI – Receive interrupt

RI is set after the data has been received in all Modes.

TI – Transmit interrupt

TI is set when data has been transmitted which signifies that SBUF is empty so that another byte can be sent

8051 MC communicate with another peripheral device through RXD & TXD pin of port 3. controller have 4 Mode of Serial communication. i.e.

SM_0	SM_1	Mode	
0	0	Mode 0	
0	1	serial Mode 1	– 8 bit data, 1 start bit & 1 stop bit
1	0	serial Mode 2	
1	1	serial Mode 3	

Mode-0 :-

→ In this Mode the serial port works like a shift register and data transmission works with a clock freq of fosc / 12.

→ Serial data is received and Transmitted through RXD 8 bits are Tx^ed / Rx^ed at a time.

Baud rate :- The rate at which serial port is capable of Transferring a Max of 9600. bits per Sec. also called shift freq.

Baud rate = 1/12 (fosc).

Mode - 1:-

In Mode-1 the serial port functions as a Standard UART Mode.

10 bits are transmitted through TXD or Received through RXD.

10 bits $\Rightarrow$ 1 start bit + 8 bit data + 1 stop data

Start bit is usually 0 & stop bit is usually 1

LSB is sent first followed by MSB for 8 bit data

Baud rate is variable

Mode-2 :-

In this Mode 11 bits are transmitted through TXD or received through RXD

11 bits $\rightarrow$ 1 start bit + 8 bit data + 9th (TB^8/RB^8) bit and 1 stop bit

Baud rate $= 1/32 (f_{osc})$

(or)

$\frac{1}{64} (f_{osc})$

$$f_{baud} = \frac{2^{SMOD}}{64} \times f_{osc}$$

Mode-3 :- Multiprocessor Mode

In this Mode 11 bits are Tx^ed / Rx^ed

The various bits are

a start bit (usually 0)

8 data bits (LSB first)

a programmable 9th bit

a stop bit

Mode-3 is same as Mode-2 except the fact is baud rate is variable

SM_2 :- It enables Multiprocessing capabilities.

TB8 :- Transfer bit 8. used for serial Mode 2 & 3. Not generally used so set it always to 0.

RB8 :- Receive bit 8. Used for serial Mode 2&3 Not generally used so set it always to 0

REN :- Receive enable.

REN is high it allows 8051 to receive data from RxD pin.

PCON — Power Mode control

SMOD	-	-	-	GF1	GF0	PD	IDL

GF1 — General purpose user flag bit - 1
set/cleared by program

GF0 — General purpose user flag bit - 0
set/cleared by program

PD — power down bit
set to 1 by program to enter power down configuration

IDL — idle Mode bit
set to 1 by program to enter idle Mode

SMOD — Serial baud rate Modify bit
set to 1 by program to double baud rate using timer-1 for Modes 1, 2, & 3
cleared to 0 by program to use timer-1 baud rate

UNIT-4 :- AVR Microcontrollers

* AVR is the name of MC series that "Atmel" produces
* AVR stands for Alf and Vegards RISC controller
* This AVR MC are "RISC MC
  RISC → Reduced instrⁿ set computers
* That means RISC MC have very few instⁿˢ and instⁿ are register - based
* AVR have faster execution than 8051

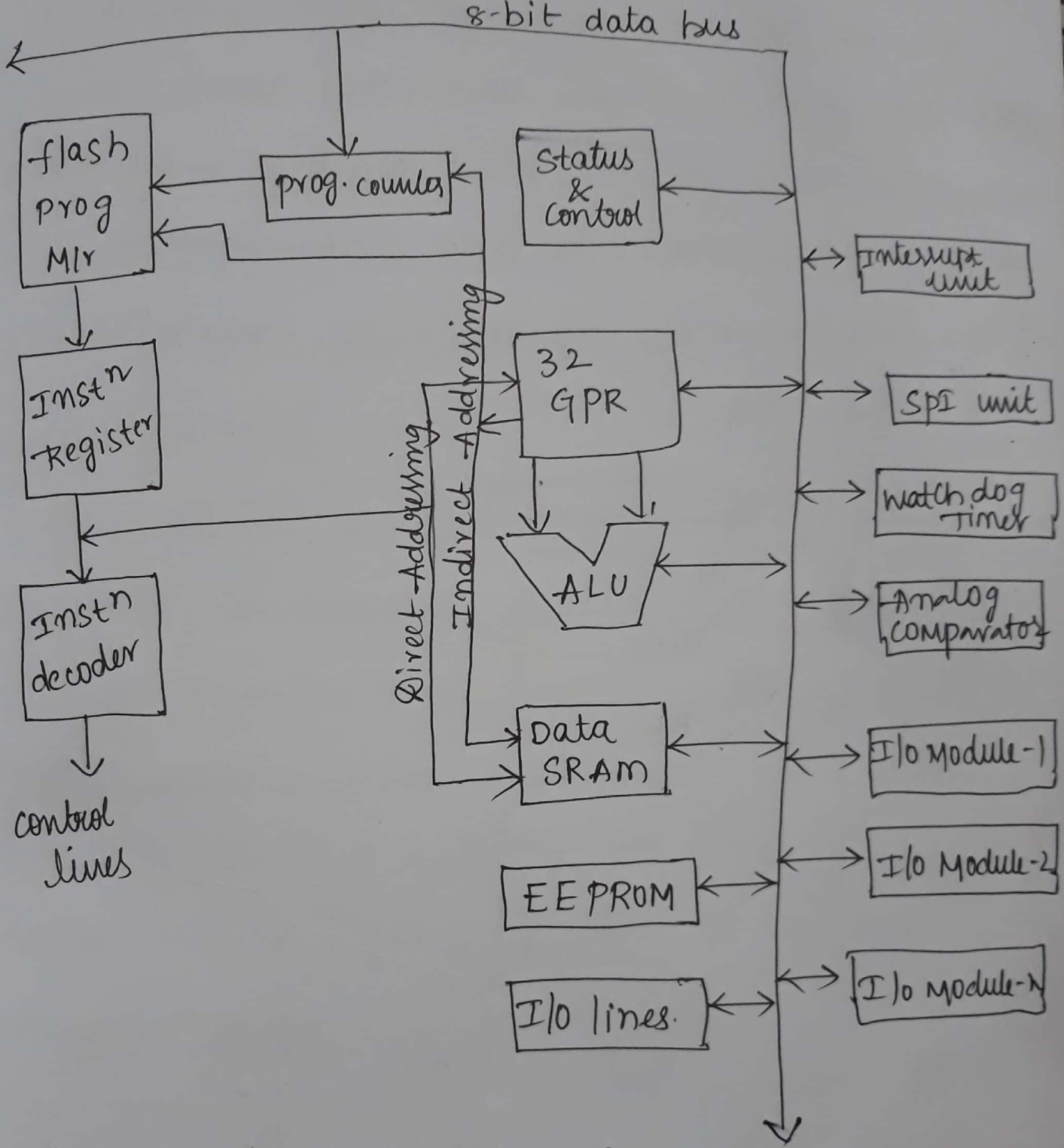
Features of AVR RISC MC :-

* It is a 8-bit RISC architecture.
* It provides RISC architecture with fixed length instⁿˢ
* RISC instⁿ set is combined with 32 General purpose working registers. Each register is connected to ALU, allowing two independent registers to be accessed in a single instⁿ.
* Most of the instⁿˢ are accessed in only one cycle
* It provides pipelining to speed the execution

* It contains programmable I/O, ADC, EEPROM, Timers, UART etc..
* It provides on chip program & data M/r
* It is available in 8 pin to 64 pin package size to suit wide variety of applications
* It provides 12 times performance speeding over conventional CISC controllers
* It operates at voltage from 2.7 V - 6.0 V
* It contains 32 general purpose registers
* It contains 8-bit Timers
* It has 3 External interrupts
* It has 8-bit status register (flag)
* 16KB flash M/r, 2KB RAM, 128 bytes of EEPROM
* It is a 40-pin MC
* 64 I/O Registers

Architecture of AVR Microcontroller :-

AT90S2313



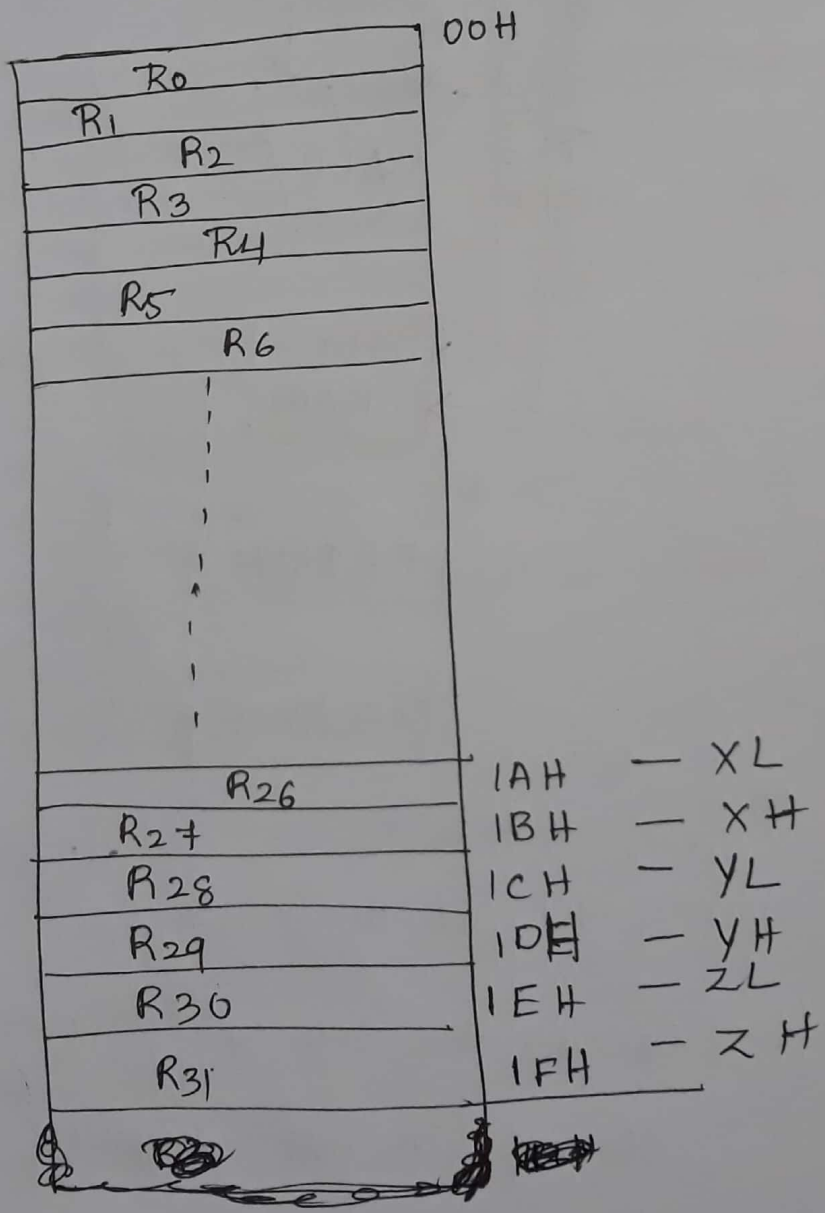
Instⁿ Reg :- Stores the set of instructions.

Instⁿ decoder :- Decodes the set of instructions. that are stored in instⁿ Registes

General purpose registers :-

It has 32x8 bit general purpose working registers with a single clock cycle access time six of 32 reg's can be used as three 16-bit indirect address register pointers for data space addressing

These three 16 bit registers X (XL, XH), Y (YL, YH) and Z (ZL, ZH)



ALU :-

ALU supports arithmetic opn b/w Reg's or b/w a constant and a register. With in a single clock cycle ALU opns b/w registers in the registers file are executed

The ALU opns are divided into three categories Arithmetic opns, Logical opns and bit opns.

Programmable flash M/r :-

It contains 2K bytes onchip in s/m programable flash M/r for program storage

It is organised as 1K x 16

Program counter :-

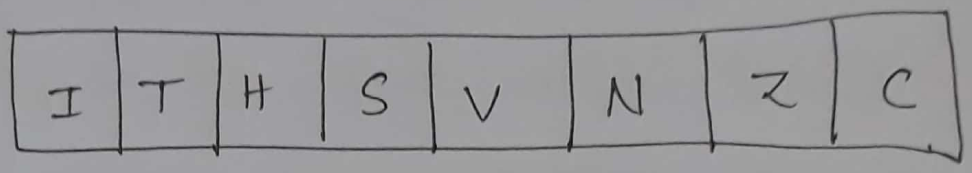
It is 10-bit wide, thus addressing 1024 program Memory addresses (1024 prog. M/r addresses).

EEPROM data M/r :-

It contains 128 bytes of EEPROM. It is organised as a separate data space in which single bytes can be read/written

Status Register :-

It contains 8 flag bits that indicates the current state of µC.



I (Global Interrupt Enable)

I = 1 ; All interrupts are enabled

I = 0 : All interrupts are disabled.

T (Bit Copy Storage)

The bit copy instn BLD (bit load) and BST (bit store) use the T-bit as source & destination for the operated bit

bit from.

register → T (BST)

T → Register (BLD)

H (Half carry flag)

It indicates a half-carry in some arithm operations

sign bit (S) :- It is an exclusive OR of sign and overflow bits.

Two's complement overflow flag (V) :-

If it is enable it supports the two's complement of arithmetics

Negative flag (N) :-

It indicate -ve result after different Arithmetic & logical opns

zero flag (Z) :-

It indicates a zero after diff. Arithmetic & logical opns

carry flag (C) :-

This flag is enable if any arithmetic & logical opns get a carry at the result

SRAM :- static Random Access m/r

It is used to supply data through the Run-time. This is volatile m/r which is pre-arranged in 8-bit registers

I/o lines :- 23 I/o lines are there to connect or interface with External device

Analog I/o Modules:-

These are used to i/p or o/p analog info from or to the exterior world.

This will contain Analog comparator & analog to digital converters

Interrupt unit :-

It is responsible for the interrupts in AVR MC's. It contains 3 External interrupts

watch dog Timer :-

It checks the error-free opn of MC. In case of a correctly running program the watch dog timer must be reset by program

It is enabled by WDE bit (watch dog enable) otherwise it stops the check of program flow.

Serial peripheral interface (SPI)

It allows high speed synchronous data T/f b/w AVR and peripheral devices or b/w several AVR's

I/O ports

various AVR are differently equipped with I/O ports At minimum, an AVR is equipped with two digital I/O ports. i.e. port B, port D

Serial commⁿ :-

It can be two types - Asynchronous commⁿ. and synchronous. commⁿ.

Asynchronous commⁿ :- Receiver has no prior intimation about the arrival of data bits/ packets because of this receiver needs some other informⁿ like baud rate, packet size, no. of stop bits etc..

synchronous commⁿ :- besides single data line it has one / more other lines for synchronization The extra lines carry clock or any Informⁿ

universal Asynchronous Rxer & Txer (UART) :-

UART is a popular example of serial asynchronous comm^{n}. In UART the arrival packet is indicated by start bit, appended by beginning of every packet, and the end of packet is indicated by stop bit. UART works in full duplex mode

The AVR UART transmits one characters pack at a time. Each of these packet contains one start bit followed by 5-9 data bits followed by parity bits finally followed by parity bits and stop bits

After receiving this the Rxer prepares it self to handle the upcoming data bits. As receiver knows the baud rate (bit rate) it starts sampling the line after every bit delay

(bit duration - inversion of bit rate)

↓

Amount of time the line represents one bit

After the completion of reception of total data bits receiver expects one parity (even/odd).

The UART requires Access to some reg's.

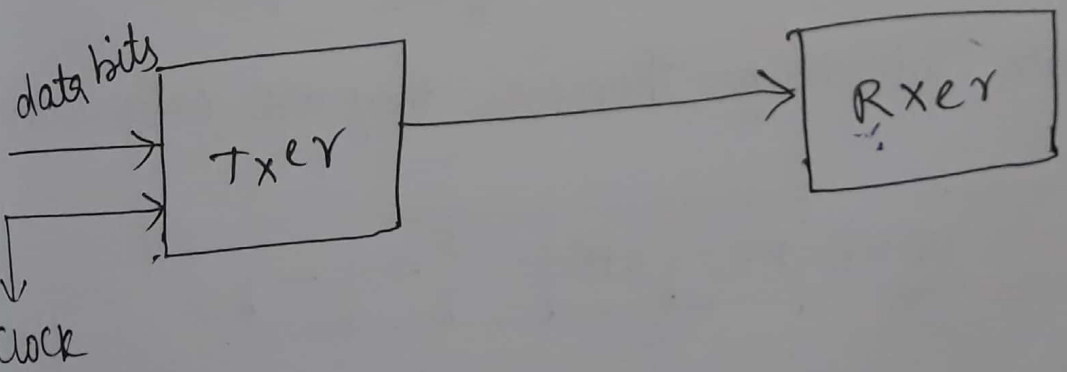
→ USART Baud Rate Register ⇒ UBRRH & UBRRL

→ USART control & status Registes - A } UCSRA

→ USART control & status Reg - B } UCSRB.

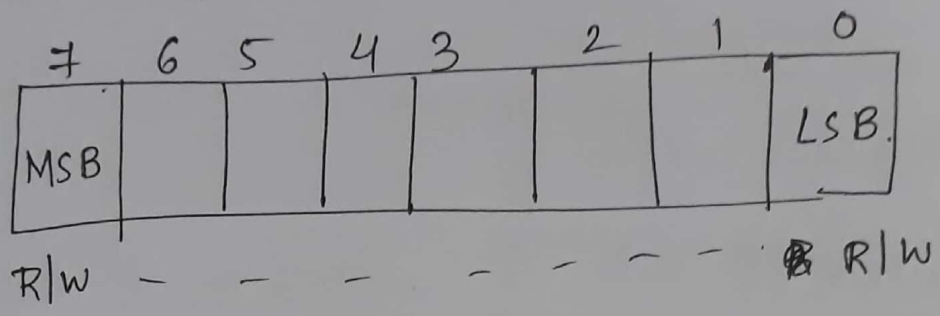
→ USART control & status Reg - C ⇒ UCSRC

→ USART Data bubbes Reg → UDA.



UART data register :- (UDR)

It is physical interface b/w data bus and Tx^{er} / Rx^{cr} reg. shift reg's.



This is implemented twice under the same I/O address.

When writing to the UDR, it acts as a UART transmit data Register, when reading from UDR It acts as UART receive data Register

UART status Register :-(USR)

The bits in UART status register represent result of data interchange. Therefore they are only read

RXC	TXC	UDRE	FE	OR	-	-	-

RxC :- Receive Complete

It is set when the entire Characters in the

receive shift register is Txed to UDR.

TXC :- Transmit complete

It is set when the entire character in the Transmit shift register has been sent to UDR

UDRE :- UART data register EMPTY.

It is set when a character written to the UDR is transferred to the transMit shift register

FE :- framing error.

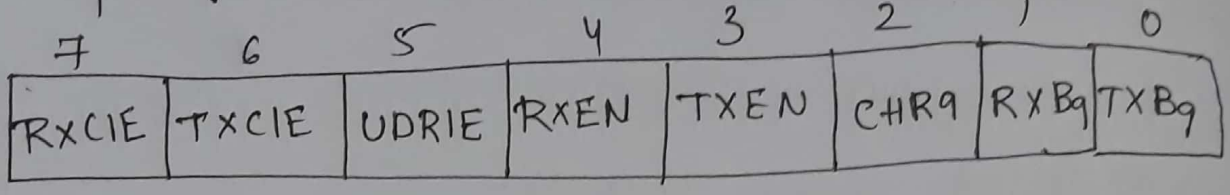
It is set when the stop bit of an incoming character is zero (framing error cond^n).

OR :- over run bit

The unread character in the UDR will be over-written if it is set.

UART control Register (UCR) :-

The bits in the UART control Register enable different modes of UART and help to handle the ninth data bit. This can be used as extra stop bit or a parity bit



RXCIE & TXCIE :- Rx/Tx complete interrupt Enable

If it is set, Receive / Transmit complete interrupt routine will be Executed

It Enables the global interrupts Execution.

UDRIE :- UART data register EMPTY interrupt Enable

If it is set, UART data register EMPTY interrupt routine will be executed

RXEN :- Receiver Enable

When it is set to 1 it enables the UART Rx

TXEN :- TransMitter Enable

When it is set to 1 it enables the UART Tx

CHR9 :- 9 bit characters bit

If it is set it indicates the Txed & Rxed characters are 9-bits long.

RXB9 & TXB9 :-

The NINth bit is read & written by using the RXB9 & TXB9 in the UCR.