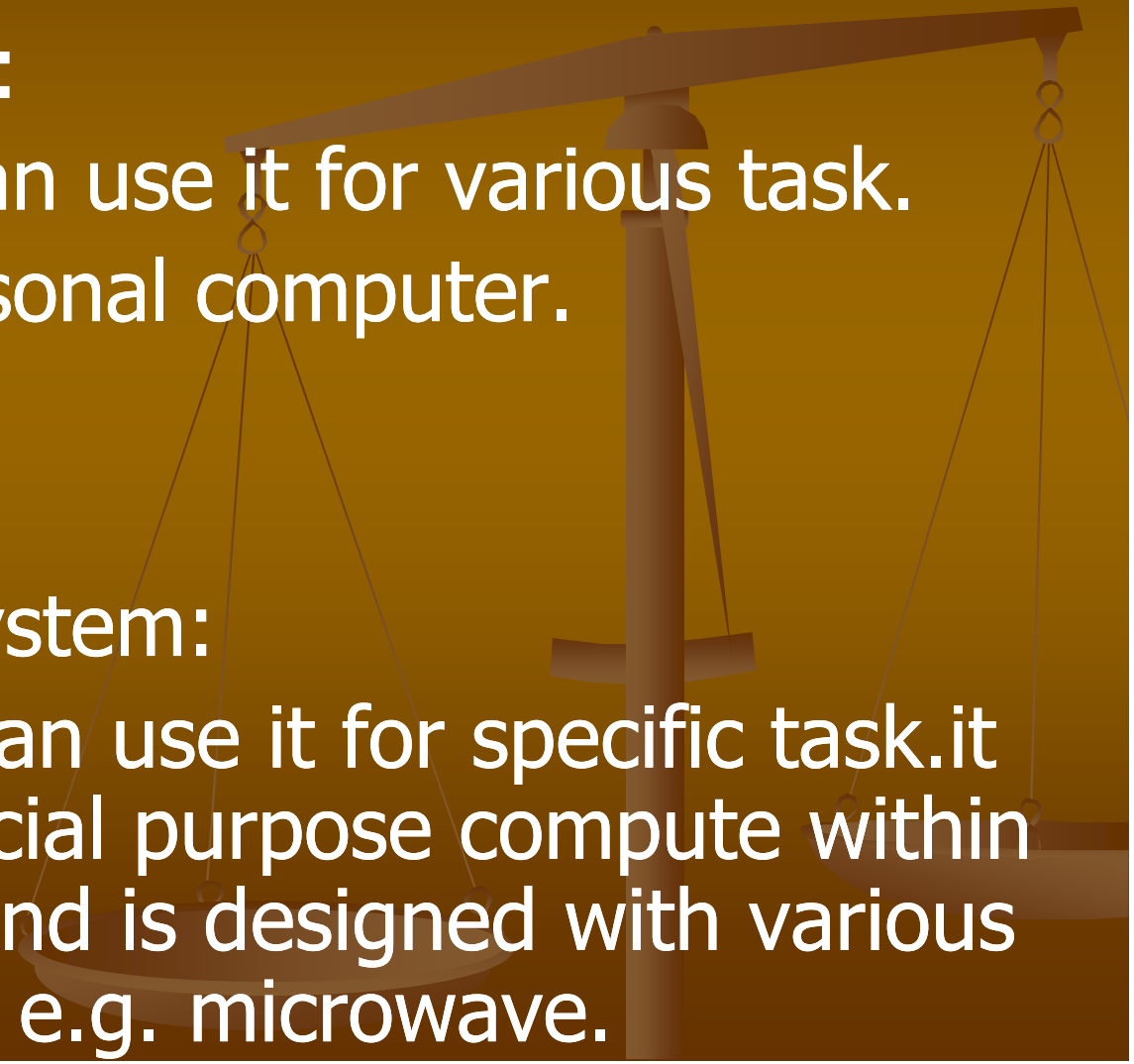
Presentation
On
Summer Training Project Report
titled as
‘EMBEDDED SYSTEMS/8051 MICROCONTROLLER’

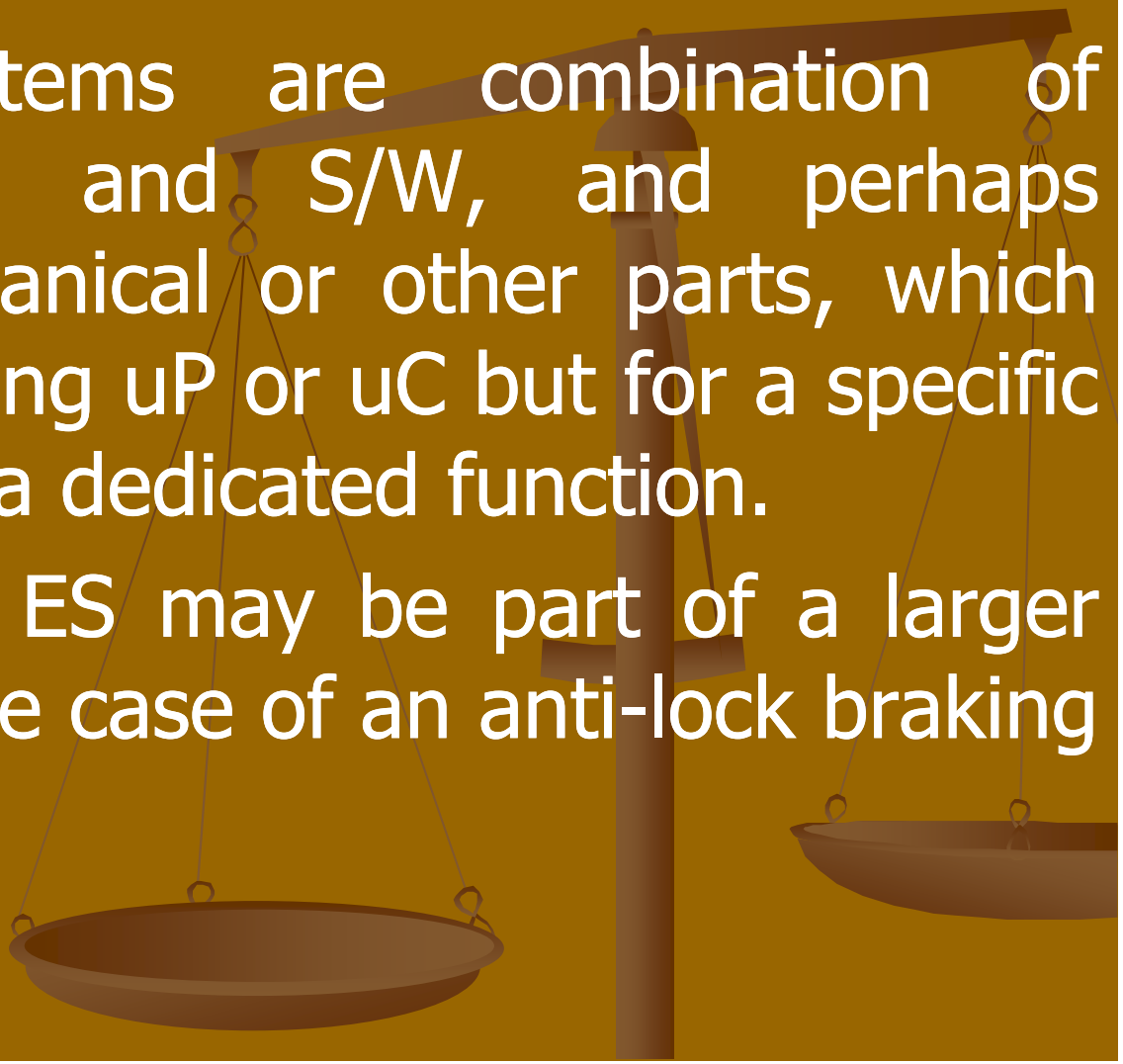
Different types of systems:



1. Open system:
   where you can use it for various task.
   For e.g.. personal computer.

2. Embedded system:
   where you can use it for specific task.it contains special purpose compute within the device, and is designed with various funtions .For e.g. microwave.

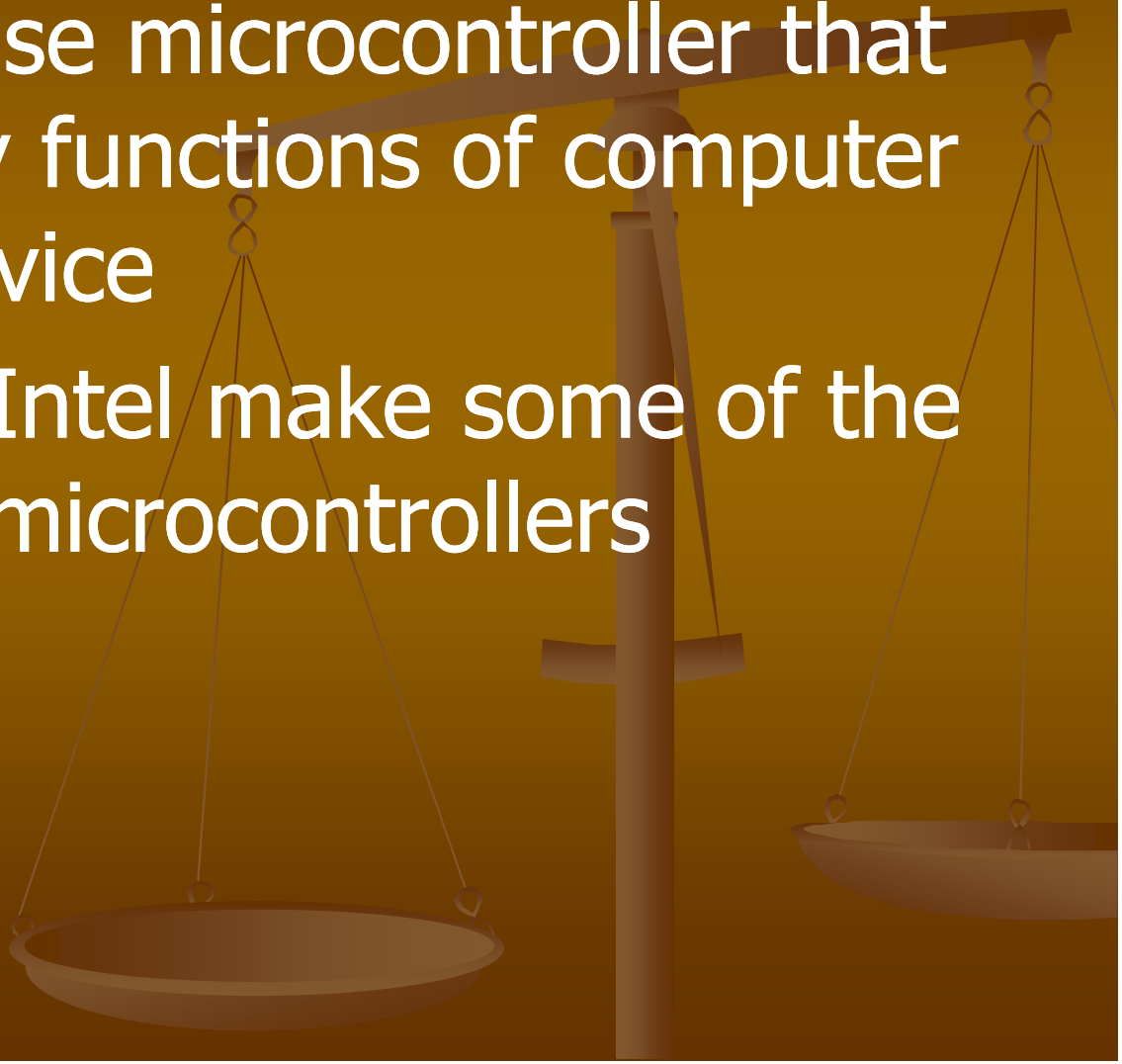
EMBEDDED SYSTEMS



- Embedded systems are combination of computer H/W and S/W, and perhaps additional mechanical or other parts, which are designed using uP or uC but for a specific task to perform a dedicated function.
- In some cases, ES may be part of a larger system , as is the case of an anti-lock braking system in a car.

EMBEDDED SYSTEMS

- ES generally use microcontroller that contains many functions of computer on a single device
- Motorola and Intel make some of the most popular microcontrollers



EMBEDDED SYSTEM DEVICES

CONSUMER ELECTRONICS

Microwave Ovens

Digital Cameras

DVD Player

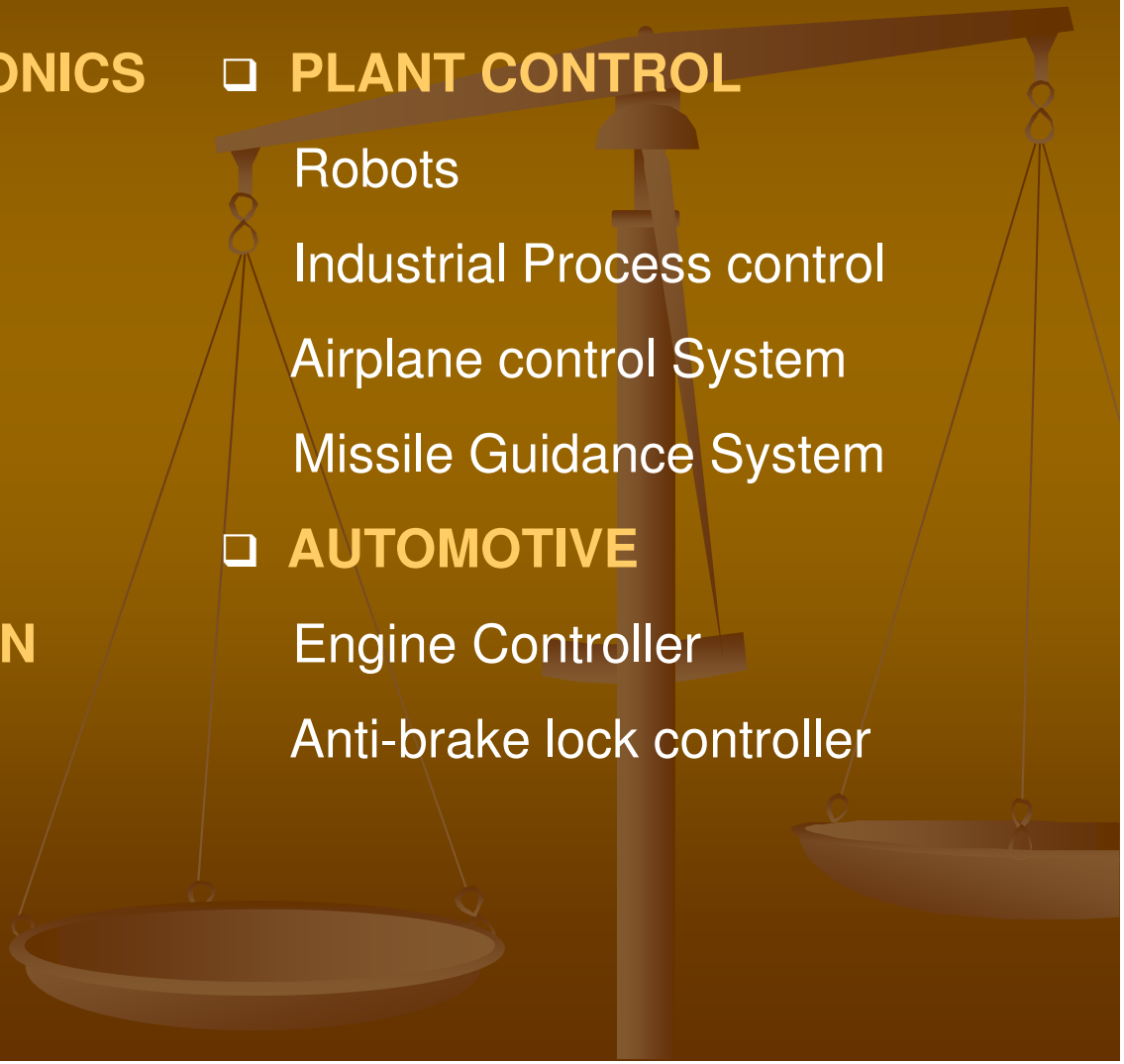
Washing Machine

TELECOMMUNICATION

Switches

Cellular Phones

- **PLANT CONTROL**
  - Robots
  - Industrial Process control
  - Airplane control System
  - Missile Guidance System
- **AUTOMOTIVE**
  - Engine Controller
  - Anti-brake lock controller



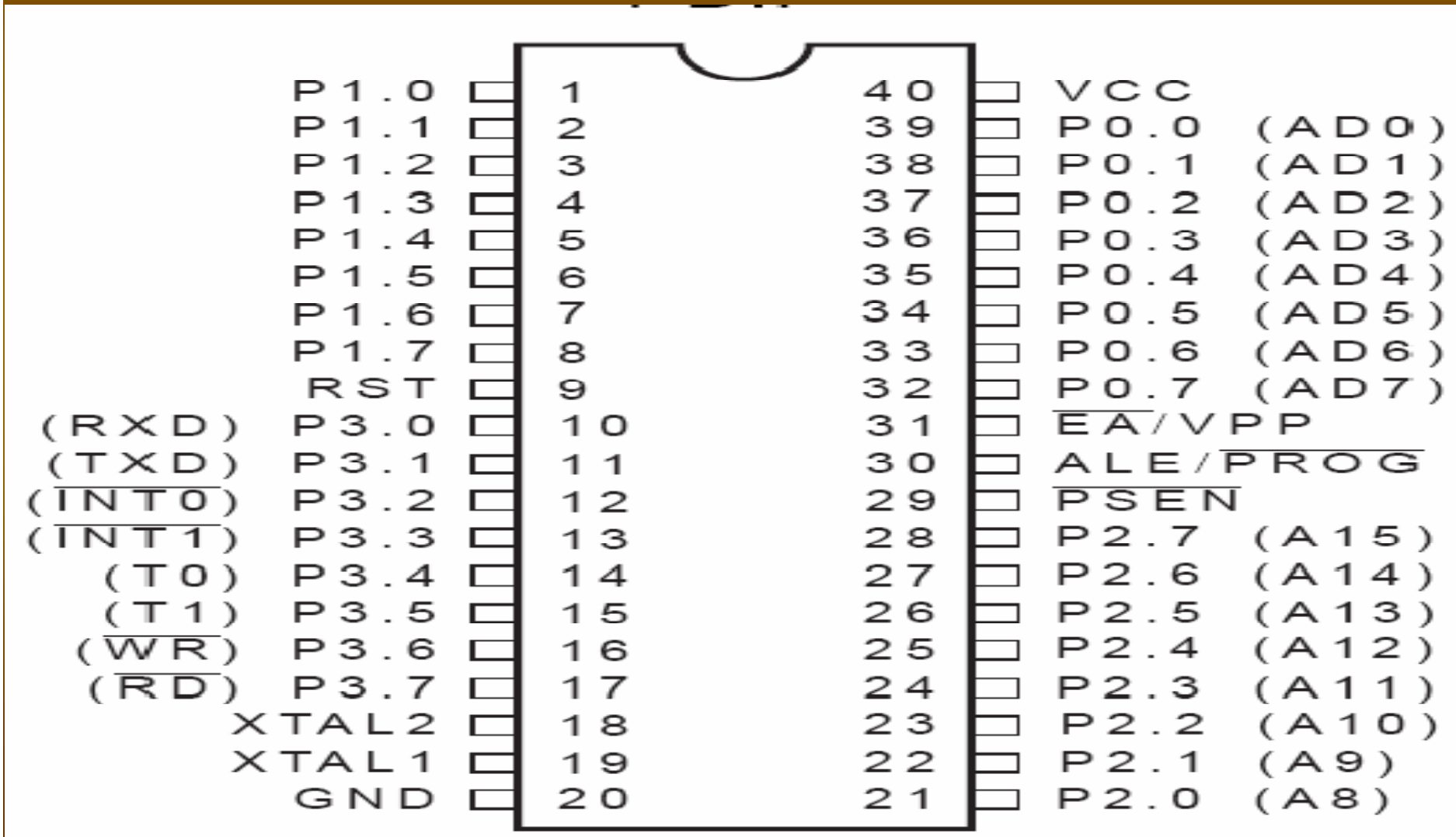
Percentage share of various verticals in Embedded Software market

Datacom	34%
Consumer Electronics	20%
Industrial Automation	19%
Automotives	10%
Office Automation	8%

Some feature that makes 8051 popular

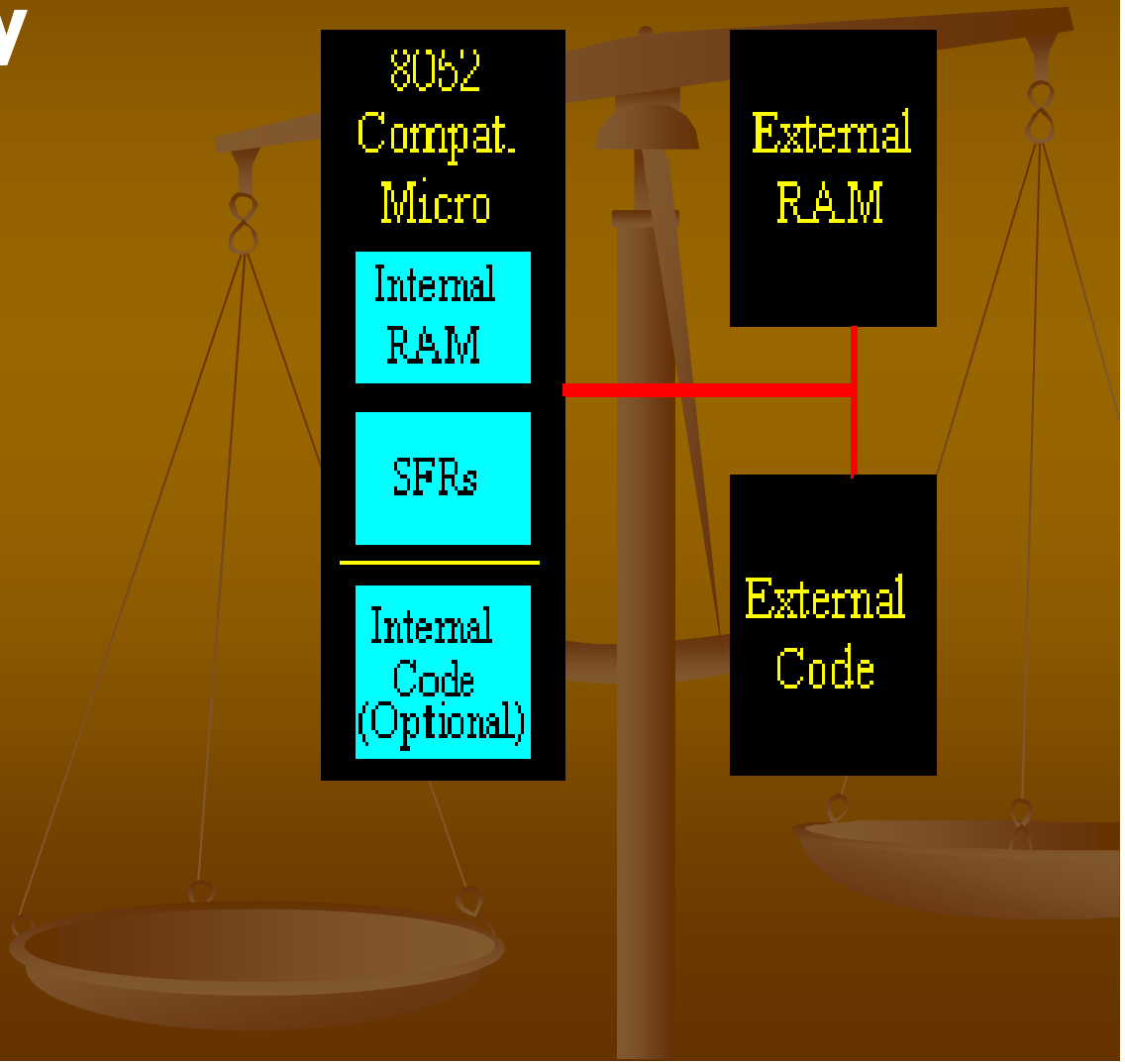
- 8-bit data bus
- 16-bit address bus
- 34 general purpose registers each of 8 bits
- 16 bit timers (usually 2, but may have more, or less).
- 3 internal and 2 external interrupts.
- Bit as well as byte addressable RAM area of 16 bytes.
- Four 8-bit ports, (short models have two 8-bit ports).
- 16-bit program counter and data pointer

PIN DIAGRAM OF 8051

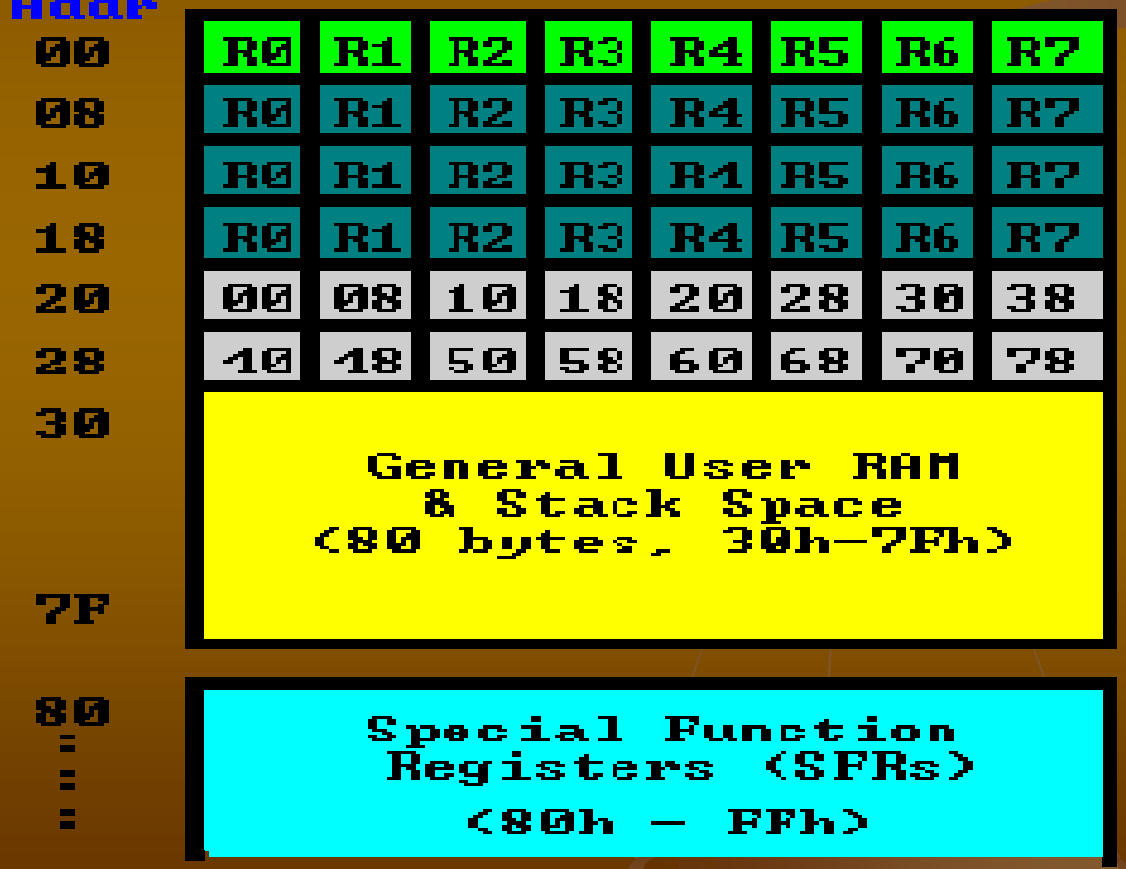


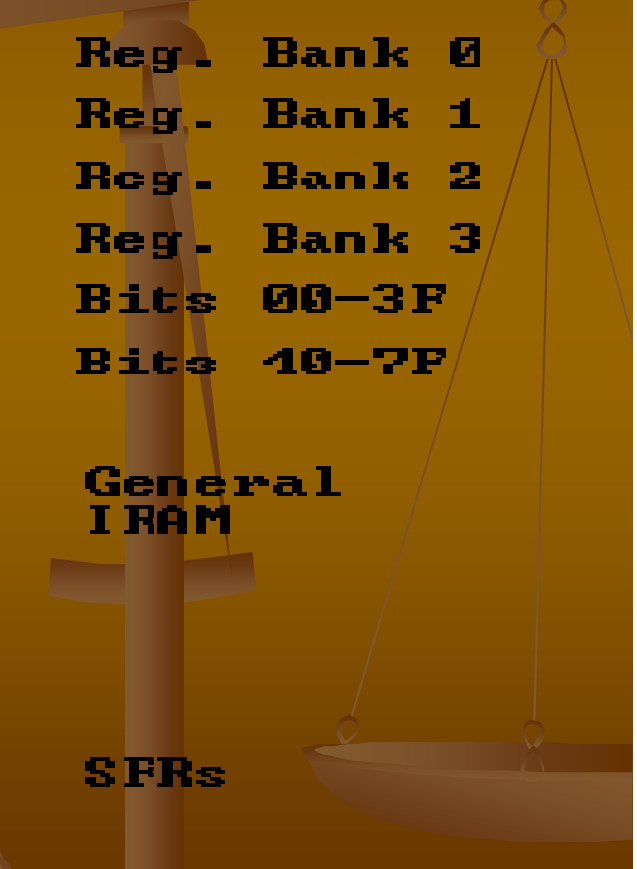
TYPES OF MEMORY

- On-Chip Memory
- External Code Memory
- External RAM



On chip memory





It has four parts on which it can be explained :

- Register banks- R0,R1,R2 and R4

are basically used to manipulate data from one memory to another memory place.

- BIT MEMORY

gives the user the ability to access a number of *bit variables*.

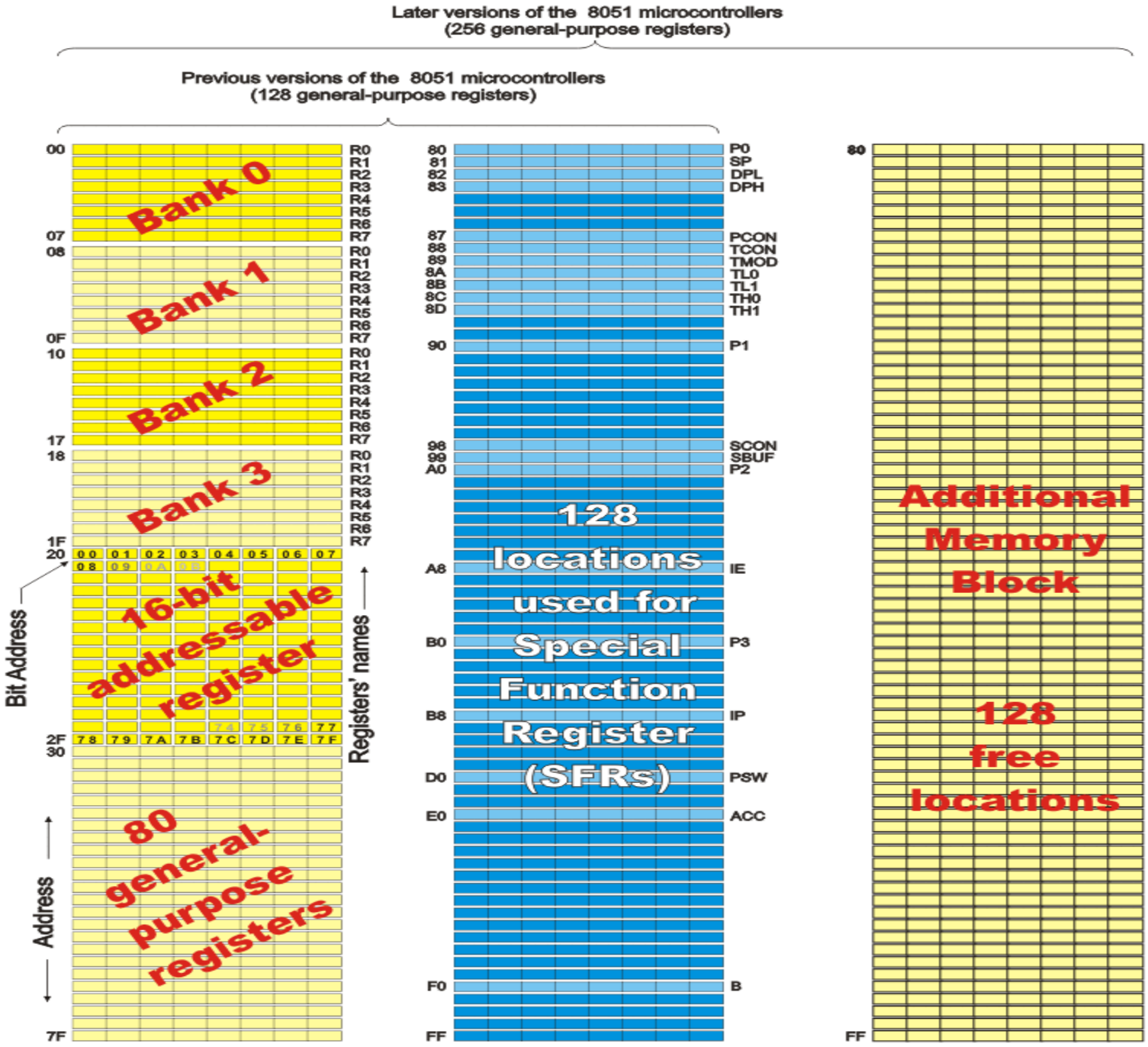
There are 128 bit variables available to the user, numbered 00h through 7Fh

- GENERAL PURPOSE REGISTERS

use to store memory addresses and data

- SPECIAL FUNCTION REGISTER(SFR)

Special Function Registers (SFRs) are areas of memory that control specific functionality of the 8051 processor . Four SFRs permit access to the 8051s 32 input/output lines .


Later versions of the 8051 microcontrollers
(256 general-purpose registers)
Previous versions of the 8051 microcontrollers
(128 general-purpose registers)
Bank 0
Bank 1
Bank 2
Bank 3
16-bit addressable register
80 general-purpose registers
Bit Address
Address
Registers' names
128 locations used for Special Function Register (SFRs)
Additional Memory Block
128 free locations
P0
SP
DPL
DPH
PCON
TCON
TMOD
TL0
TL1
TH0
TH1
P1
SCON
SBUF
P2
IE
P3
IP
PSW
ACC
B

SFR

80	P0	SP	DPL	DPH				PCON	87
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
90	P1								97
98	SCON	SBUF							9F
A0	P2								A7
A8	IE								AF
B0	P3								B7
B8	IP								B9
C0									C7
C8									CF
D0	PSW								D7
D8									DF
E0	ACC								E7
E8									EF
F0	B								F7
F8									FF

Blue background are I/O port SFRs
Yellow background are control SFRs
Green background are other SFRs

TIMER

The 8051 comes equipped with two timers, both of which may be controlled, set, read, and configured individually. The 8051 timers have three general functions:

1) Keeping time and/or calculating the amount of time between events,
2) Counting the events themselves,
3) Generating baud rates for the serial port.

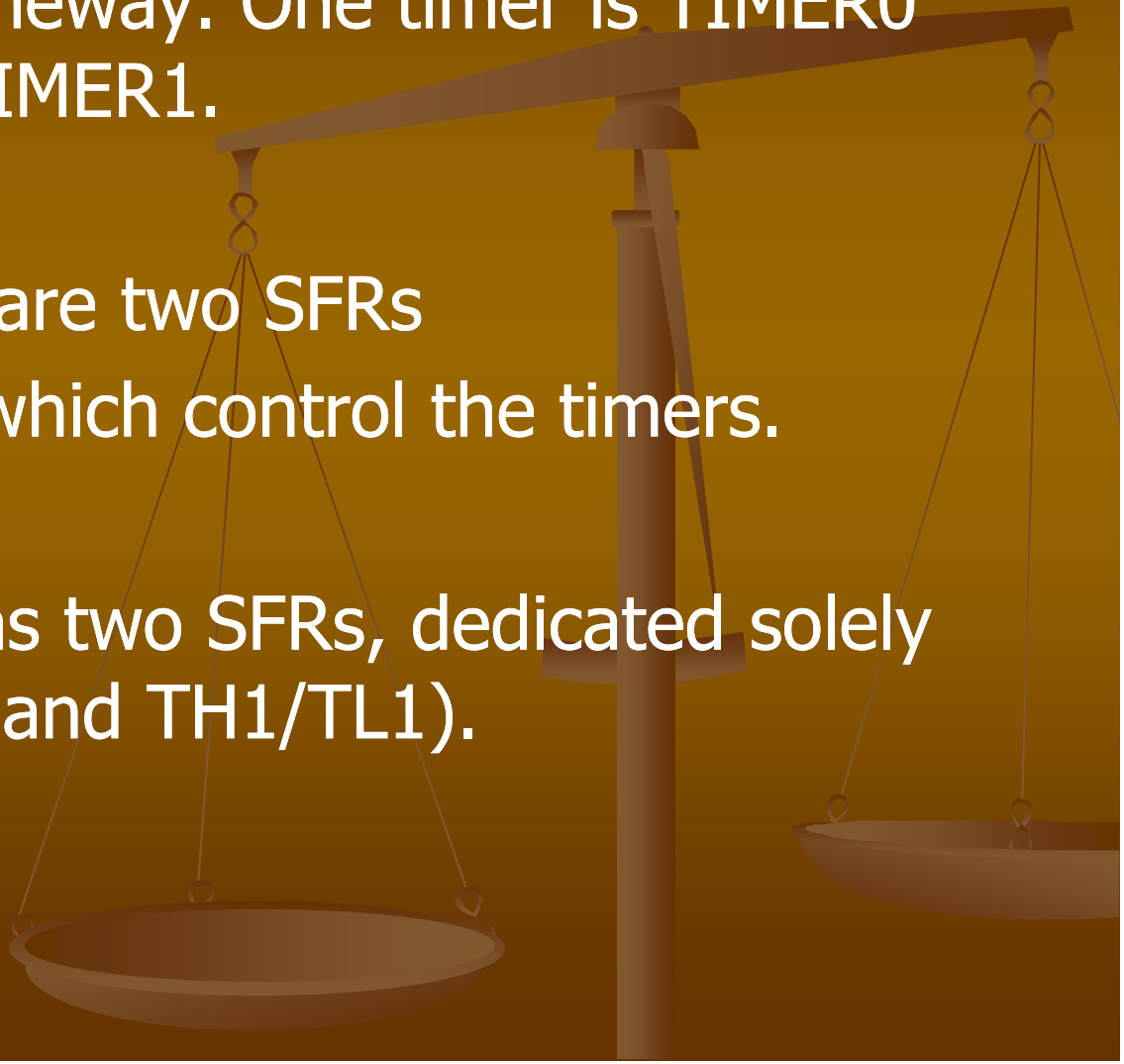
Timer SFRs

- The 8051 has two timers which each function essentially the sameway. One timer is TIMER0 and the other is TIMER1.

- The two timers share two SFRs

(TMOD and TCON) which control the timers.

- Each timer also has two SFRs, dedicated solely to itself (TH0/TL0 and TH1/TL1).



Timer SFR'S

SFR Name	Description	SFR Address
TH0	Timer 0 High Byte	8Ch
TL0	Timer 0 Low Byte	8Ah
TH1	Timer 1 High Byte	8Dh
TL1	Timer 1 Low Byte	8Bh
TCON	Timer Control	88h
TMOD	Timer Mode	89h

TIMER MODES

TxM1	TxM0	Timer Mode	Description of Mode
0	0	0	13-bit Timer.
0	1	1	16-bit Timer
1	0	2	8-bit auto-reload
1	1	3	Split timer mode

TMOD SFR'S

Bit	Name	Explanation of Function	Timer
7	GATE1	When this bit is set the timer will only run when INT1 (P3.3) is high. When this bit is clear the timer will run regardless of the state of INT1.	1
6	C/T1	When this bit is set the timer will count events on T1 (P3.5). When this bit is clear the timer will be incremented every machine cycle.	1
5	T1M1	Timer mode bit (see below)	1
4	T1M0	Timer mode bit (see below)	1

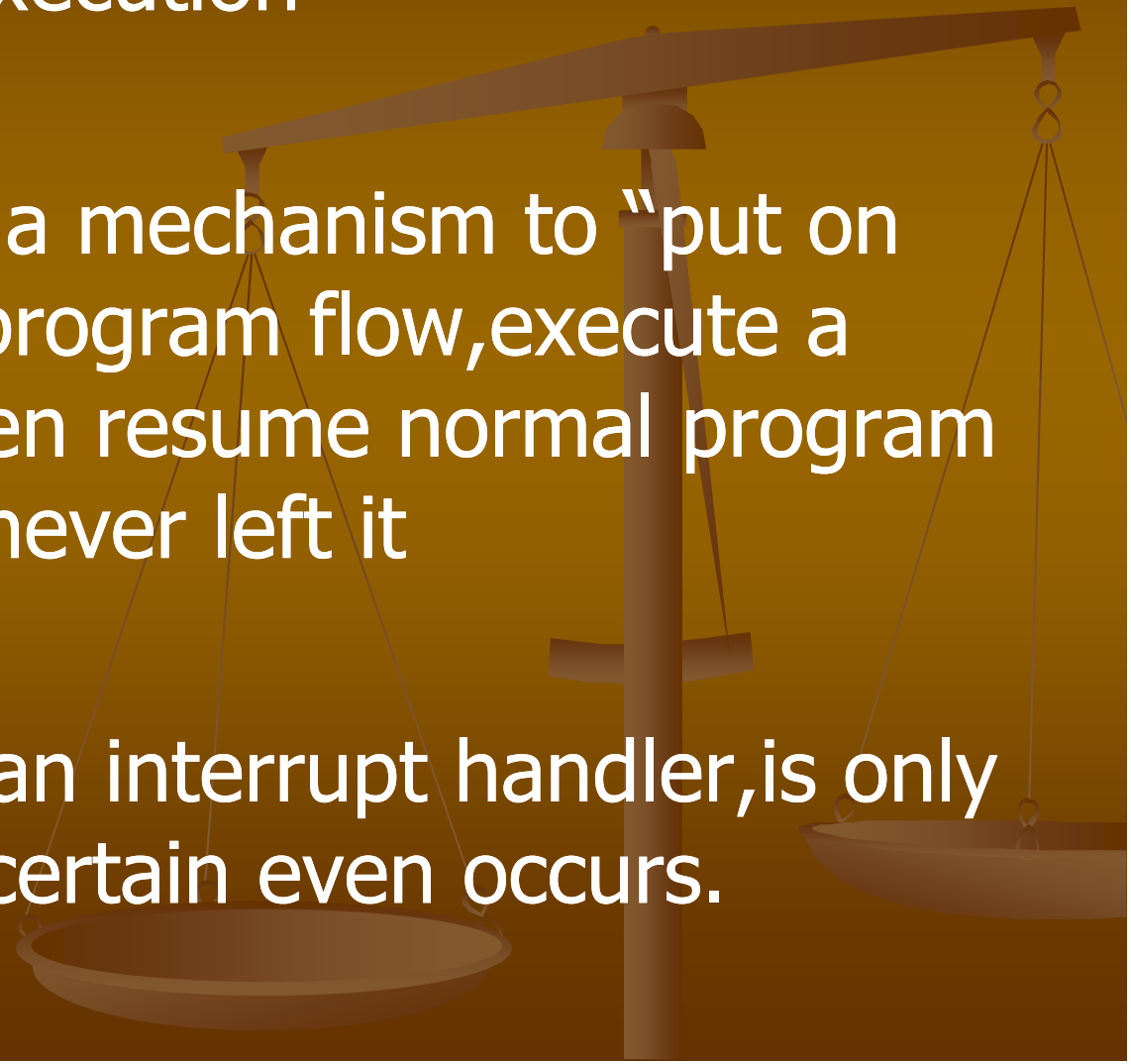
3	GATE0	When this bit is set the timer will only run when INT0 (P3.2) is high. When this bit is clear the timer will run regardless of the state of INT0.	0
2	C/T0	When this bit is set the timer will count events on T0 (P3.4). When this bit is clear the timer will be incremented every machine cycle.	0
1	T0M1	Timer mode bit (see below)	0
0	T0M0	Timer mode bit (see below)	0

TCON SFR'S

Bit	Name	Bit Address	Explanation of Function	Timer
7	TF1	8Fh	**Timer 1 Overflow**. This bit is set by the microcontroller when Timer 1 overflows.	1
6	TR1	8Eh	**Timer 1 Run**. When this bit is set Timer 1 is turned on. When this bit is clear Timer 1 is off.	1
5	TF0	8Dh	**Timer 0 Overflow**. This bit is set by the microcontroller when Timer 0 overflows.	0
4	TR0	8Ch	**Timer 0 Run**. When this bit is set Timer 0 is turned on. When this bit is clear Timer 0 is off.	0

Interrupt

- An interrupt is some event which interrupts normal program execution
- Interrupts give us a mechanism to "put on hold" the normal program flow,execute a subroutine,and then resume normal program flow as if we had never left it
- Subroutine called an interrupt handler,is only executed when a certain even occurs.



8051 Microcontroller Interrupts

- There are five interrupt sources for the 8051, which means that they can recognize 5 different event that can interrupt regular program execution.
- Event may be one of the timers”overflowing”,receiving a character via the serial port,trasmitting a character via the serial port or one of the two “external events”
- Each interrupt can be enabled or disabled by setting bits in the IE register. Also, as seen from the picture below the whole interrupt system can be disabled by clearing bit EA from the same register.

IE SFR

Bit	Name	Bit Address	Explanation of Function
7	EA	AFh	Global Interrupt Enable/Disable
6	-	AEh	Undefined
5	-	ADh	Undefined
4	ES	Ach	Enable Serial Interrupt
3	ET1	ABh	Enable Timer 1 Interrupt
2	EX1	AAh	Enable External 1 Interrupt
1	ET0	A9h	Enable Timer 0 Interrupt
0	EX0	A8h	Enable External 0 Interrupt

INTERRUPT HANDLING

Interrupt	Flag	Interrupt Handler Address
External 0	IE0	0003h
Timer 0	TF0	000Bh
External 1	IE1	0013h
Timer 1	TF1	001Bh
Serial	RI/TI	0023h

SERIAL INPUT/OUTPUT PORT

One of the 8051's many powerful features is it's integrated *UART*, otherwise known as a serial port. The fact that the 8051 has an integrated serial port means that you may very easily read and write values to the serial port. If it were not for the integrated serial port ,writing a byte to a serial line would be a rather tedious process requiring turning on and off one of the I/O lines in rapid succession to properly "clock out" each individual bit ,including start bits, stop bits, and parity bits. However, we do not have to do this. Instead, we simply need to configure the serial port's operation mode and baud rate. Once configured, all we have to do is write to an SFR to write a value to the serial port or read the same SFR to read a value from the serial port.

The 8051 will automatically let us know when it has finished sending the character we wrote and will also let us know whenever it has received a byte so that we can process it. We do not have to worry about transmission at the bit level--which saves us quite a bit of coding and processing time.

SCON SFR

Bit	Name	Bit Addres	Explanation of Function
7	SM0	9Fh	Serial port mode bit 0
6	SM1	9Eh	Serial port mode bit 1.
5	SM2	9Dh	Mutliprocessor Communications Enable (explained later)
4	REN	9Ch	Receiver Enable. This bit must be set in order to receive characters.
3	TB8	9Bh	Transmit bit 8. The 9th bit to transmit in mode 2 and 3.
2	RB8	9Ah	Receive bit 8. The 9th bit received in mode 2 and 3.
1	TI	99h	Transmit Flag. Set when a byte has been completely transmitted.
0	RI	98h	Receive Flag. Set when a byte has been completely received.

SM0	SM1	Serial Mode	Explanation	Baud Rate
0	0	0	8-bit Shift Register	Oscillator / 12
0	1	1	8-bit UART	Set by Timer 1 (*)
1	0	2	9-bit UART	Oscillator / 32 (*)
1	1	3	9-bit UART	Set by Timer 1 (*)

THANK YOU