1. Block diagram of microcontroller

2. Architecture of 8051 microcontroller
4KB on chip memory, 128 bytes of data memory.
- Program memory can be increased to 64KB.

B Register - It is used during multiply and divide operations to store the second operand for multiply & divide instructions "MUL AB & DIV AB" and "DIV AB & R" respectively.

PSW (Program Status Word) - User-defined flag

<table>
<thead>
<tr>
<th>CY</th>
<th>AC</th>
<th>FO</th>
<th>RS1</th>
<th>RS0</th>
<th>OV</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSW7</td>
<td>PSW6</td>
<td>PSW5</td>
<td>PSW4</td>
<td>PSW3</td>
<td>PSW2</td>
<td>PSW1</td>
</tr>
</tbody>
</table>

- CY → Carry flag
- AC → Auxiliary Carry flag
- OV → Overflow flag
- FO → Flag 0, available to the user for general purpose
- P → Parity Flag
- RS1 & RS0 -> Register bank selector

Accumulator - (16-bit register)

Stack pointer (SP) - (8-bit register)

Data Pointer (DPTR) - (16-bit register)

It consists of a higher byte (DPH) and a lower byte (DPL) of a 16-bit external data RAM address.

Port 0, Port 1, Port 2, Port 3 bitches and Data - $ (8-bit data bus)

Each latch and corresponding drives of port 0-3 is allotted to the corresponding on-chip I/O port.

Serial port data buffer - (SBUF) (SBVF)

Consists of transmit buffer (parallel-in - serial-out register) and serial data receive buffer (serial-in-parallel-out register).

Timing Registers - (Two 16-bit timing registers)

T0H & T0L represent higher & lower byte of timing register 0.

T1H & T1L represent higher and lower byte of timing register 1 respectively.
- **Control Registers**: (8-bit registers)
  - Instruction Pointer (IP), Interrupt Enable (IE), Timer Mode (TMOD),
  - Timer Control (TCON), Serial Port Control (SCON) and Power Control (PCON)
- **Oscillator**: quartz crystal
- **Turing and Control Unit**
  - Address latch enable (ALE), Program store enable (PSEN), RD, WR
- **Instruction Register**: Used to decide the opcode of any instruction to be executed.
- **Program Address Register**
  - On-chip EPROM
  - RAM = 128 bytes (Overall 256 bytes on-chip)
  - RAM Address Register: Used to generate address of RAM internally.
- **AHU** (Arithmetic and Logic Unit) → Operands in TMP1 and TMP2
- **SFR** (Special Function Register): This register bank is a set of registers, which can be addressed using their respective addresses in the range of 80H to FFH. Then
  - ON: IP, IE, TMOD, TCON, SCON, PCON

**Memory Organization**

```
+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+
| FFFF      | FFEE      | FF       | FF       | FF        | FF        | FF        | FF        | FF        |
| 60 K bytes| 64 K bytes| 128 bytes| 256 bytes| 64 K bytes|
+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+
| AND       | OR        | FF        | FF        |
| 4 K bytes | 0        | 0         | 0         |
+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+-----------+
| 0000      | 0000      | 0000      | 0000      |
```

- **Program Memory** (Read-Only) Structure
- **Data Memory** (Read/Write) Structure

- **Program Memory**
  - When EA pin is low/grounded, the program memory is external and when
  - EA pin is high/Vcc, the address from 0000H to OFFFH will refer to on-chip memory and the address from 1000H to FFFFH can refer to external memory.
8051 Program Memory

When ALE is high, Port 0 is used as the low byte of the Program Counter (PC1) as an address. Port 2 is used as the high byte of the Program Counter (PC1). Then PSEN strobes the EPROM and the code byte is read into the microcontroller.

External Data Memory

MOVX instruction is used to access the external data memory.

However, 128 bytes of RAM can be divided into three segments:

1. Register Banks - 00H - 1FH (4 Banks and each Bank has 8 registers (Rx))
2. Port addressable segment - 20H - 2FH (Address as either byte or individual bit)
3. Scratch Pad Area - 30H - 7FH or general purpose RAM.
Upper 128 bytes of the on-chip RAM are used for special function registers. Only 25 bytes are used. Other bytes reserved for advanced versions of microcontroller.
Two 8051 microcontrollers have 16-bit timers/counters such as TIMER0 (T0) and TIMER1 (T1). Each timer can be programmed to count external clock pulses of 8051 microcontroller. These timers are used for the following functions:

- Calculate time delay between two events
- Count the number of events
- Generate baud rate for serial ports
- Frequency measurement
- Pulse width measurement

---

**Timer 1 register**

```
TH1 | TL1
---|---
P15 | D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
```

**Timer 0 register**

```
TH0 | TL0
---|---
P15 | D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
```

---

**Other Timer Modes**

<table>
<thead>
<tr>
<th>M1</th>
<th>M0</th>
<th>Operating Mode</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Mode 0</td>
<td>13-bit timer mode</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Mode 1</td>
<td>16-bit timer mode</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Mode 2</td>
<td>8-bit timer mode</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Mode 3</td>
<td>8-bit timer mode</td>
</tr>
</tbody>
</table>

---

**Timer Mode 0** - Count value \( 0000 \text{H} - 1FFFF \text{H} \)

Overflow flag is set to zero after \( 2^5 \times 2^8 = 8192 \) machine cycles.

---

**Circuit Diagram**

```
Osc \rightarrow \frac{1}{12} \rightarrow \frac{c}{T} = 0 \rightarrow O\uparrow \rightarrow \leftarrow \frac{c}{T} = 1
```

Clock frequency input to \( T_{X(Pin)} \) is \( \frac{c}{T} = 12 \times 25 \)
Timer mode 1 - Count value → 0000H - FFFFH
Overflow flag is set to zero after $2^8 \times 2^8 = 65536$ machine cycles.

\[
\text{Timer delay} = 12 \times (65,536 - (\text{TH} \times \text{TL} \times 256))
\]

Timer mode 2 - (has auto-reload feature) Count value → 00H - FFH
Overflow flag is set to zero after $2^8 = 256$ machine cycles.

\[
\text{Timer delay between overflows} = 12 \times (256 - \text{TH})
\]

Timer mode 3 -

Timer 0 divides into two 8-bit counters, Timer 0 and Timer 1. TH0 and
TH1 are two separate timers with overflow flags TF1 and TF0 respectively.

---

Counters →

\[
\text{Osc.} \rightarrow \div 12 \rightarrow \text{TH} \rightarrow \text{TF} \rightarrow \text{Interrupt}
\]

---

Timer 0 counts CPU cycles using TR1, TF1 and Timer 1 interrupt.

---

\[\text{Counters} \rightarrow \text{Tx} \text{ pins (To and T1)}\]
→ **Control Registers**

1. **TCON Register** (Timer/Counter Control Register) -
   
<table>
<thead>
<tr>
<th>D8</th>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF1</td>
<td>TR1</td>
<td>TF0</td>
<td>TR0</td>
<td>IE1</td>
<td>IT1</td>
<td>IE0</td>
<td>IT0</td>
<td></td>
</tr>
</tbody>
</table>

   - TF1 → Timer 1 overflow flag
   - TR1 → Timer 1 run control flag
   - IE1 → External Interrupt 1 edge flag
   - IT1 → Interrupt 1 type control bit

2. **TMOD Register** (Time/Mode Control Register) -
   
<table>
<thead>
<tr>
<th>D7</th>
<th>D6</th>
<th>D5</th>
<th>D4</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>GATE</td>
<td>C/T</td>
<td>M1</td>
<td>M0</td>
<td>GATE</td>
<td>C/T</td>
<td>M1</td>
<td>M0</td>
</tr>
</tbody>
</table>

   - **Gate** - 1 then timer/counter x will operate only when INTx pin is high for hardware control. When GATE = 0, timer/counter x will run only if TRx = 1 for software control.
   - **C/T (Clock/Timer)** - 0 then timer is used for timer delay otherwise 1 then timer is used as counter by counting pulse from external input pin x (Tx, Tx and Tx).
   - **M1 & M0** - Timer/Counter operating mode selector bits

(5) **Interrupts of 8051** -

<table>
<thead>
<tr>
<th>Interrupt Source</th>
<th>Flag</th>
<th>Vector Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Interrupt 0</td>
<td>IE0</td>
<td>000BH</td>
</tr>
<tr>
<td>Timer 0</td>
<td>TF0</td>
<td>000BH</td>
</tr>
<tr>
<td>External Interrupt 1</td>
<td>IE1</td>
<td>0013H</td>
</tr>
<tr>
<td>Timer 1</td>
<td>TF1</td>
<td>0018H</td>
</tr>
<tr>
<td>Serial Port</td>
<td>RI/TI</td>
<td>0028H</td>
</tr>
</tbody>
</table>

→ Each interrupt can be enabled separately. Each interrupt has a separate vector address, and can be programmed to one of two priority levels.
Internal interrupts can be enabled or disabled by setting bit of IE (Interrupt enable). Whole interrupt system can be disabled by clearing the EA bit of the control register.

![Diagram of interrupt system]

Interrupts of 8051

1. **Interrupt Enable (IE) Register**
   - EA: Global interrupt enable/disable
   - ES: Enable Serial Interrupt
   - ETx: Enable Timer x Interrupt
   - EXx: Enable External x Interrupt

2. **Interrupt Priority (IP) Register**
   - PS: Serial port interrupt priority level
   - PTx: Timer x interrupt priority level
   - PXx: External interrupt x priority level

3. **Timer Control Register (ICON)**

Execution of interrupt:
- Main program execution → Interrupt occurs → Load vector address of interrupt
- Jump back to main → Execution of interrupt service routine → Jump to vector address → Program execution → Start executing next instruction of main program.
6. Addressing modes of 8051
   (i) Immediate Addressing - hard data immediately. Eg - MOV A, #FFH
   (ii) Register Addressing mode - Registers R0 to R7 (Register bank). Eg - MOV A, R0.
   (iii) Direct Addressing - specified 8-bit address field. Eg - MOV A, 03H.
   (iv) Indirect Addressing - specifies a register which contains the address of the operand. Eg - MOV A, @R7.
   (v) Indexed Addressing - Only program memory can be accessed and it can only be read. Eg - MOV A, @A + DPTR [Combines offset to register + contents of program counter to point to register].
   (vi) Relative Addressing - Used generally, used in certain jump instructions.
   The relative address is an 8-bit signed number (-128 to 127), which is added to the program counter (PC) to determine the address of the next instruction. Eg - JMP A+DPTR
   (vii) Absolute Addressing - Used with JMP and CALL instructions.
   The 11 least significant bits of the destination address come from the opcode and the upper five bits are the current upper five bits of the program counter (PC). Eg: CALL addr11
   (viii) Register Addressing Used with JSRMP and ACAAL instructions

7. 8051 Instruction set - (256 possible instructions)
(1) Arithmetic Instructions - ADD, ADDC (Add register to accumulator with carry), SUBB (Subtract from accumulator).
(2) Logical Instructions - ANL (Logical AND), ORL (Logical OR),
   XRL (Logical Exclusive-OR), CLR (Clear), CPL (Complement),
   RL (Rotate left), RLC (Rotate left through carry), RR (Rotate right),
   RRC (Rotate right through carry), SWAP (Swap nibbles within accumulator).

\[ [A_{3-0}] \leftrightarrow (A_{7-4}) \]
Absolute → within same 2K block
Address → anywhere in 64K memory
Absolute → preceding 128 - following 127

(3) Data Transfer Instructions - MOV, MOVX (Move code and data), NOVX (Move data between accumulator and a byte of external memory)

(4) Boolean Operations Instructions - CLR (Clear), SETB (Set Bit), CPL (Complement), ANL, ORL, MOV, JC (Jump if carry), INC (Increment), INCX (Increment X Register), INDX (Increment X Register), JBC (Jump if carry), JNC (Jump if not carry), JB (Jump if bit set), JNB (Jump if bit not set)

(5) Branching Instructions - ACALL (Absolute subroutine call), LCALL (Long subroutine call), RET (Return from subroutine), RETI (Return from interrupt), AJMP (Absolute jump), LJMP (Long jump), SJMP (Short jump), JZ (Jump if equal), JNZ (Jump if not equal), CJNE (Compare and jump if not equal), DJNZ (Decrement and jump if not zero), NOP (No operation)

(6) PUSH, POP and Exchange Instructions - PUSH, POP, XCH (Exchange), XCHD (Exchange lower-order digit)

Assembly language programs -

eg - Add 49H, which is in the external memory location 9001H and 56H, which is in the external memory location 9002H. The result is to be stored in the external memory location 9003H.

MOV DPTR, #9001
MOVX A, @DPTR
MOV B, A
INC DPTR
MOVX A, @DPTR
ADD A, B
INC DPTR
MOVX @DPTR, A
LJMP 0000
3. Application of microcontrollers:
   (1) Steppe-motor control
   (2) Traffic light control
   (3) Display (LED's)
   (4) A/D Converter Interface
   (5) Keyboard Interface
   (6) Washing Machine Control
   (7) Time Delay

- CONTRACTION
- (to) ADE
- (to) IN
- (to) ENTER