EEE1024: Fundamentals of Electrical and Electronics Engineering

Dr. Sanchit Khatavkar

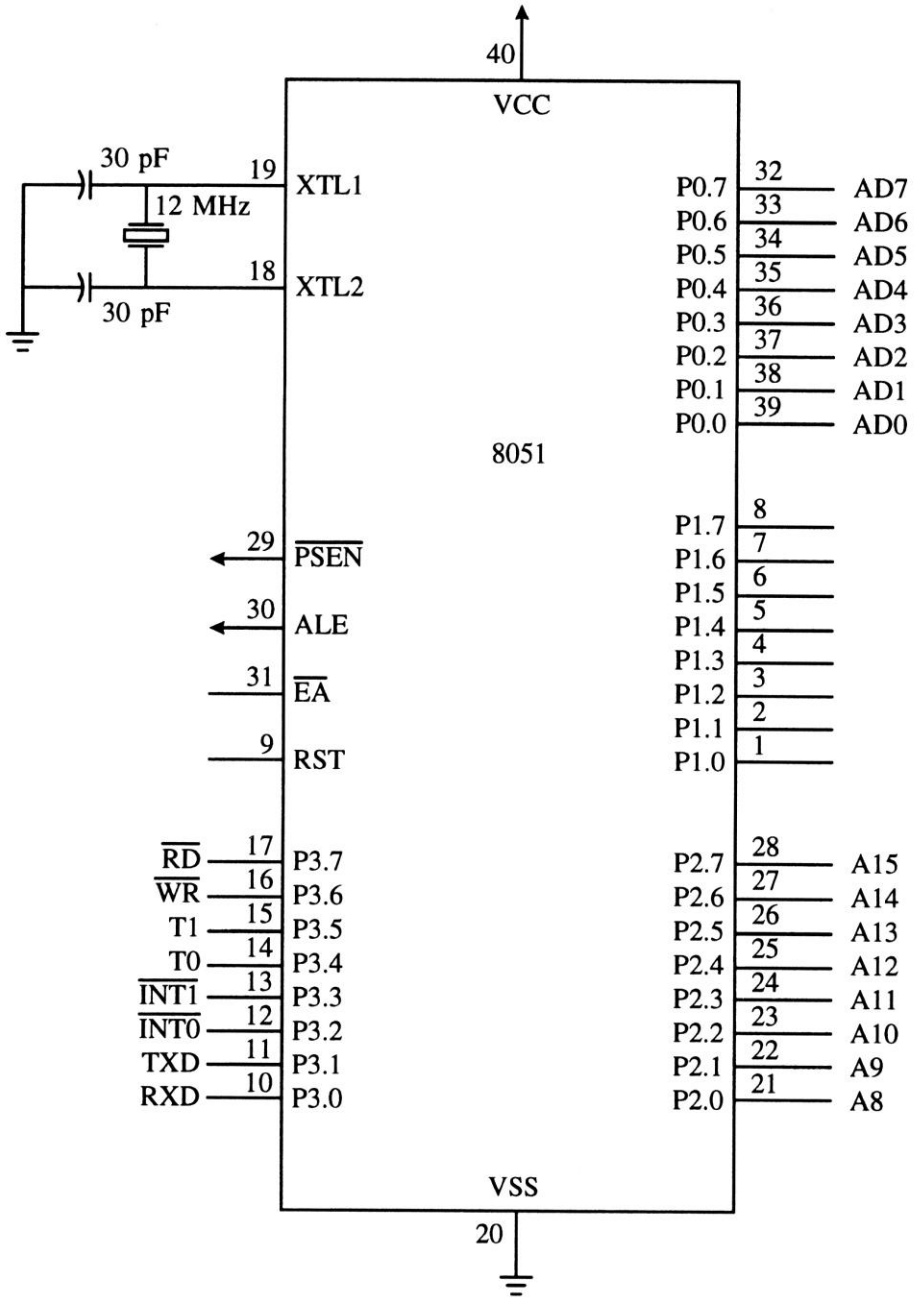
The 8051 Microcontroller

Comparison of the 8051 Family Members

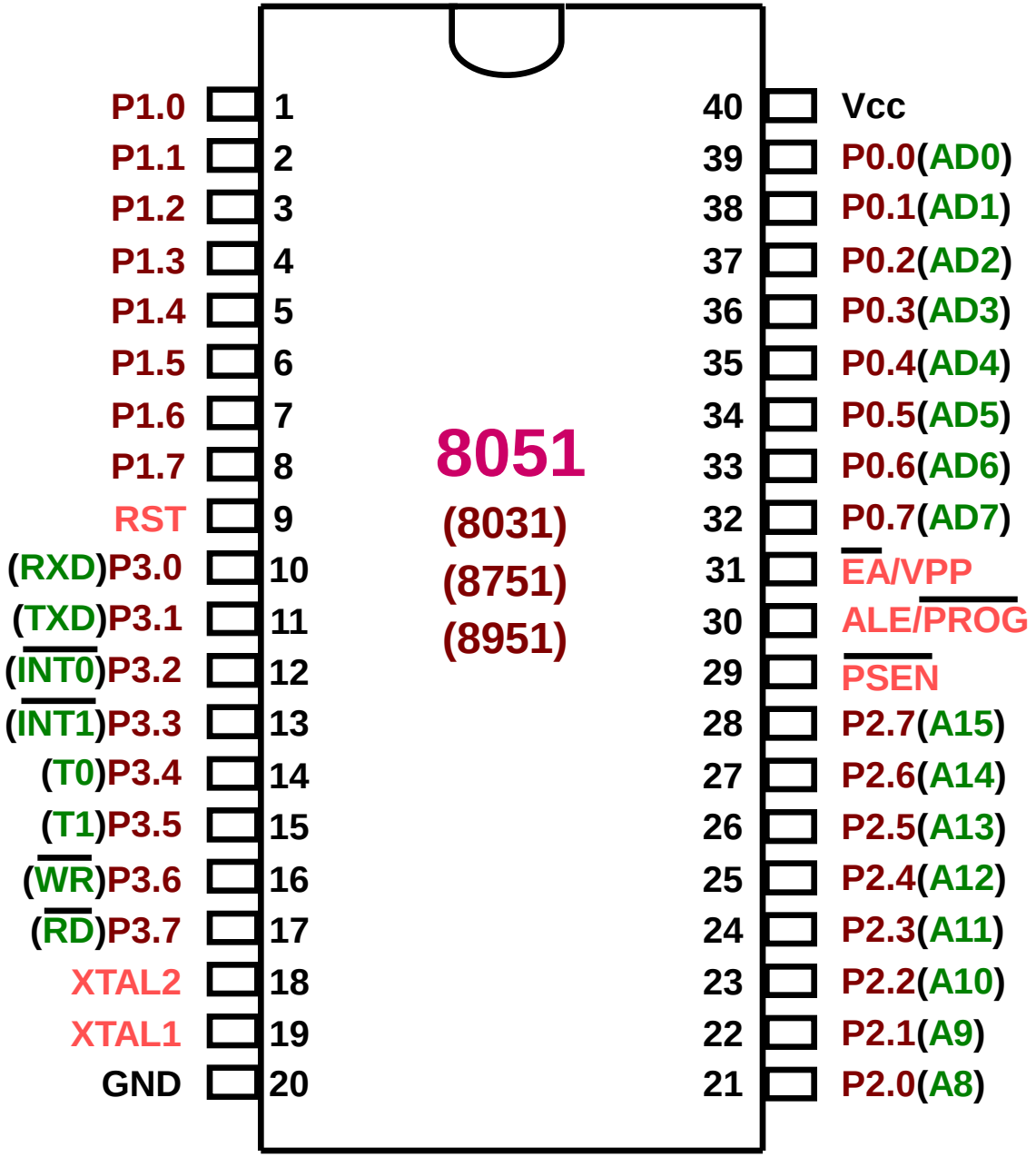
- ROM type
  - 8031 no ROM
  - 80xx mask ROM
  - 87xx EPROM
  - 89xx Flash EEPROM
- 89xx
  - 8951
  - 8952
  - 8953
  - 8955
  - 898252
  - 891051
  - 892051
- Example (AT89C51,AT89LV51,AT89S51)
  - AT= ATMEL(Manufacture)
  - C = CMOS technology
  - LV= Low Power(3.0v)

8051 Schematic Pin out

40 pins!



8051 Foot Print



- **Pins 1 to 8** − These pins are known as **Port 1.** This port doesn't serve any other functions. It is internally pulled up, bi-directional I/O port.
- **Pin 9** − It is a RESET pin, which is used to reset the microcontroller to its initial values.
- **Pins 10 to 17** − These pins are known as **Port 3**. This port serves some functions like interrupts, timer input, control signals, serial communication signals RxD and TxD, etc.
- **Pins 18 & 19** − These pins are used for interfacing an external crystal to get the system clock.
- **Pin 20** − This pin provides the power supply to the circuit.
- **Pins 21 to 28** − These pins are known as **Port 2**. It serves as I/O port. Higher order address bus signals are also multiplexed using this port.
- **Pin 29** − This is PSEN pin which stands for ***Program Store Enable***. It is used to read a signal from the external program memory.
- **Pin 30** − This is EA pin which stands for ***External Access*** input. It is used to enable/disable the external memory interfacing.
- **Pin 31** − This is ALE pin which stands for ***Address Latch Enable***. It is used to demultiplex the address-data signal of port.
- **Pins 32 to 39** − These pins are known as **Port 0**. It serves as I/O port. Lower order address and data bus signals are multiplexed using this port.
- **Pin 40** − This pin is used to provide power supply to the circuit.

IMPORTANT PINS (IO Ports)

- **One of the most useful features of the 8051 is that it contains four I/O ports (P0 - P3)**

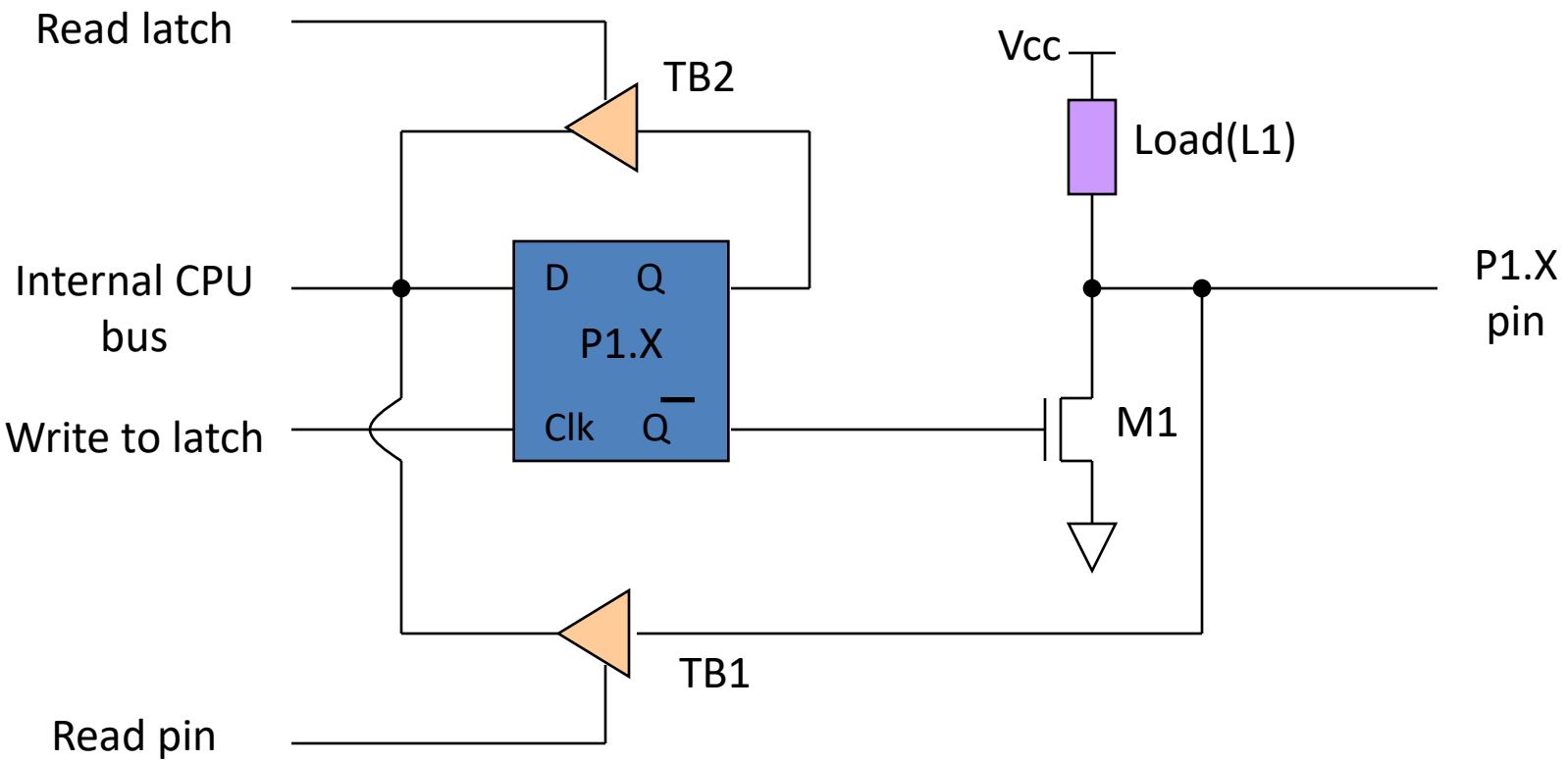
- Port 0 （pins 32-39） : P0 （P0.0～P0.7）
  - 8-bit R/W - General Purpose I/O
  - Or acts as a multiplexed low byte address and data bus for external memory design

- Port 1 （pins 1-8） : P1 （P1.0～P1.7）
  - Only 8-bit R/W - General Purpose I/O

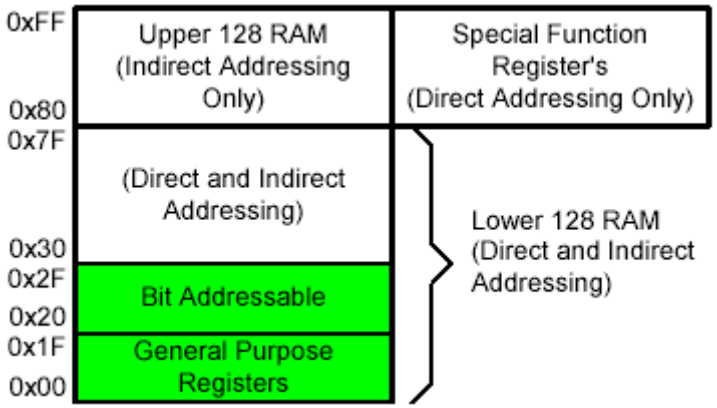
- Port 2 （pins 21-28） : P2 （P2.0～P2.7）
  - 8-bit R/W - General Purpose I/O
  - Or high byte of the address bus for external memory design

- Port 3 （pins 10-17） : P3 （P3.0～P3.7）
  - General Purpose I/O
  - if not using any of the internal peripherals (timers) or external interrupts.
- **Each port can be used as input or output (bi-direction)**

8051 Port 3 Bit Latches and I/O Buffers



On-Chip Memory
Internal RAM



Register Banks

- Active bank selected by PSW [**RS1**,**RS0**] bit
- Permits fast "context switching" in interrupt service routines (ISR).

PSW: PROGRAM STATUS WORD. BIT ADDRESSABLE.

CY	AC	F0	RS1	RS0	OV	—	P

CY	PSW.7	Carry Flag.
AC	PSW.6	Auxiliary Carry Flag.
F0	PSW.5	Flag 0 available to the user for general purpose.
RS1	PSW.4	Register Bank selector bit 1 (SEE NOTE 1).
RS0	PSW.3	Register Bank selector bit 0 (SEE NOTE 1).
OV	PSW.2	Overflow Flag.
—	PSW.1	User definable flag.
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of '1' bits in the accumulator.

The 8051 Assembly Language

Overview

- Data transfer instructions
- Addressing modes
- Data processing (arithmetic and logic)
- Program flow instructions

Data Transfer Instructions

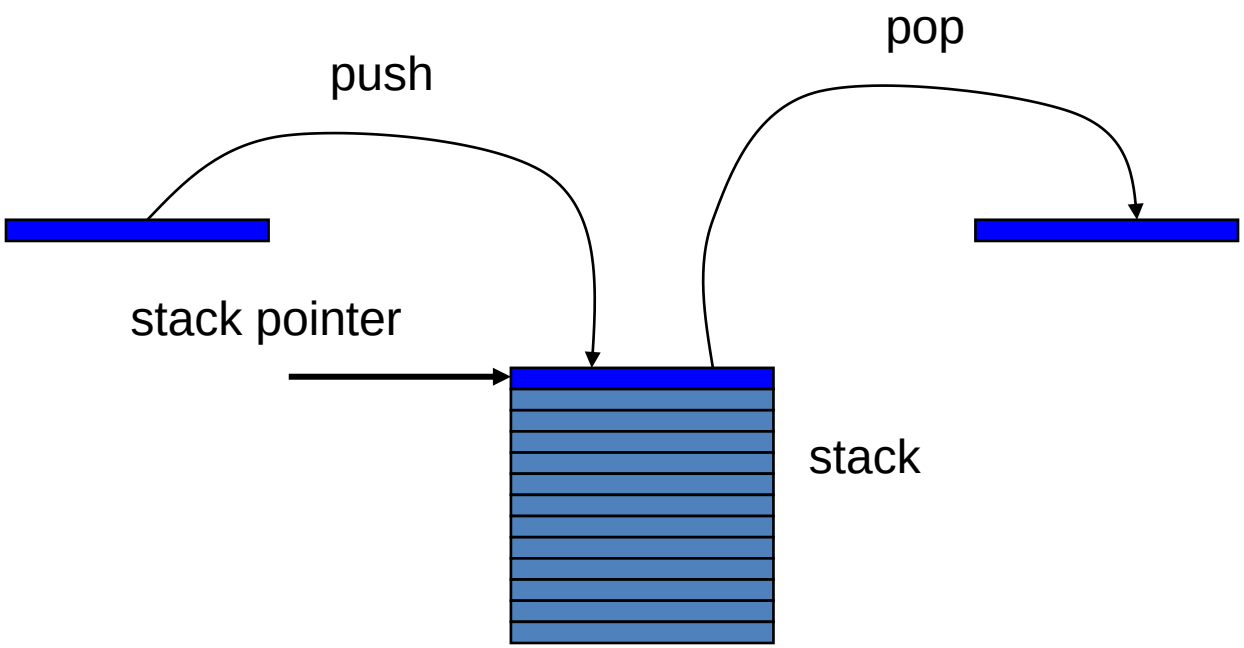
- MOV dest, source dest ← source
- Stack instructions

```
PUSH byte      ;increment stack pointer,
                      ;move byte on stack
POP byte       ;move from stack to byte,
                      ;decrement stack pointer
```

- Exchange instructions

```
XCH a, byte    ;exchange accumulator and byte
XCHD a, byte   ;exchange low nibbles of
                      ;accumulator and byte
```

Stacks



Arithmetic Instructions

- Add
- Subtract
- Increment
- Decrement
- Multiply
- Divide
- Decimal adjust

Arithmetic Instructions

Mnemonic	Description
`ADD A, byte`	add *A* to byte, put result in *A*
`ADDC A, byte`	add with carry
`SUBB A, byte`	subtract with borrow
`INC A`	increment *A*
`INC byte`	increment byte in memory
`INC DPTR`	increment data pointer
`DEC A`	decrement accumulator
`DEC byte`	decrement byte
`MUL AB`	multiply accumulator by b register
`DIV AB`	divide accumulator by b register
`DA A`	decimal adjust the accumulator

Logic Instructions (Bitwise)

ANL → AND

ORL → OR

XRL → XOR

CPL → Complement

Examples:

```
        00001111
ANL     10101100
        00001100

        00001111
ORL     10101100
        10101111

        00001111
XRL     10101100
        10100011

CPL     10101100
        01010011
```

Other Logic Instructions

CLR - clear

RL - rotate left

RLC - rotate left through Carry

RR - rotate right

RRC - rotate right through Carry

SWAP - swap accumulator nibbles

Module -6 Syllabus

- **Measuring Instruments:**

  Classification of instruments, Working principle of PMMC, MI, Digital & Smart Meters, Ammeter, Voltmeter & wattmeter.

- **Sensors:**

  Transducers classification& selections, Resistive, Inductive and capacitive sensors, Optical and Digital sensors.

Acknowledgements

1. https://www.tutorialspoint.com/microprocessor/microcontrollers_8051_architecture.htm
2. https://circuitglobe.com/classification-of-measuring-instruments.html