XMUT202 Digital Electronics

8051 Microcontrollers

Week 10 Lecture 1

School of Engineering and Computer Science

Victoria University of Wellington

Today’s topic

- 8051 microcontrollers.
- Block diagram of 8051 microcontrollers.
- Components of 8051 microprocessors.
- 8051 development kits.

8051: Context

8051: Released by Intel in 1980. 40 years old (!)

- Released as the MCS-51
- Known now as the 8051
- Chief architect: John Wharton
- Originally Intel-only, but many ‘binary compatible’ devices by other manufacturers appeared.
- These variants often feature additional peripherals, built-in memory, etc.
- In ECEN202, we’ll be working with the AT89C51 binary compatible device.

8051

An 8051 microcontroller is **an 8-bit Harvard architecture** microcontroller
Consists of:

- a CPU,
- RAM,
- ROM,
- I/O ports,
- timers, and
- serial communication ports

All combined on one chip.

It can control simple to highly complex operations in embedded systems since it can carry out instructions that are retrieved from its internal memory.

Applications of 8051 MicroControllers

The following industries use the 8051 microcontroller.

Industrial Automation: robotics and control systems.

Consumer electronics: microwave ovens and remote controllers

Automotive: Dashboard screens and engine control devices.

Medical devices: the infusion pumps and patient monitoring systems

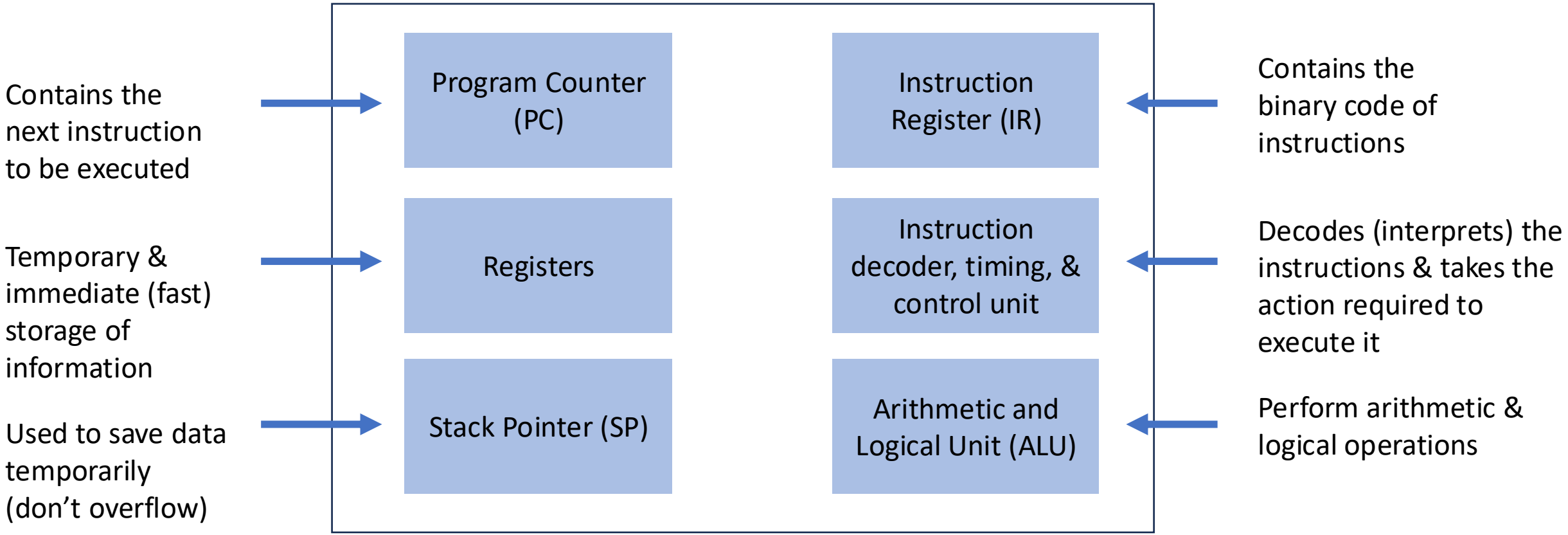
How do 8051 microcontrollers work?

A microcontroller integrates the functions of a microprocessor into a single integrated circuit, functioning as a miniature computer.

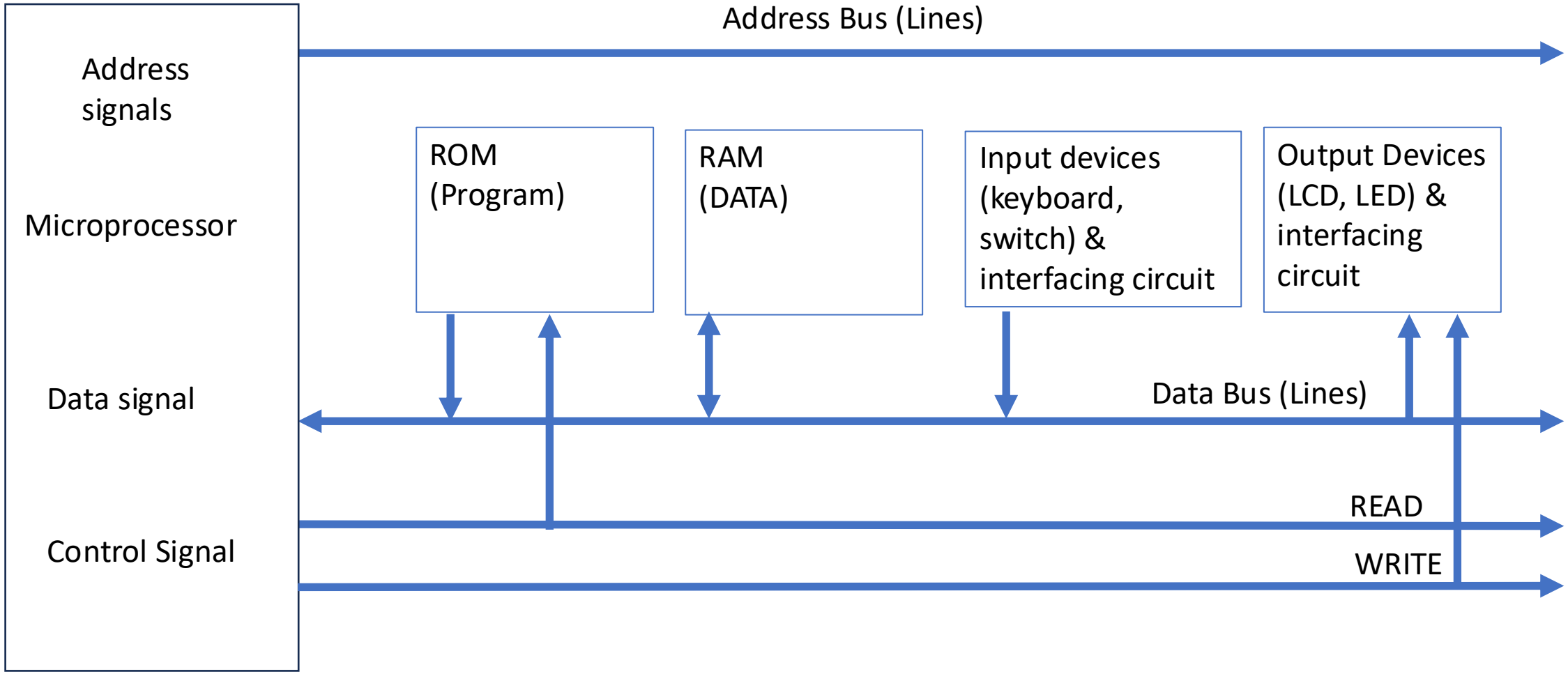
It depends extensively on on-chip features, including RAM, ROM, I/O ports, timers, serial ports, clock circuits, and interrupts, to support a variety of applications.

It *retrieves*, *decodes*, and sequentially *executes* instructions from memory.
It can carry out operations according to *preprogrammed logic* by interacting with external devices via I/O ports and timers.

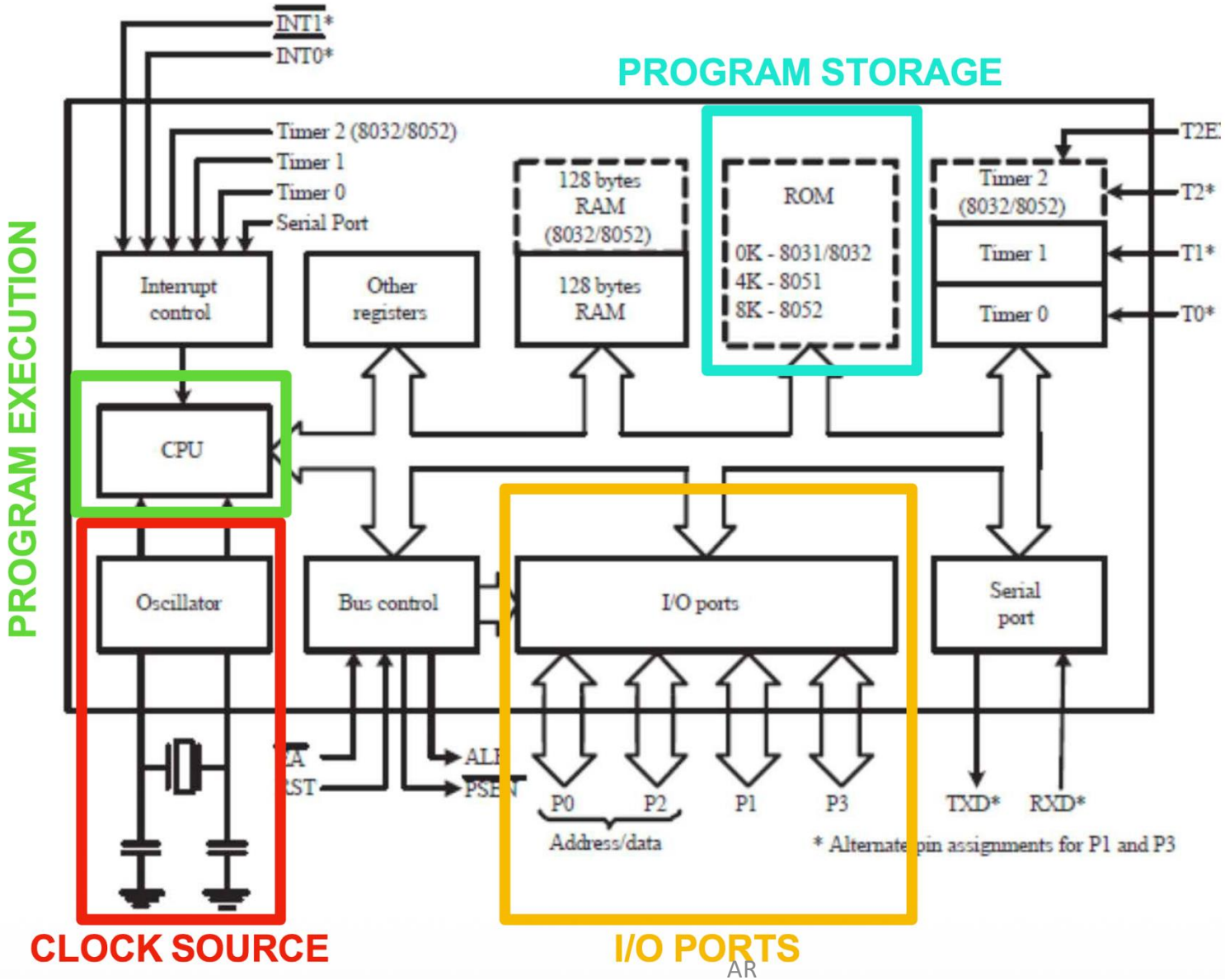
8051 microcontrollers



8051 microcontrollers



8051 microcontrollers: Diagram



8051 - Key Components

1. Central Processing Unit (CPU)

This component controls all controller operations. The in-built elements:

- **ALU:** It is a short form of Arithmetic Logic Unit. This element performs all mathematical operations and also performs logical operations, including NAND, NOT, AND, XOR, and OR.
- **Program Counter:** A 16-bit data address is stored in the program counter. It also tracks the program sequence.
- **Stack Pointer:** It is a variable element that stores another variable's address. It can store addresses of up to 8 bits, where the data was stored last time.
- **Accumulator:** An 8-bit register. It performs automatic storage for all mathematical operations within the controller.
- **Registers:** These are commonly known as the storing units. The registers most commonly used in the 8051 microcontroller are general-purpose registers. This controller comprises 34 general-purpose registers. Two of all the registers - A and B - deal with the mathematical core of the controller, whereas the other 32 registers are part of the internal RAM.
- **Timing and Control:** It performs internal operations with data-flow control and a clock signal at the 8051 microcontroller frequency.

8051 - Key Components

2. Oscillator Circuit

This component of the 8051 microcontroller generates clock pulses to drive the controller's operations. Therefore, it is also called a clock generator.

The oscillator circuit comes with a frequency resonator and various electrical components. The frequency resonator determines the clock pulse frequency and operating speed.

3. ROM & RAM

Like all microcontrollers, the 8051 has certain memory to store programming code and data. This memory is often called a memory chip. Memory is generally divided into two types: RAM and ROM, as discussed below.

- **RAM stands for Random-Access** Memory. It is responsible for storing data, which is why it is named chip data memory.
- **ROM:** The word “ROM” stands for Read-Only Memory. Its primary responsibility is to store the program instructions. ROM can read the program while the controller is operating.

8051 - Key Components

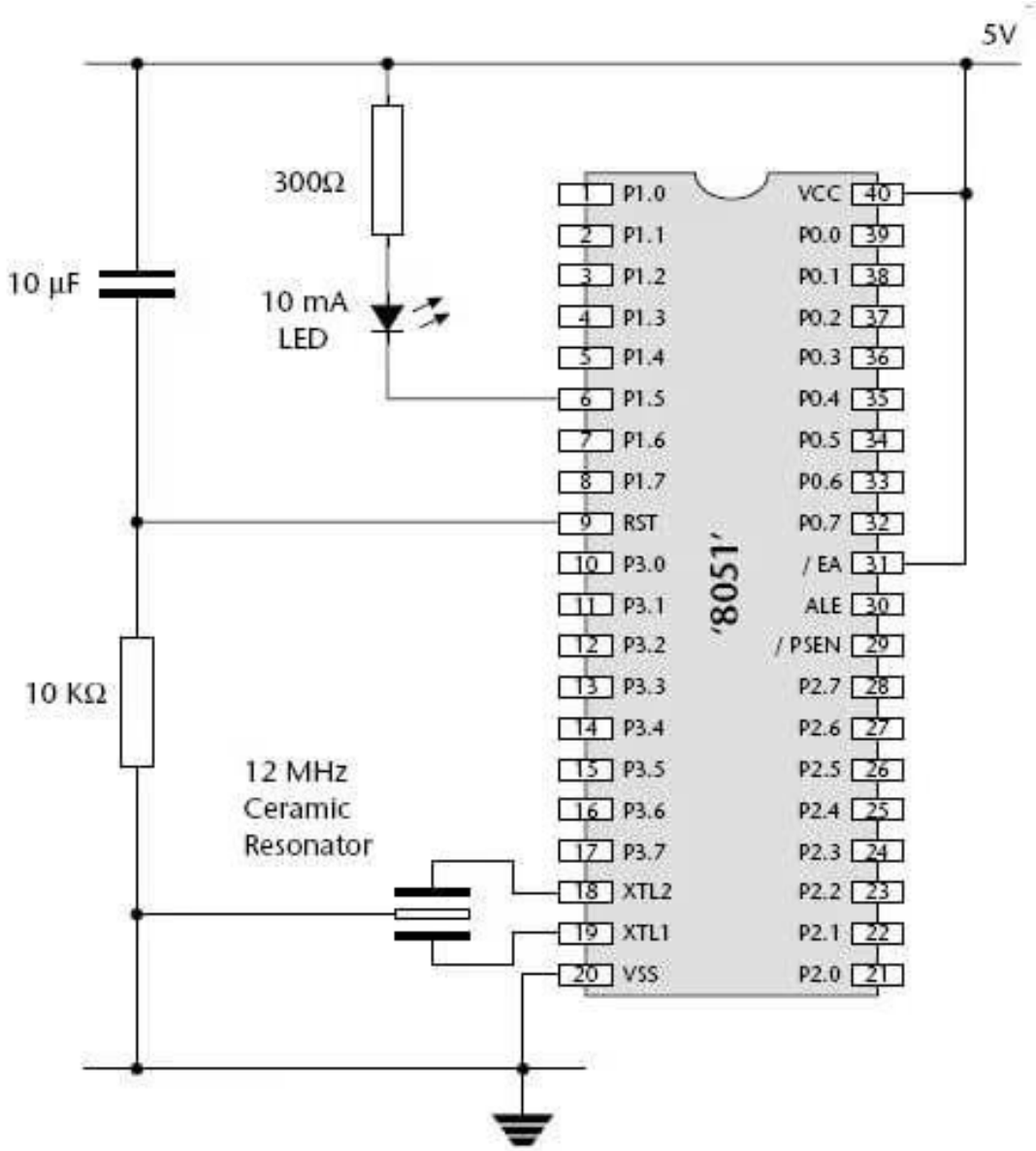
4. Timers & Counters
Two 16-bit (2-byte) timers and counters are available on an 8051 microcontroller.
The counters are further divided into an 8-bit register.
They are used in calculating the number of events, like pulse counting, frequency measurements, pulse width measurement, and more.
In addition, timers and counters are also used to determine the time period.

5. Interrupt Logic Circuitry
This logic circuit senses interrupts in an 8051 microcontroller.
It consists of interrupt-priority registers, interrupt-enable registers, and several other components.

6. Serial & Parallel I/O Ports
The 8051 controller has 4 I/O ports for connecting to external devices.
Since a controller controls various machine operations, I/O ports are required to transfer data.

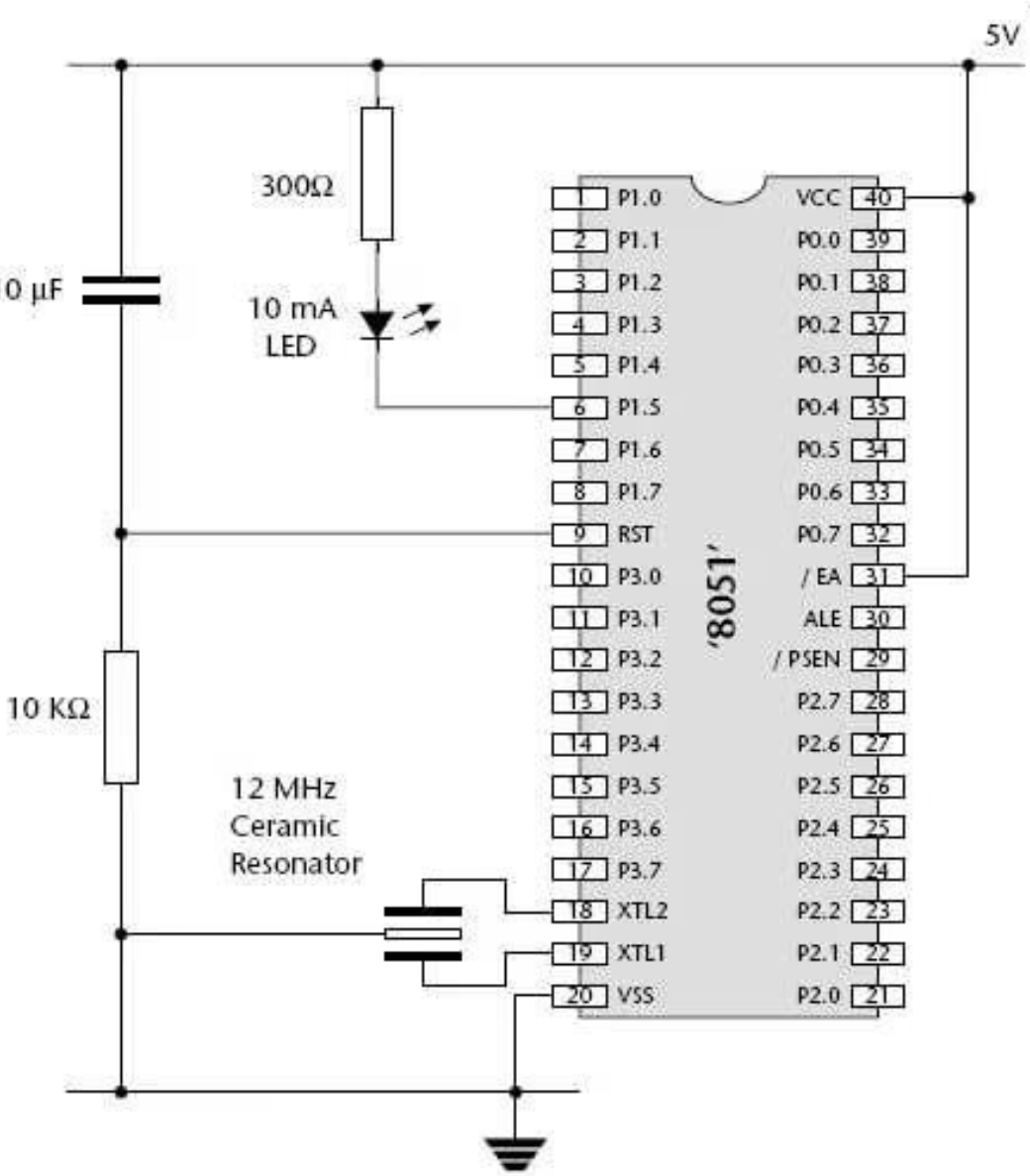
8051 – Pin diagram

There are **40 pins** in total for the 8051 microcontroller. We can use a pin diagram, a visual representation of the **pins** on an integrated circuit (IC) or microcontroller, to see their connections and functions.



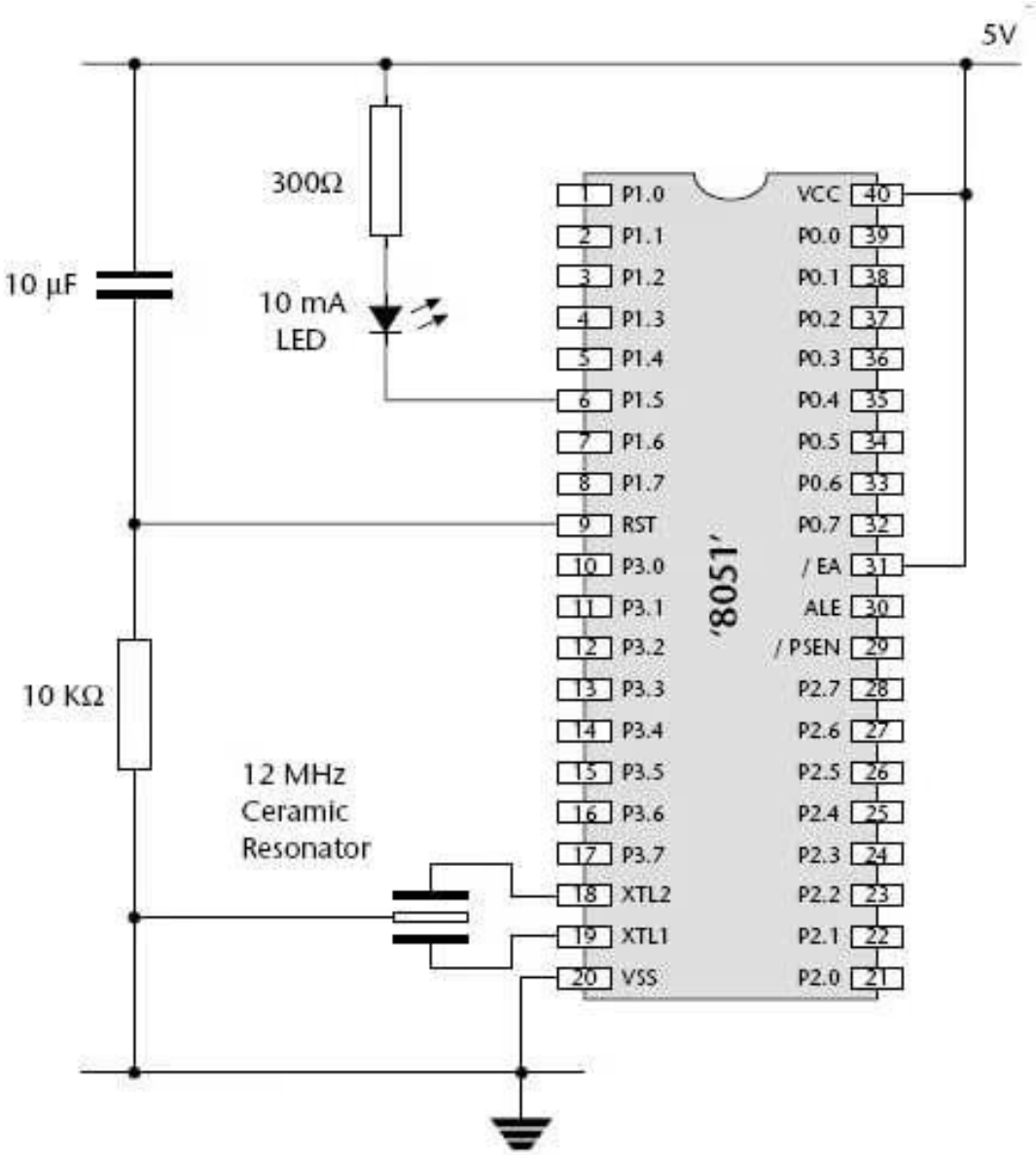
8051 – Pin diagram

Pin	Function
1 to 8	These pins combine to make Port 1. This is a bi-directional I/O port.
9	This pin sets the controller to the primary value; therefore, it is known as a Reset pin.
10 to 17	Port 3 is formed with these eight pins. This port is used for various functions, like interrupts, timer input, and serial communication indicators for transfer and receival of the data. Port 3 is also known as a domestic pull-up port.
18 and 19	These two pins are used to interface the given system clock with the outer crystal oscillator.
20	It is denoted as Vss. This pin symbolizes 0V or ground voltage of the 8051 controller.
21 to 28	Port 2 is formed with these eight pins. This port is used as an I/O port. It multiplexes the senior order address bus.



8051 – Pin diagram

Pin	Function
29	It is PSEN or a Program Store Enable. You can interpret the sign from the outer program memory with this pin of the controller.
30	It belongs to EA or External Access. It is used to prohibit or permit the interfacing of the outer memory. This pin is linked to the supply voltage to set it to high.
31	It belongs to ALE or Address Latch Enable. It is used to demultiplex the Port 0 address data indication for interfacing with the outer memory.
32 to 39	These eight pins combine to form Port 0 of the controller. This is a bi-directional I/O port that multiplexes data bus signals and lower-order addresses. To utilize this port, you need outer connected pull-up resistors.
40	With this pin, you can provide your circuit with the power supply.



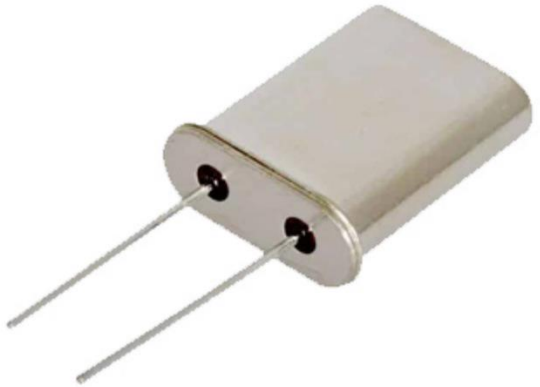
Clock source

8051 programs are run sequentially, one instruction executing after the previous one.

- A clock source is needed to increment a program counter (discussed later).
- Other functions within the microcontroller (timers, communications, etc.) need to operate synchronously as well.
- To make sure that all of these operate synchronously, the 8051 uses a master oscillator.
- The 8051's oscillator outputs a square wave. This square wave's period is determined by a relatively stable crystal (external).

Many modern processors can execute one instruction per clock cycle. (or more!)

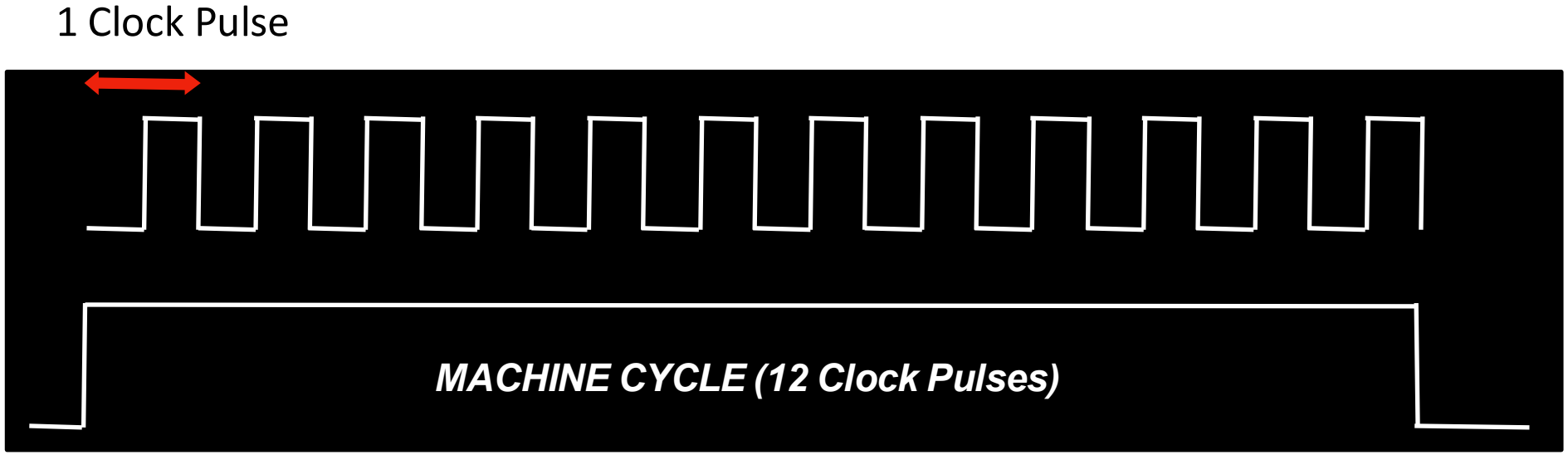
- The 8051 requires 12 clock cycles per "machine cycle".
- Instructions on the 8051 require 1 or 2 machine cycles
- Given a 12 MHz clock, this means that we can execute 0.5- 1 million instructions per second (MIPS).



OSCILLATOR HARDWARE
(internal, generates square wave from crystal's oscillation)

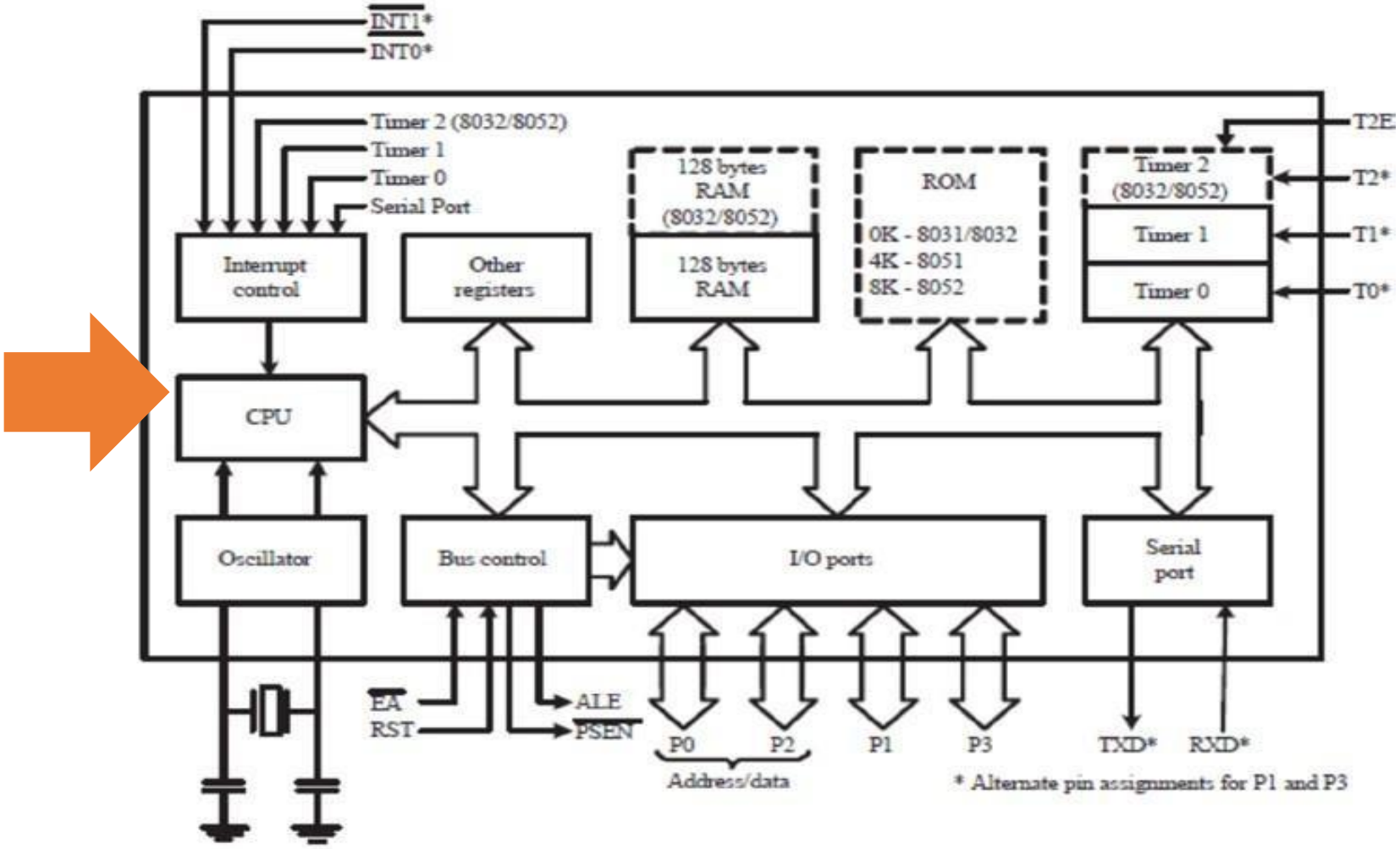
Machine Cycle

The oscillator generates *clock pulses = 1/12* machine cycle
The CPU (etc.) is sequenced by these machine cycles



Given a 40 MHz Crystal, find the time required for one machine cycle.
40 MHz / 12 = 3.33 MHz
1/3.33 MHz = 0.30 µs

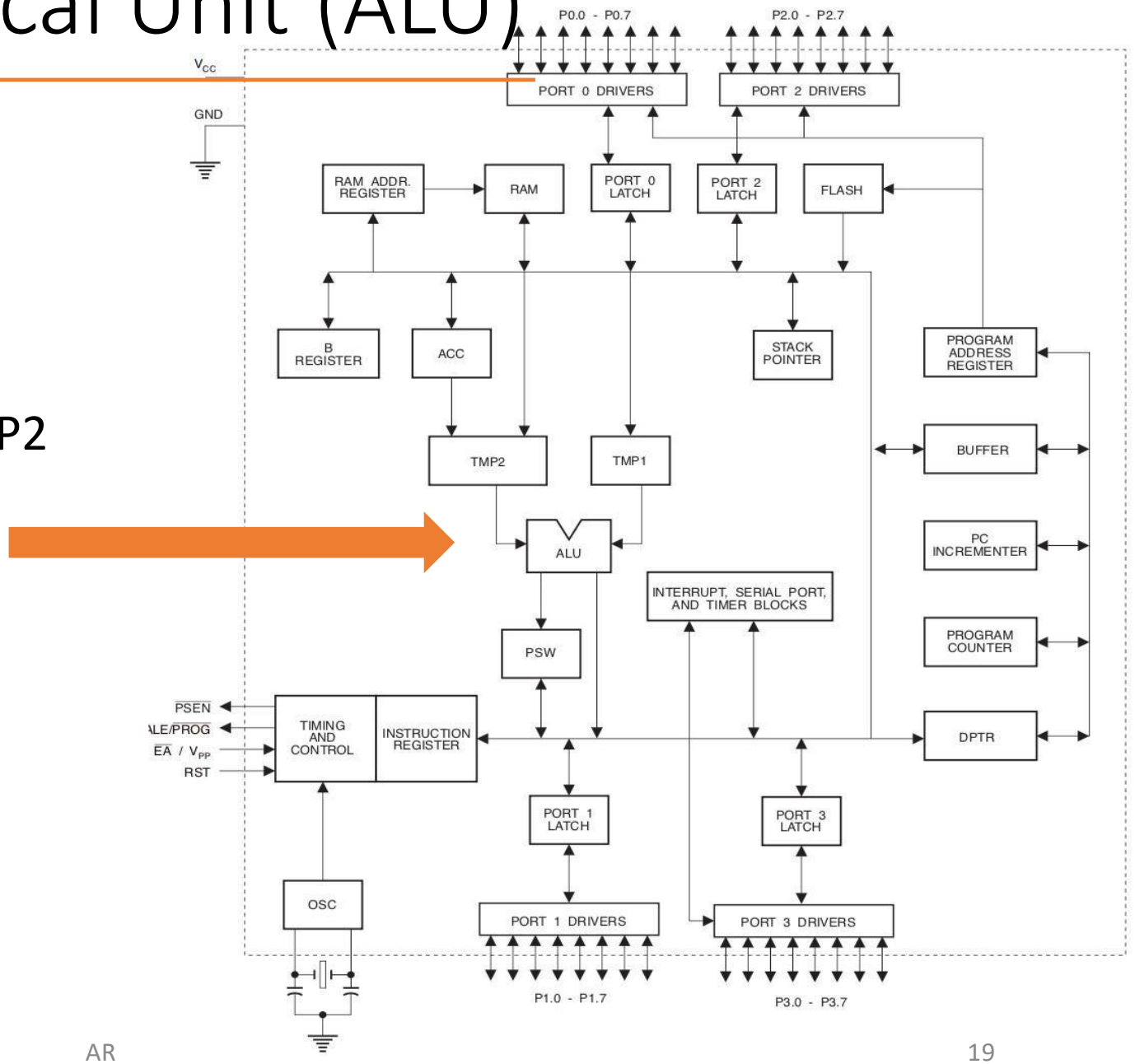
8051 Arithmetic Logical Unit (ALU)



8051 Arithmetic Logical Unit (ALU)

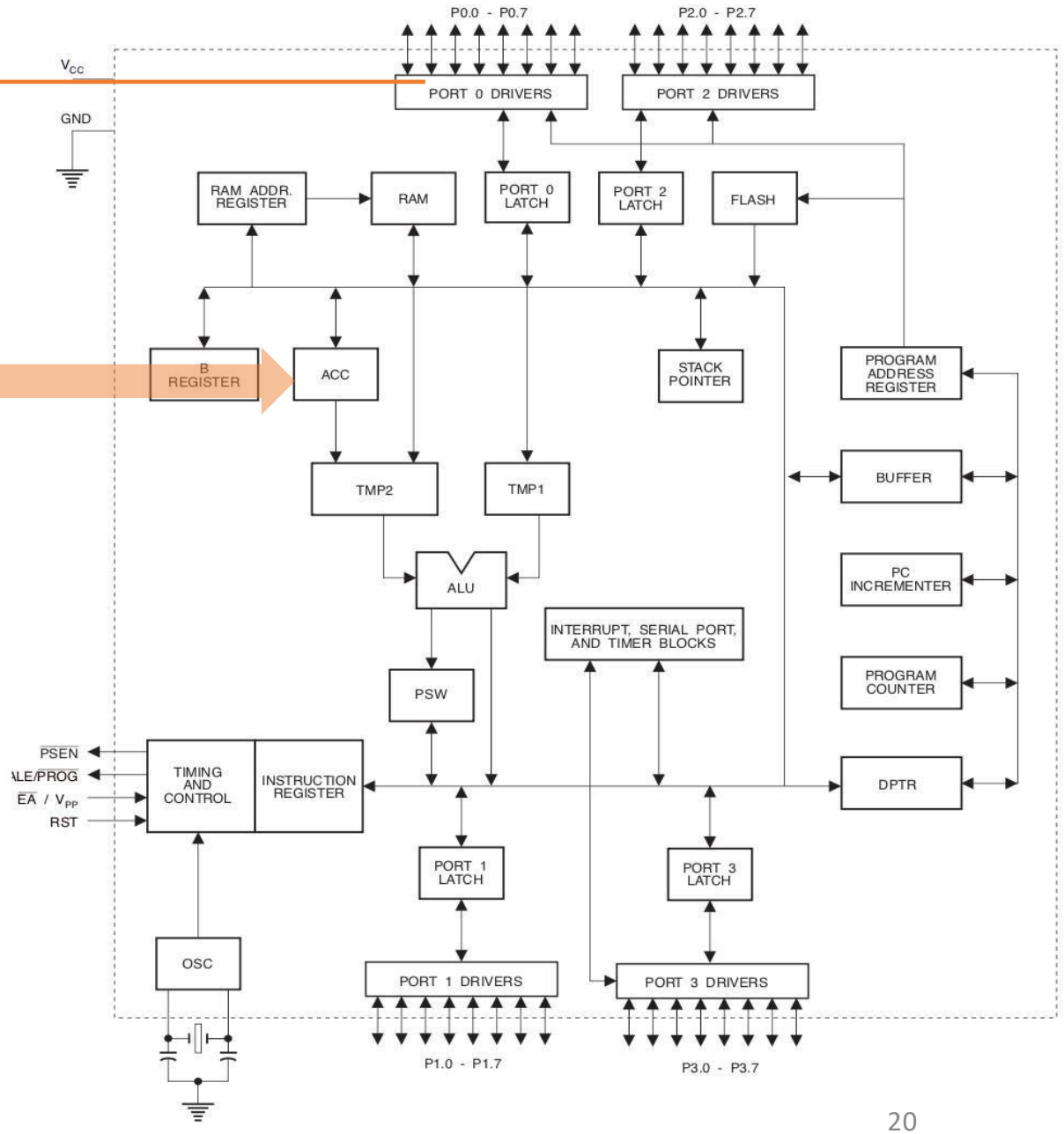
ALU: Arithmetic Logic Unit
Processes the values of TMP1 and TMP2 registers.

Register: An accessible amount of storage that may be quickly acted upon by the CPU



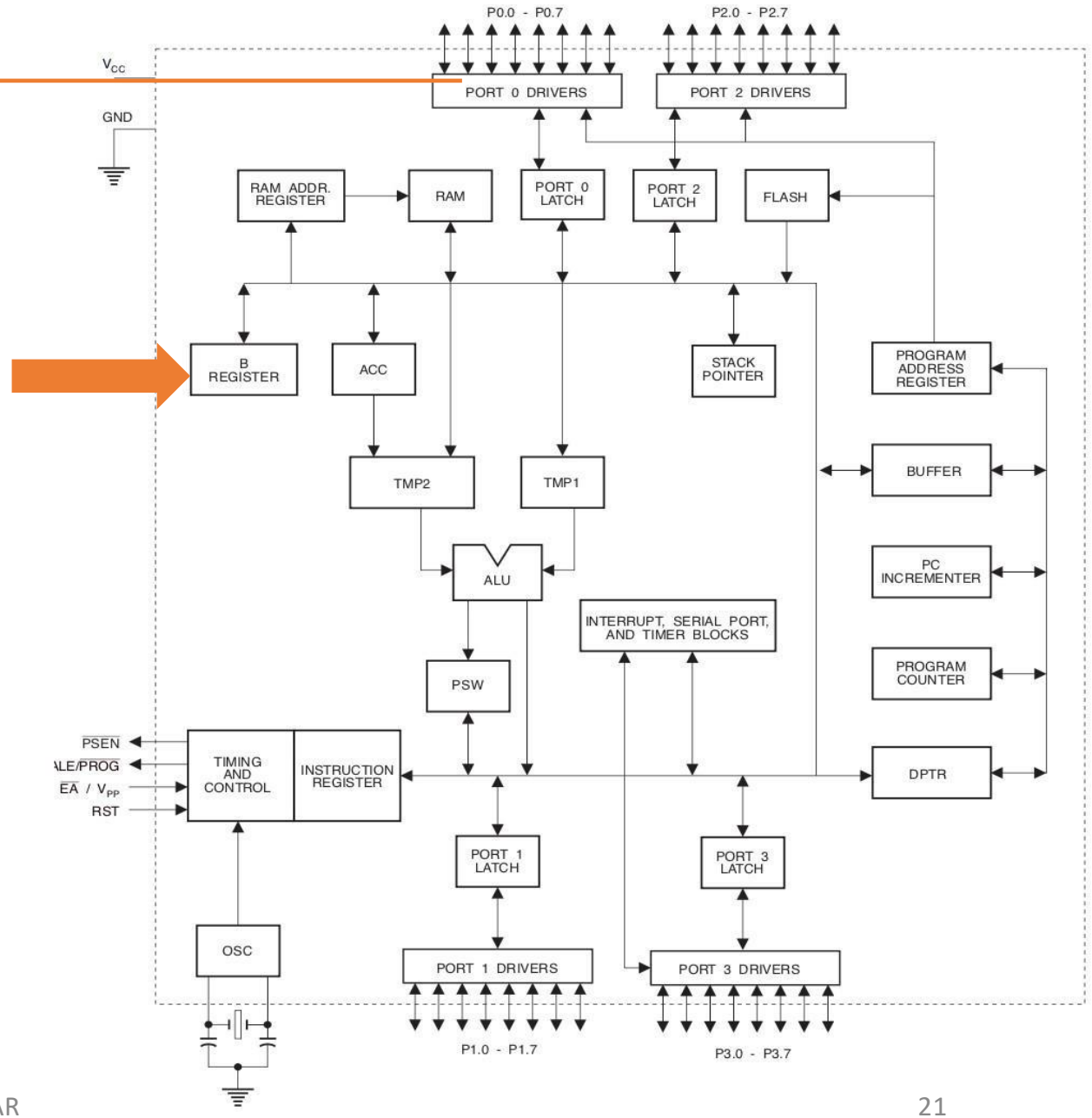
8051 Accumulator (ACC)

- Often referred to as Register A
- Results of ALU operations are stored in this register.
- 8-bit (1 byte)
- Many instructions in the 8051 use the A register.
- Values in the accumulator may be further processed (e.g., sent back to the ALU).
- The accumulator value may also be stored in other memory locations.



8051 B Register

- Like the A register (ACC), the B Register is 8-bit.
- Used by only two instructions
- (MUL AB and DIV AB)
- Multiply the values of the A and B registers
- Divide the values of the A and B registers
- The B register may also be used as a more general-purpose storage register.



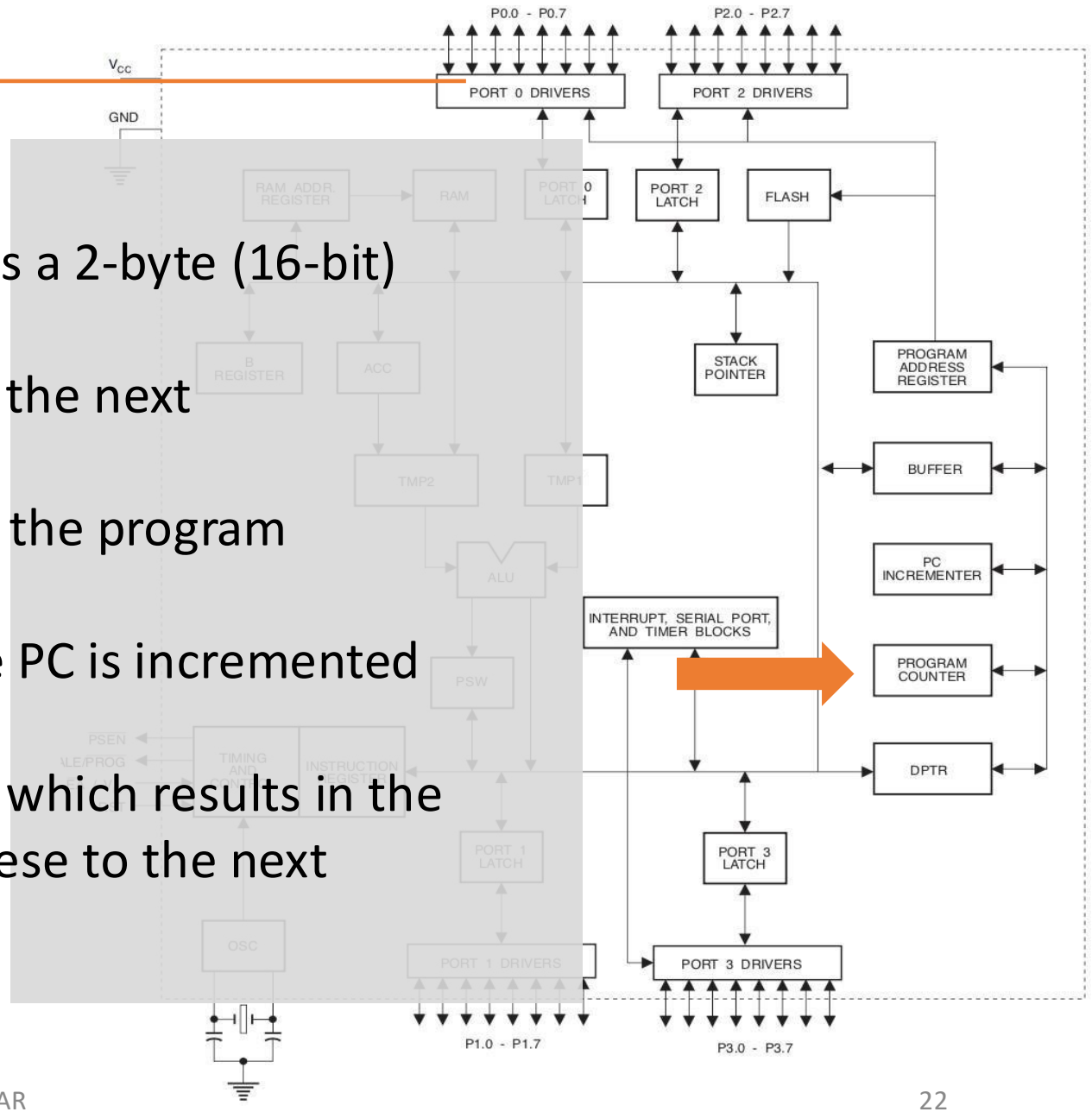
8051 Program Counter

Unlike previously discussed registers, this is a 2-byte (16-bit)

- 0000 0000 0000 0000

The program counter holds the address of the next instruction to be executed.

- When the 8051 is booted up, by default, the program counter is set to 0x 00 00
- If an instruction makes use of 1 byte, the PC is incremented by 1.
- Some instructions require 2 (or 3) bytes, which results in the PC incrementing by 2 or 3 to move past these to the next instruction.

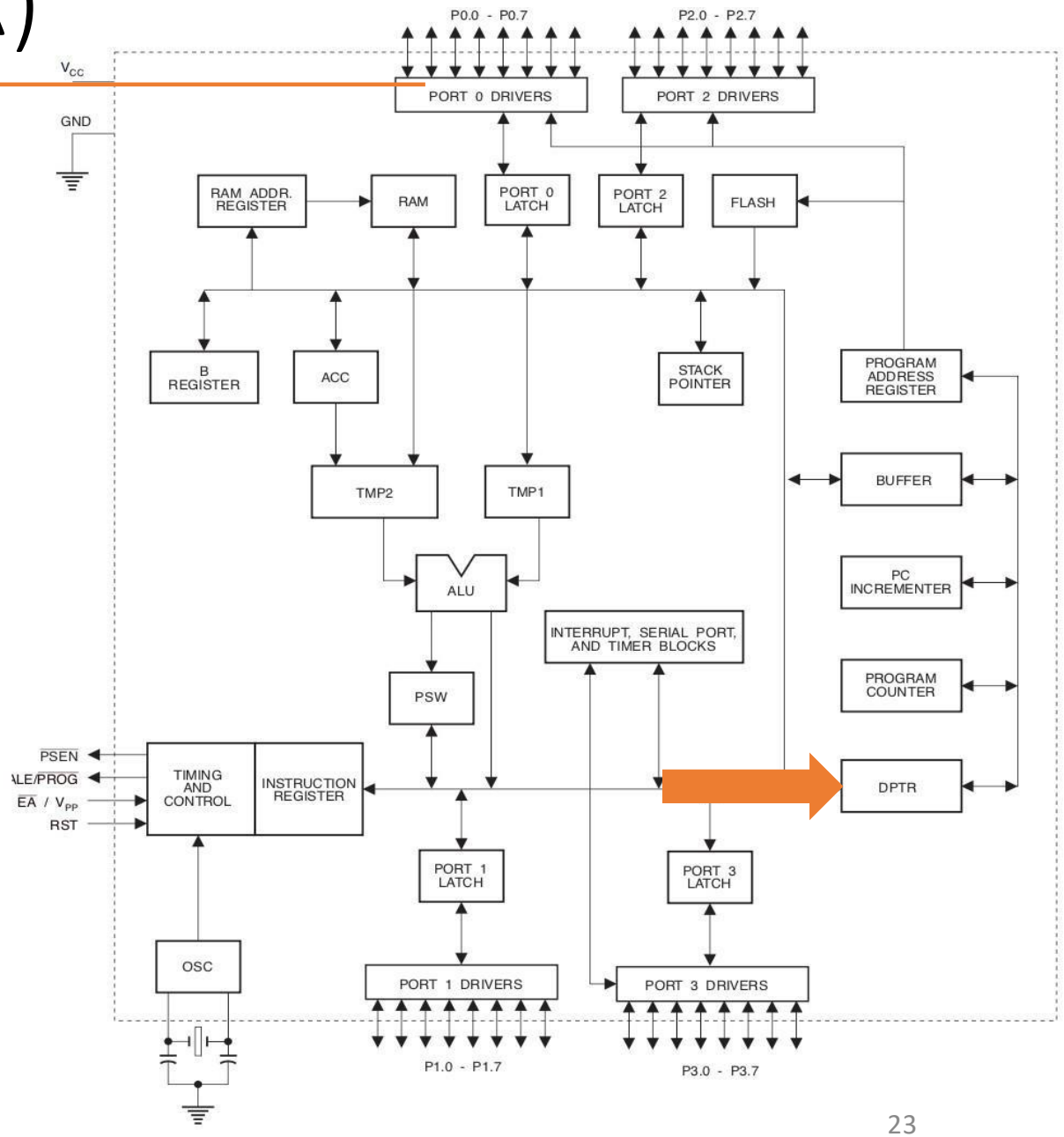


8051 Data Pointer (DPTR)

User-accessible 2-byte register

Typically used to access external memory.

- 2^16 bits of memory may be addressed.
- The DPTR may also be used to store 2-byte values conveniently.

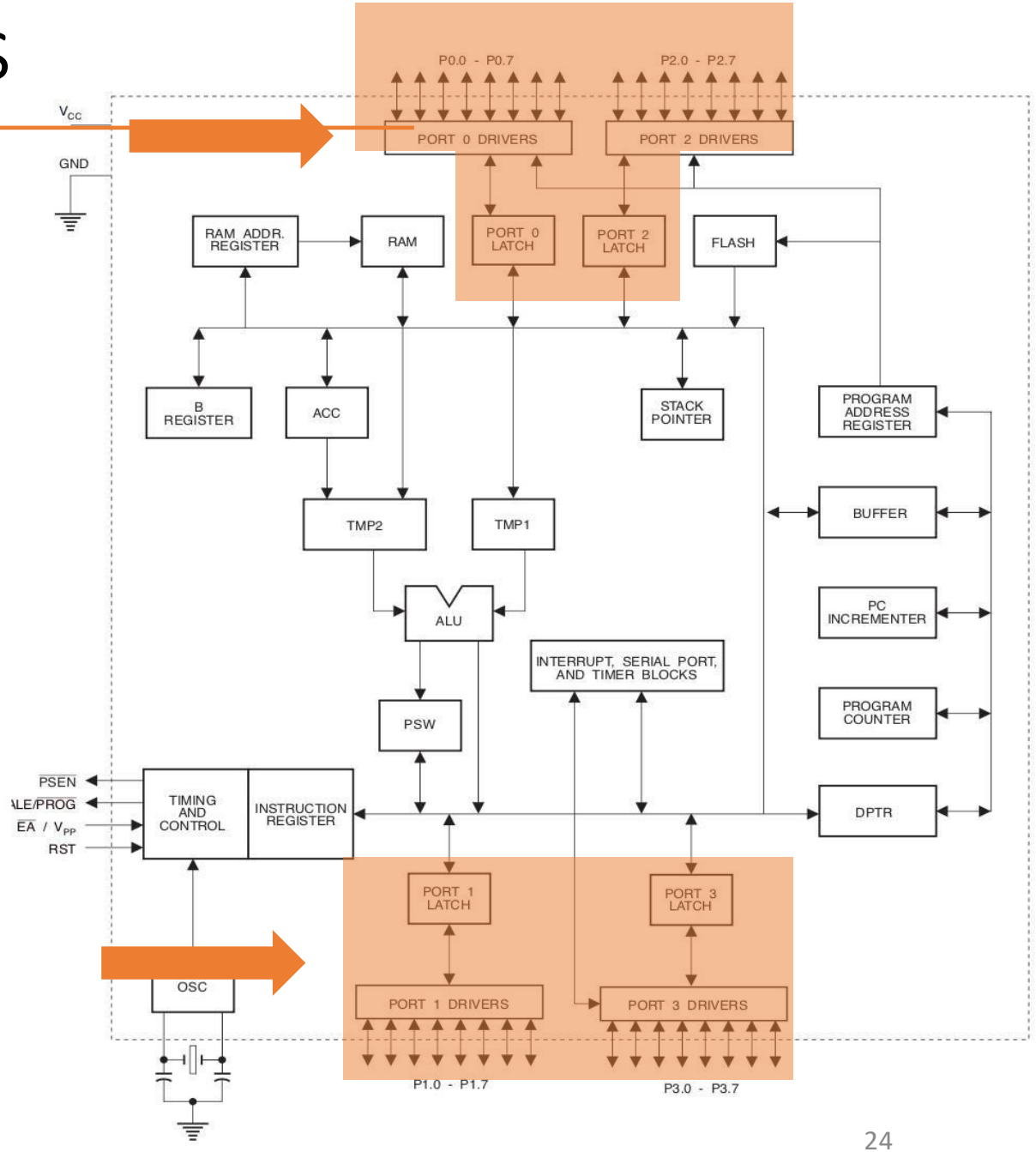


8051 Input/Output Ports

Microcontrollers typically feature numerous I/O ports.

The 8051 features 4 8-bit I/O ports.

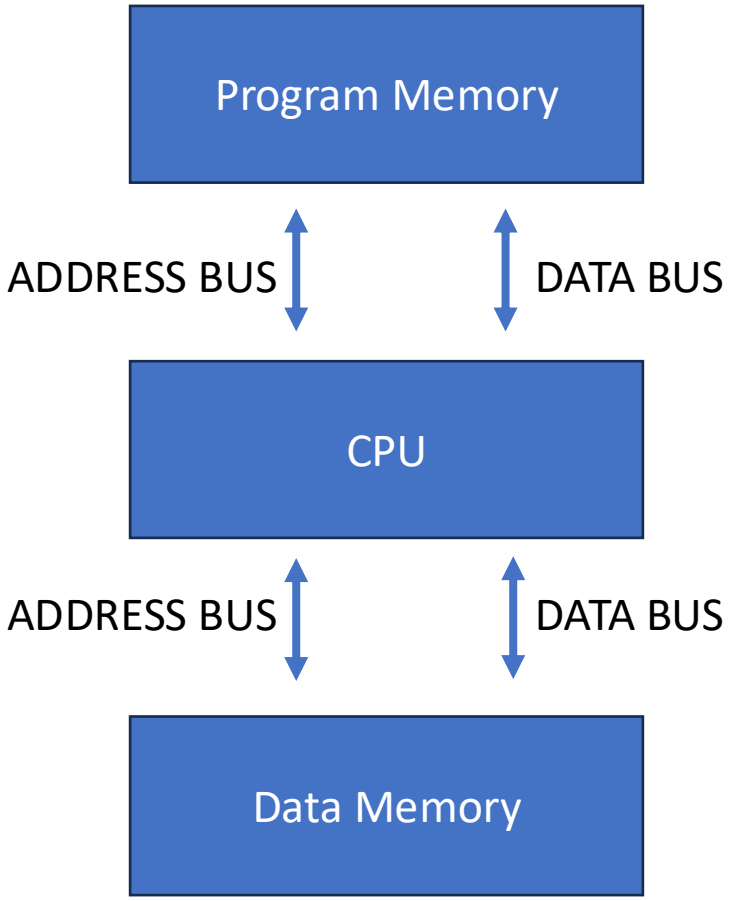
- P0, P1, P2, P3
- These pins default to input; you must specify if you wish them to be set up as outputs.
- This is done by simply writing to the port.



Memory: RAM & Storage

Computers with a Harvard Architecture have a separate program and data memories.

- Microcontrollers have a ‘volatile’ data memory.
- RAM loses state when the system resets.
- They have a non-volatile program memory.
- Retains state in power-off conditions.
- Historically, this was some form of ROM (read-only memory), originally programmable only once.



Memory: RAM & Storage

Modern microcontrollers (including the C8051F020) use flash memory for program memory.

- Flash memory may be reprogrammed a relatively large number of times, but not during program execution.
- Program memory is often embedded in the microcontroller, but may also consist of external memory modules.

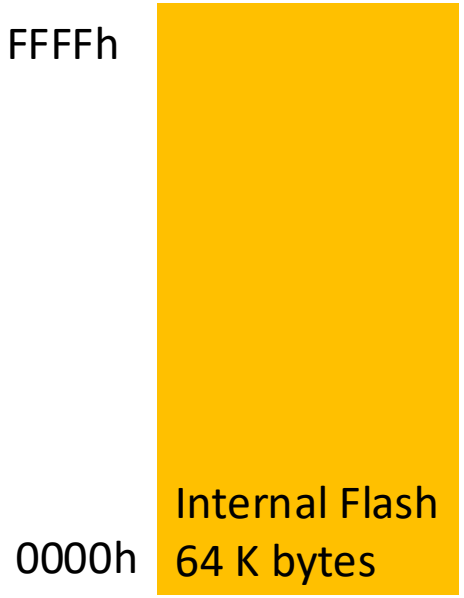
ROM and Flash: Program Memory

The C8051F020 has 64 KB of internal flash.

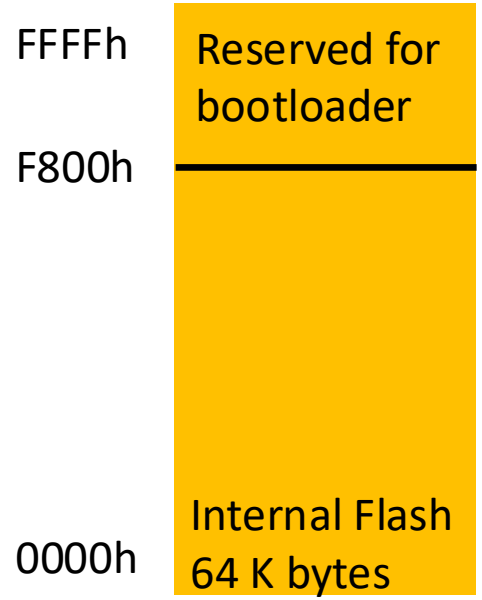
- See page 24 of the data sheet (C8051F02X.pdf) for *much* more information.

While most programs are stored in this in-system programmable flash…

- …the C8051F020 has 2KBytes of EEPROM
- The EEPROM may be edited programmatically, and is sometimes used to store variables that need to be retained after a reboot cycle.



If we wish to use the Flash to hold a bootloader (discussed in the next slide!), Then addresses F800h and FFFFh are reserved for the bootloader



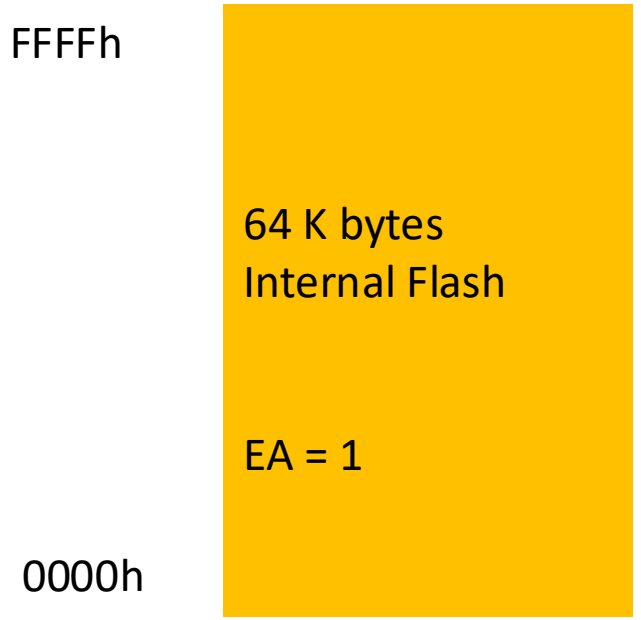
Flash Program Memory

FLASH memory:

- Can be reprogrammed in-circuit.
- Provides non-volatile data storage.
- Allows field upgrades of the 8051 firmware.

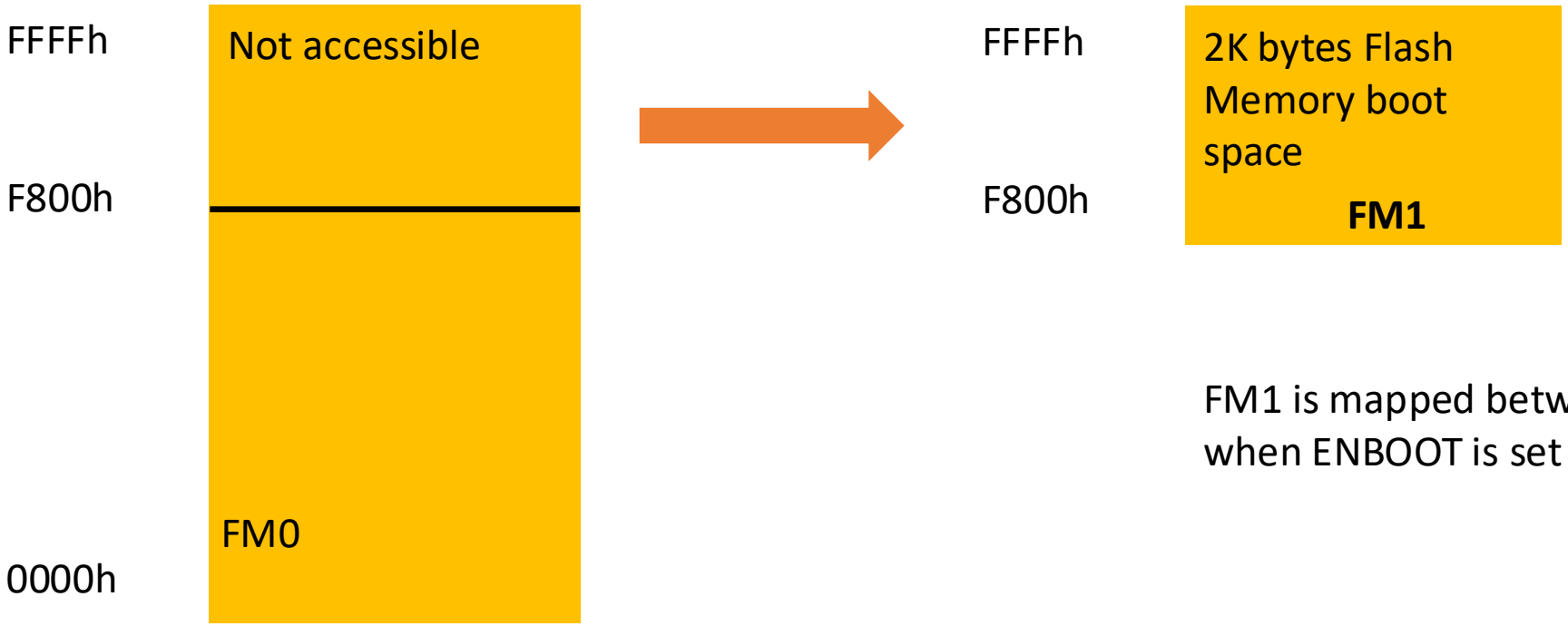
Register EA needs to be assigned as 1.

The AT89C51AC3's program memory consists of 65,536 bytes of FLASH.



Flash Program Memory

FLASH Memory Architecture with ENBOOT = 1 (boot mode)



FM1 is mapped between FFFFh and F800h when ENBOOT is set in the AUXR1 register

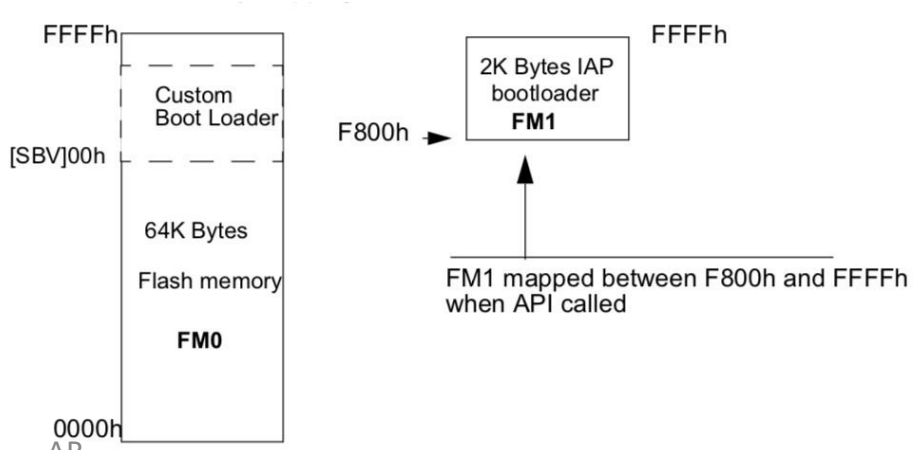
Flash-Based Bootloader

Early microcontrollers (and some contemporary basic/specialised ones) were programmed using custom programmers.

These required removing the microcontroller (or its data ROM) from the circuit and programming it with high voltages.

Contemporary microcontrollers can be programmed ‘in-system,’ allowing for simple, rapid development and iteration/revision of firmware.

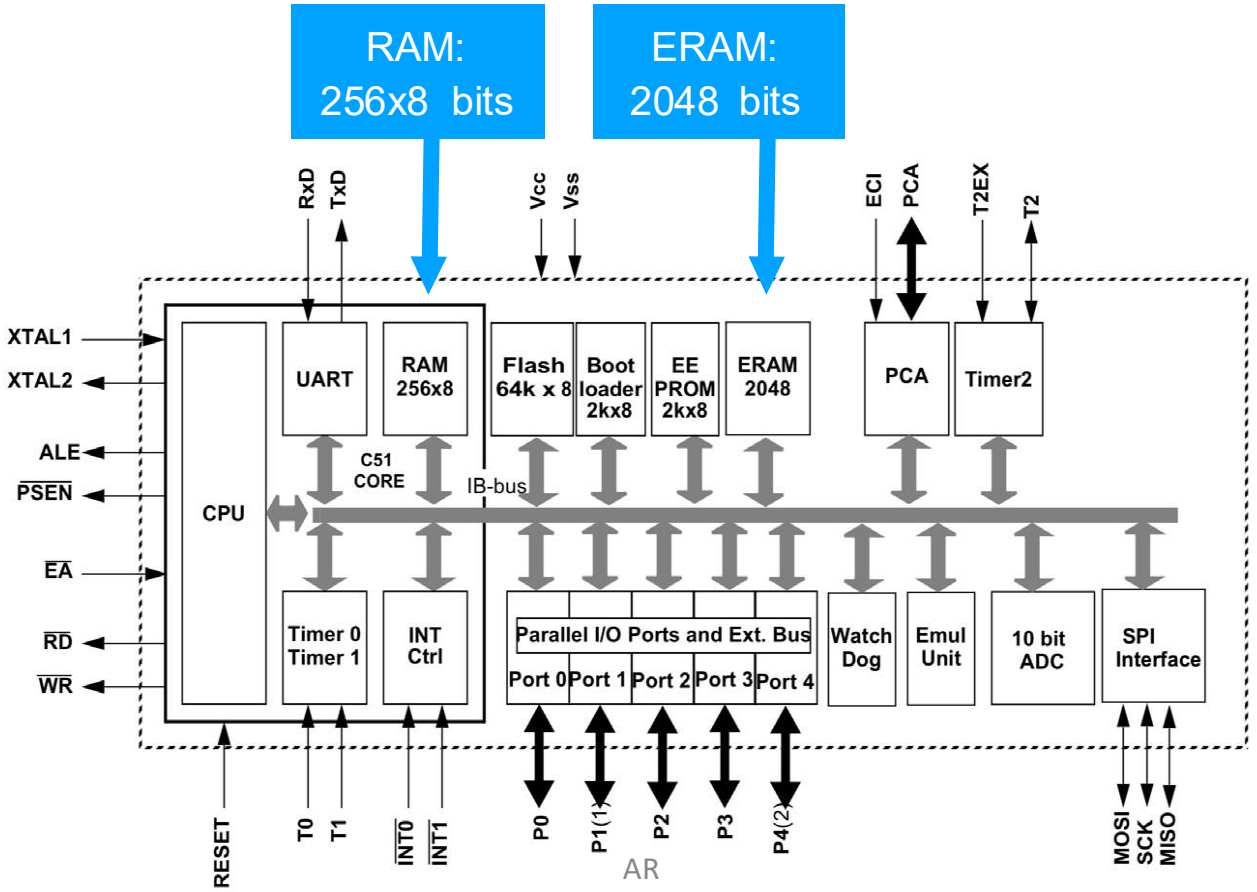
As flash memory requires some specific steps to be programmed, a specific ‘serial bootloader’ may be used to allow the flash to be programmed in-system via the microcontroller’s serial port.



Volatile Memory: RAM

Originally, the 8051 had 128 Bytes of volatile RAM.

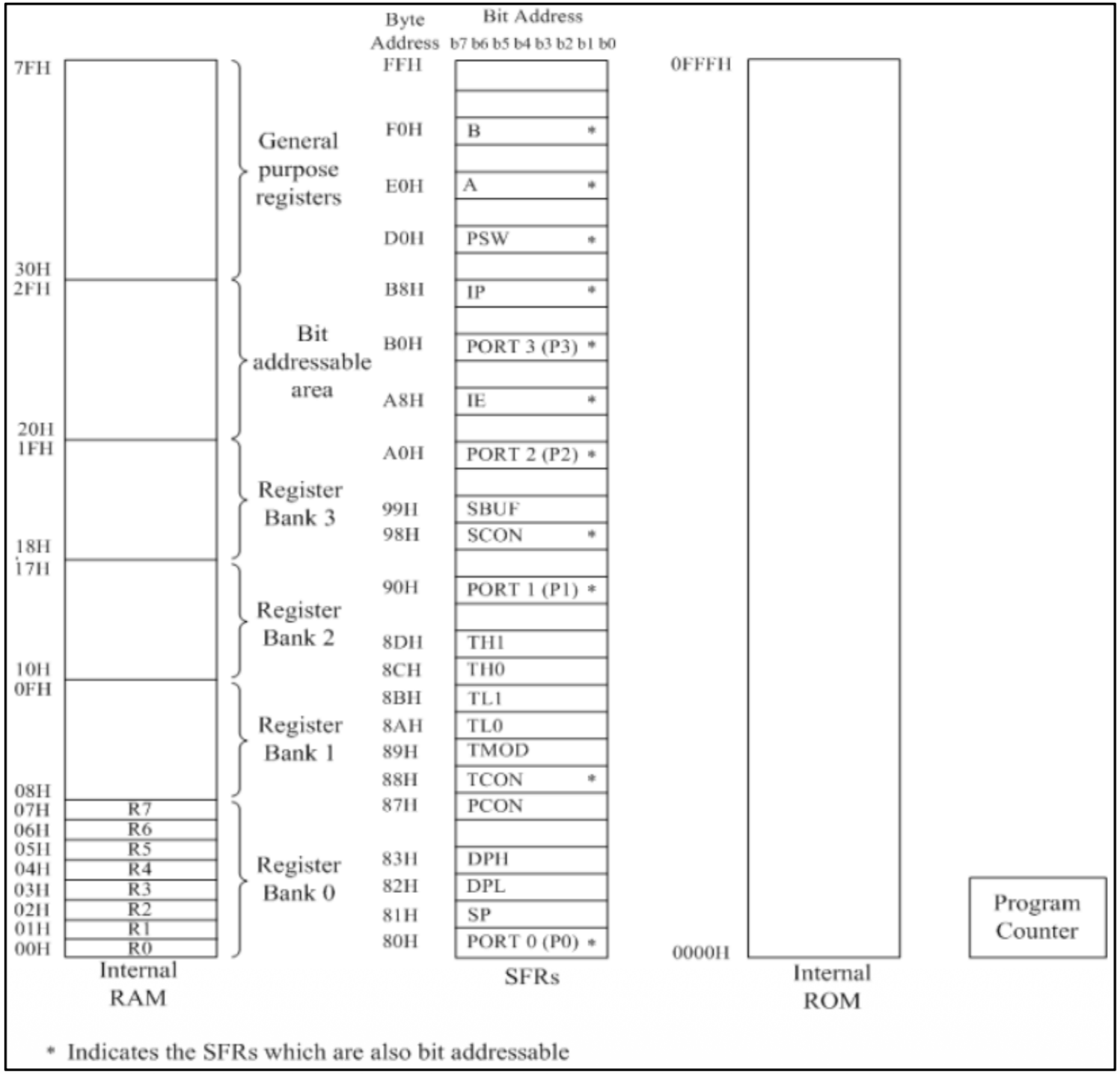
- The AT89C51AC3 has a whopping 256 Bytes alongside 2 KBytes of additional RAM (called the “expanded RAM segment”, ERAM).
- This RAM is subdivided into a number of blocks, some general purpose and some with very specific functions



The AT89C51 Microcontroller core

The programming model shows all the registers that are available to the software developer.

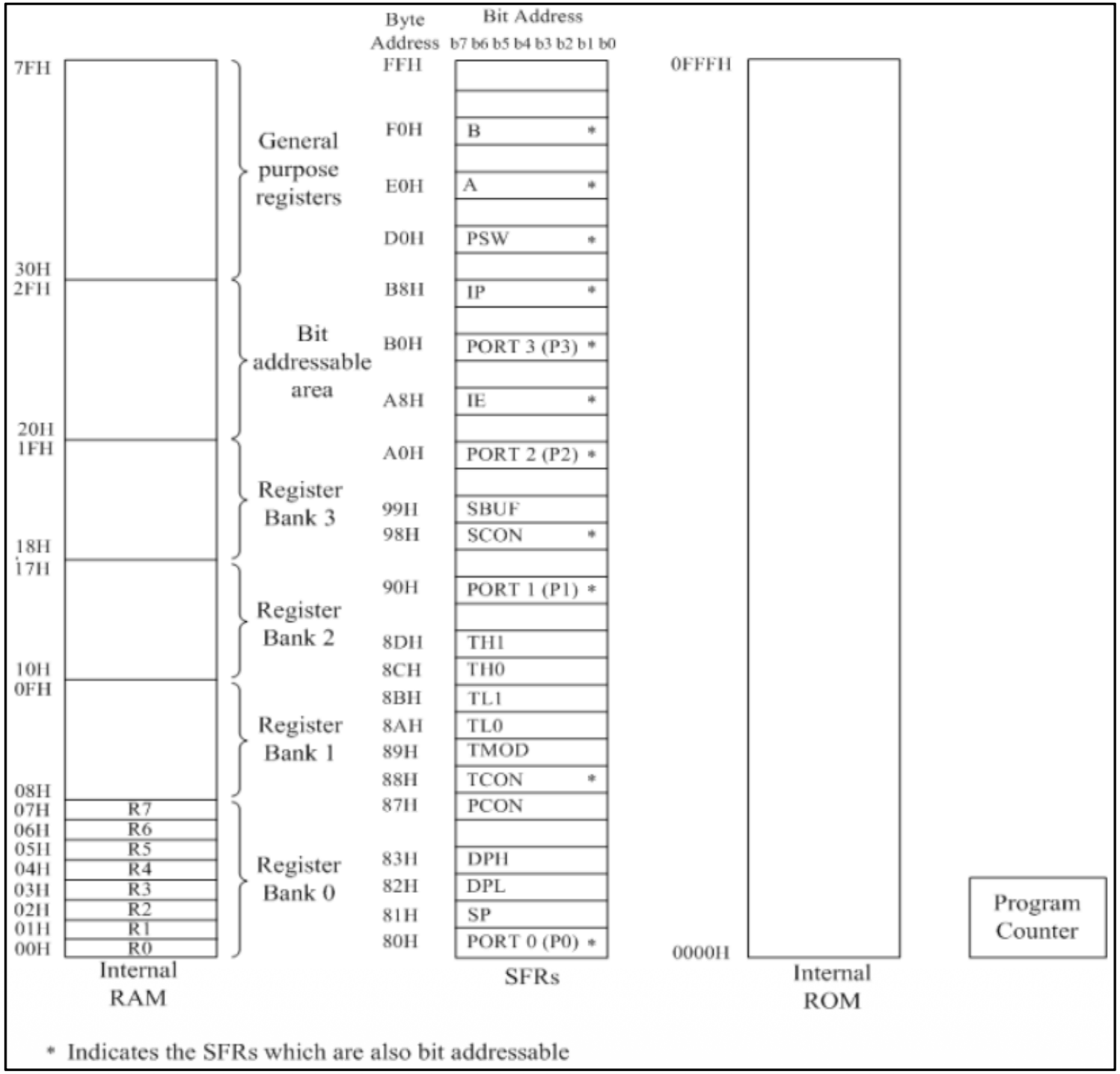
Notice that all registers are assigned addresses, except the program counter.



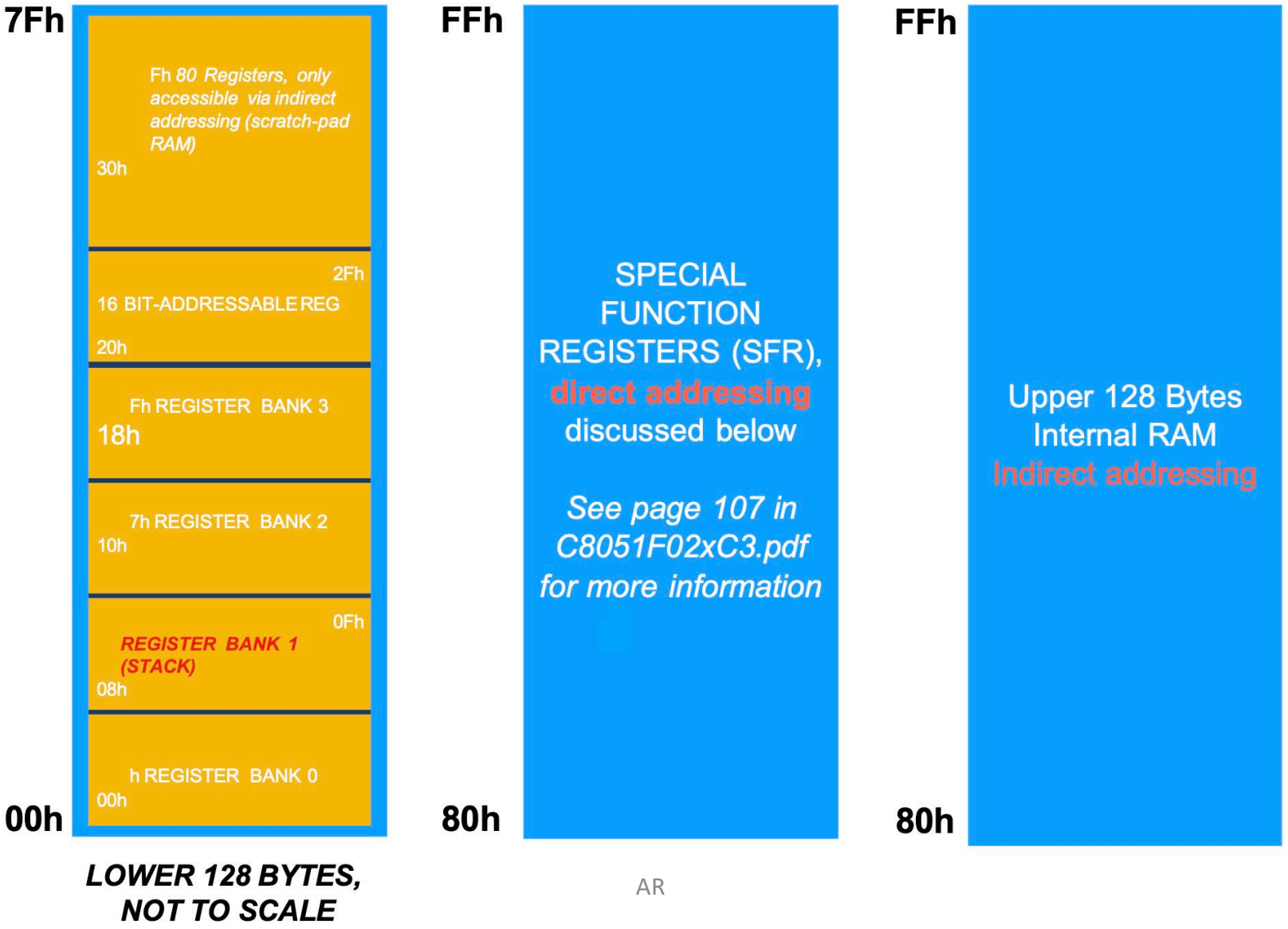
The AT89C51 Microcontroller core

The programming model shows all the registers that are available to the software developer.

Notice that all registers are assigned addresses, except the program counter.



8051 Data Memory Map

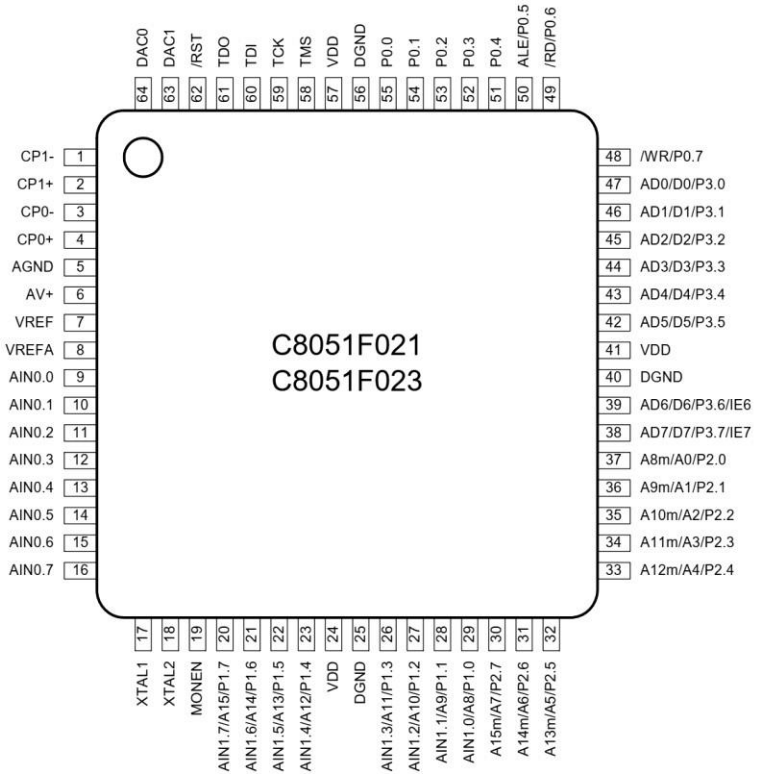


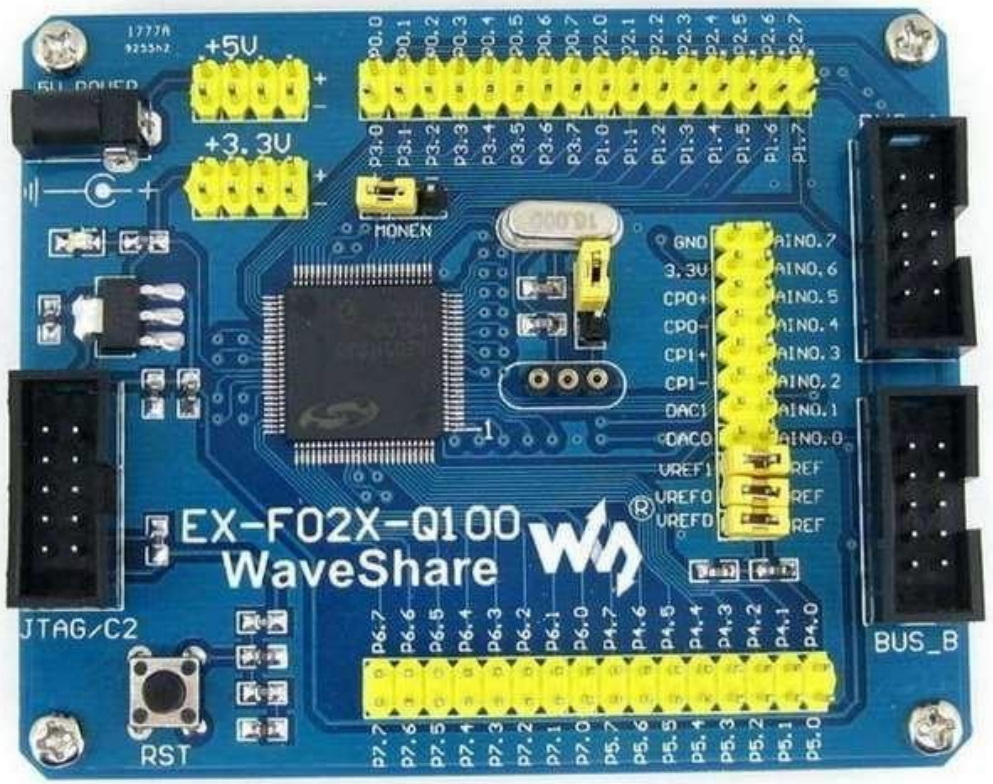
8051 Addressing Modes

A key part of computer operation involves accessing memory; this may be done on the 8051 using 5 main approaches:

IMMEDIATE ADDRESSING MODE	REGISTER ADDRESSING MODE	DIRECT ADDRESSING MODE	INDIRECT ADDRESSING MODE	INDEXED ADDRESSING MODE
• The data is included in the instruction. • MOV A,#48H • The # shows that the data is 'immediate' • In a sense, this data is hard- coded into the instruction. Fast but less flexible.	• The data operand is in a specified register. • Only some registers may be used: R0 through R7 of each of the 8051's banks. • MOV A,R7 • Contents of R7 are copied to ACC.	• The address of a location in RAM is specified, and its contents are operated upon. Only works with internal RAM & SFR's • MOV A,10H • Contents of address are copied to ACC.	• Slower: the contents of a location of the address stored in a register are fetched. • MOV A,@R7 • The @ indicates an address • The upper 128 bytes of RAM are accessible this way.	• Used to step through data (as in lookup tables). • We won't be exploring this in depth (and you won't be tested on it!), but see details about the MOVC instruction in C8051F02xC3.pdf

8051 Development kits

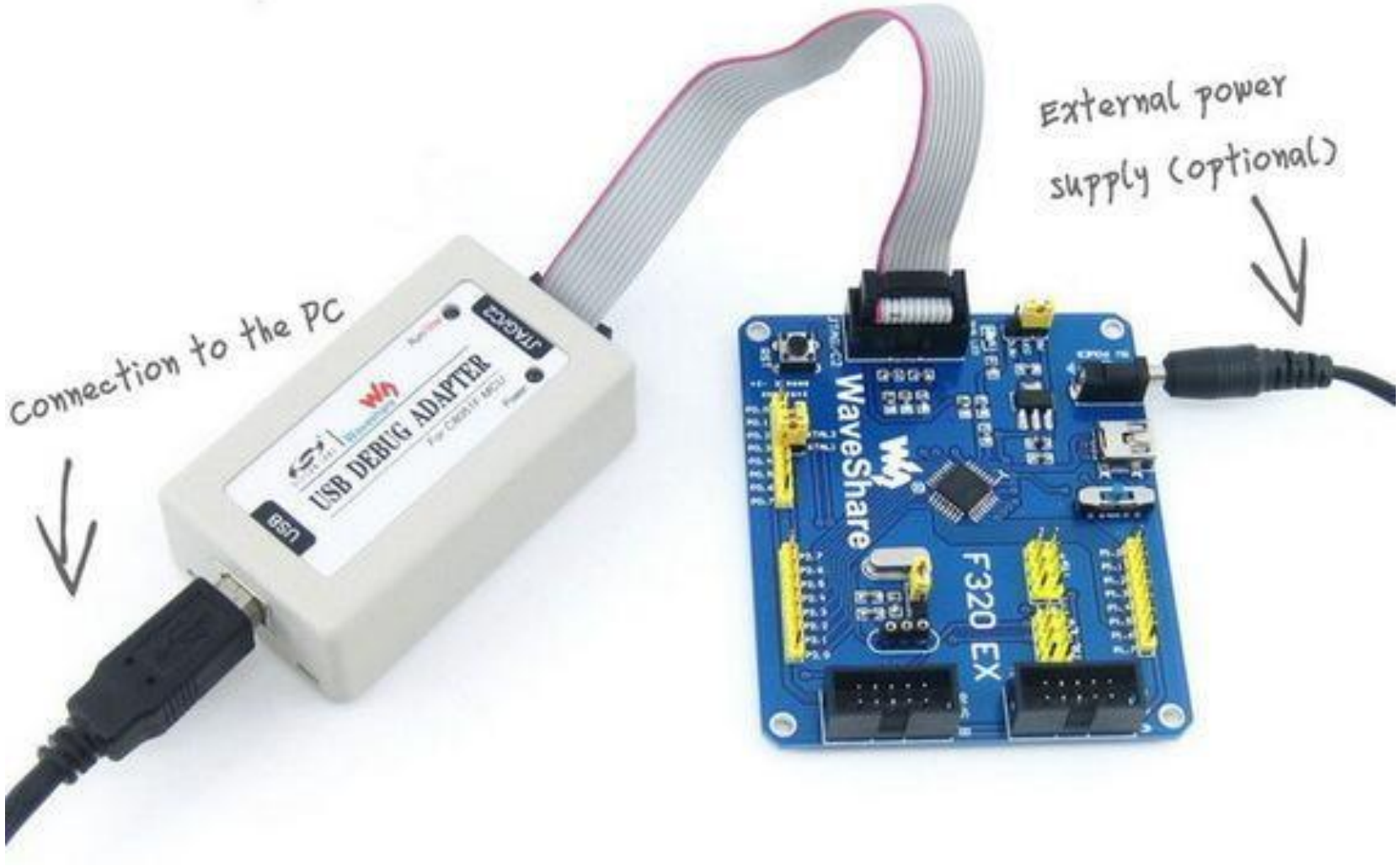




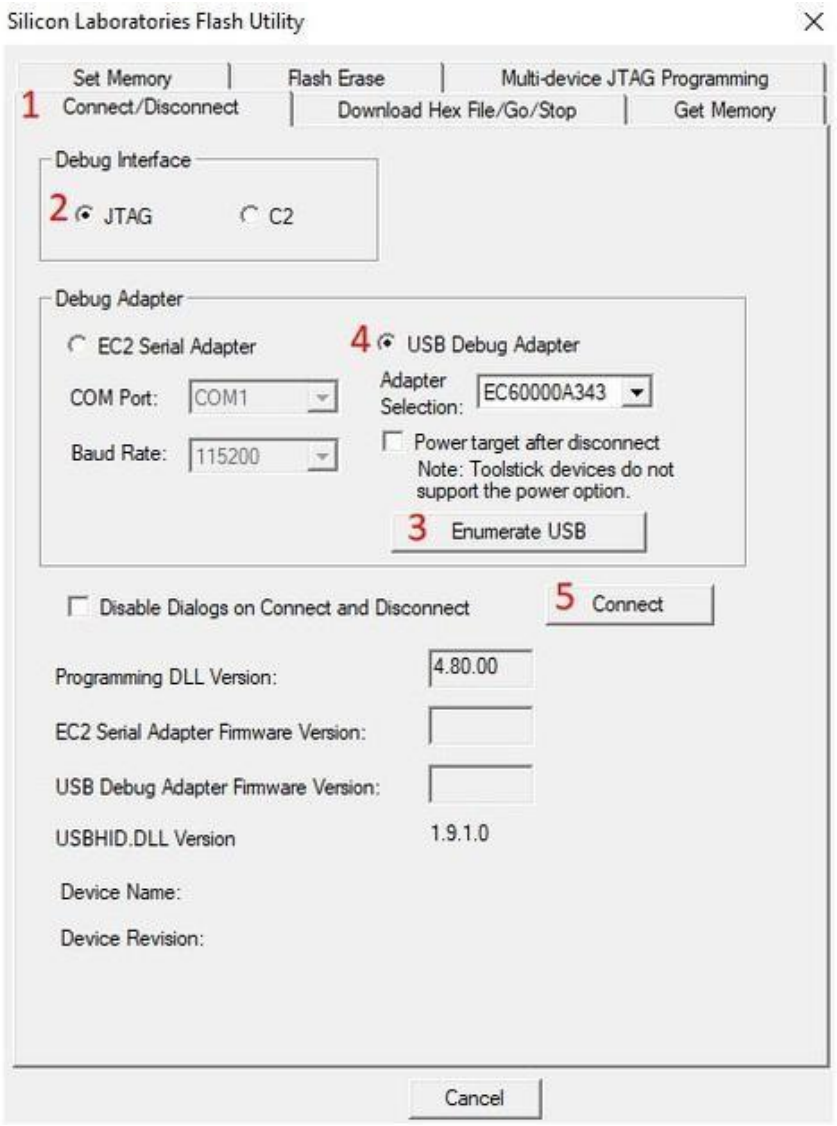
Microcontroller kit: CPU, crystal clock, setting jumpers, extension connectors, and reset button.

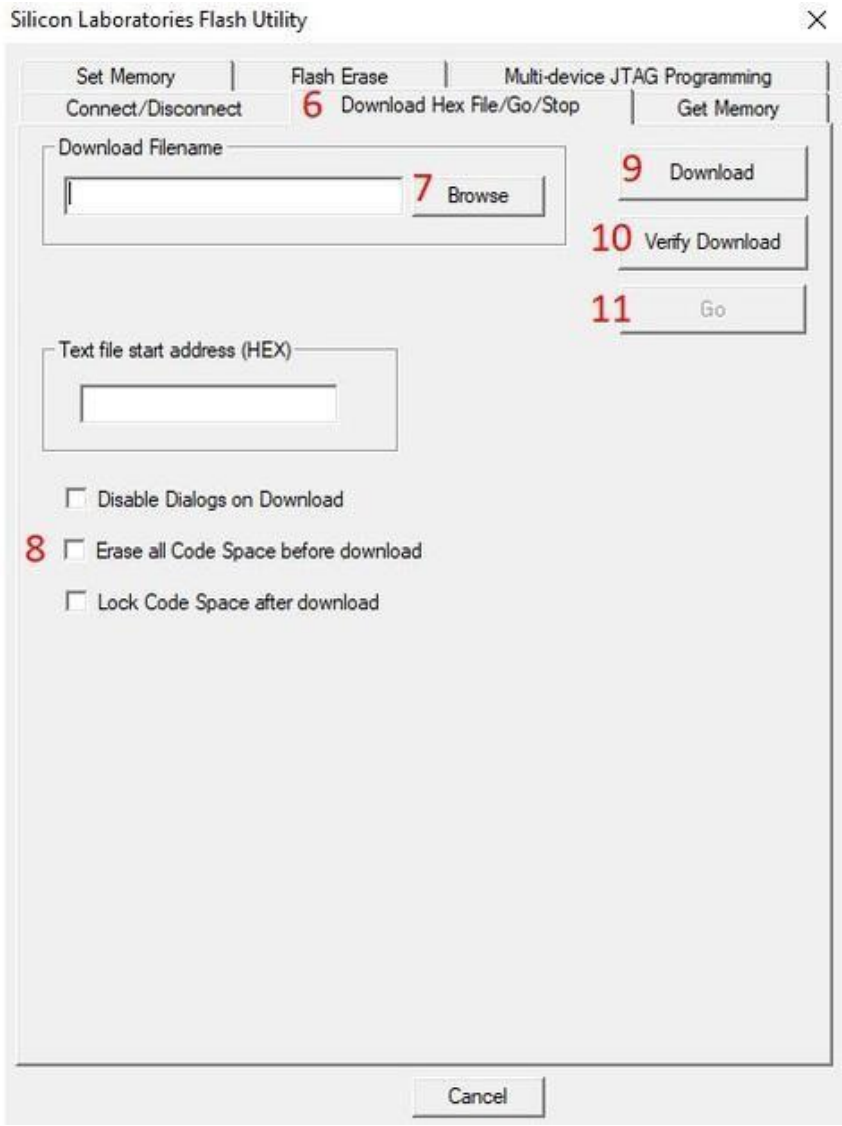
CPU: Power/GND, crystal oscillator, external clock/timer, instruction bus, address bus, data bus, interrupts, etc.

8051 Development kits

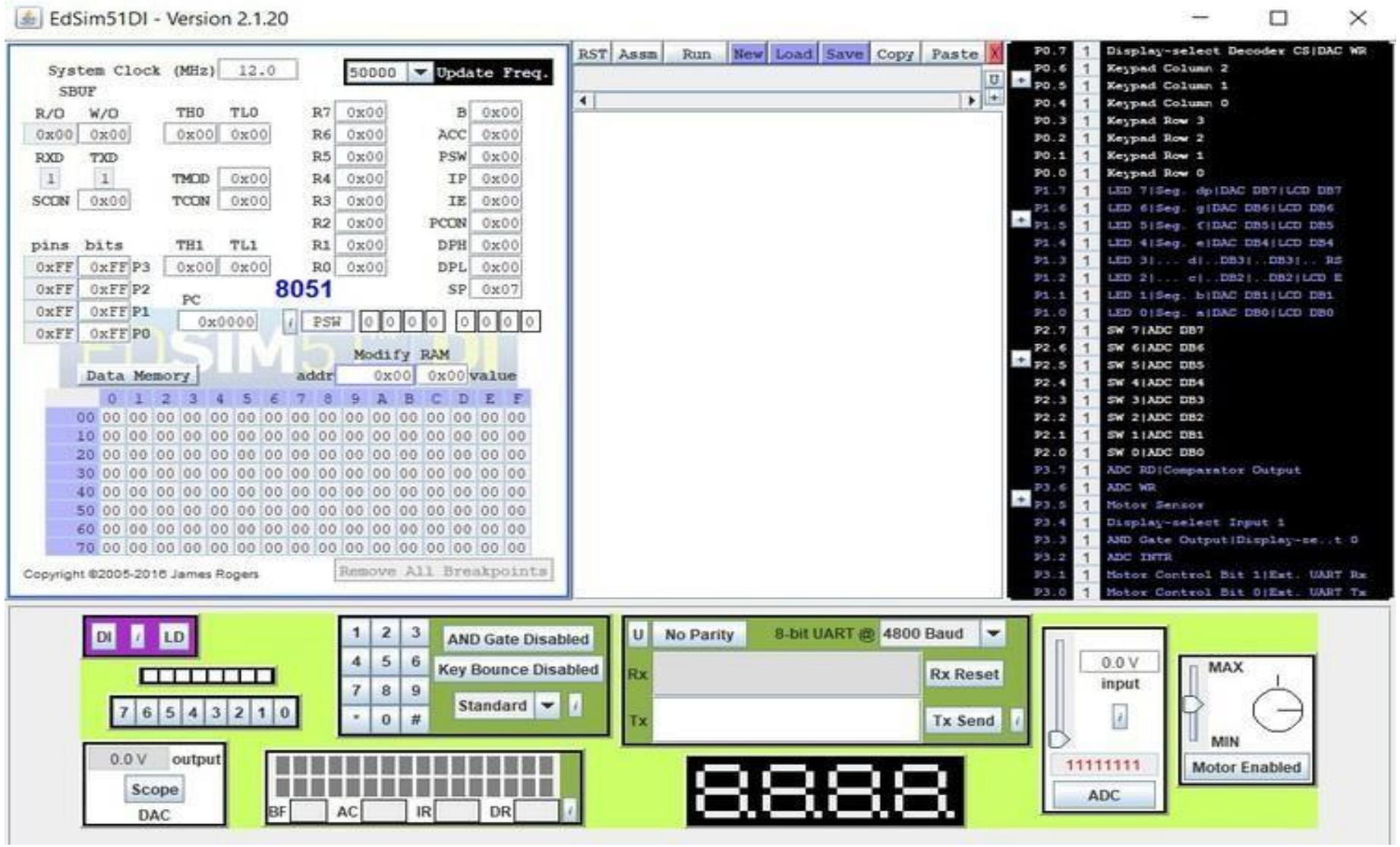


8051 Development kits

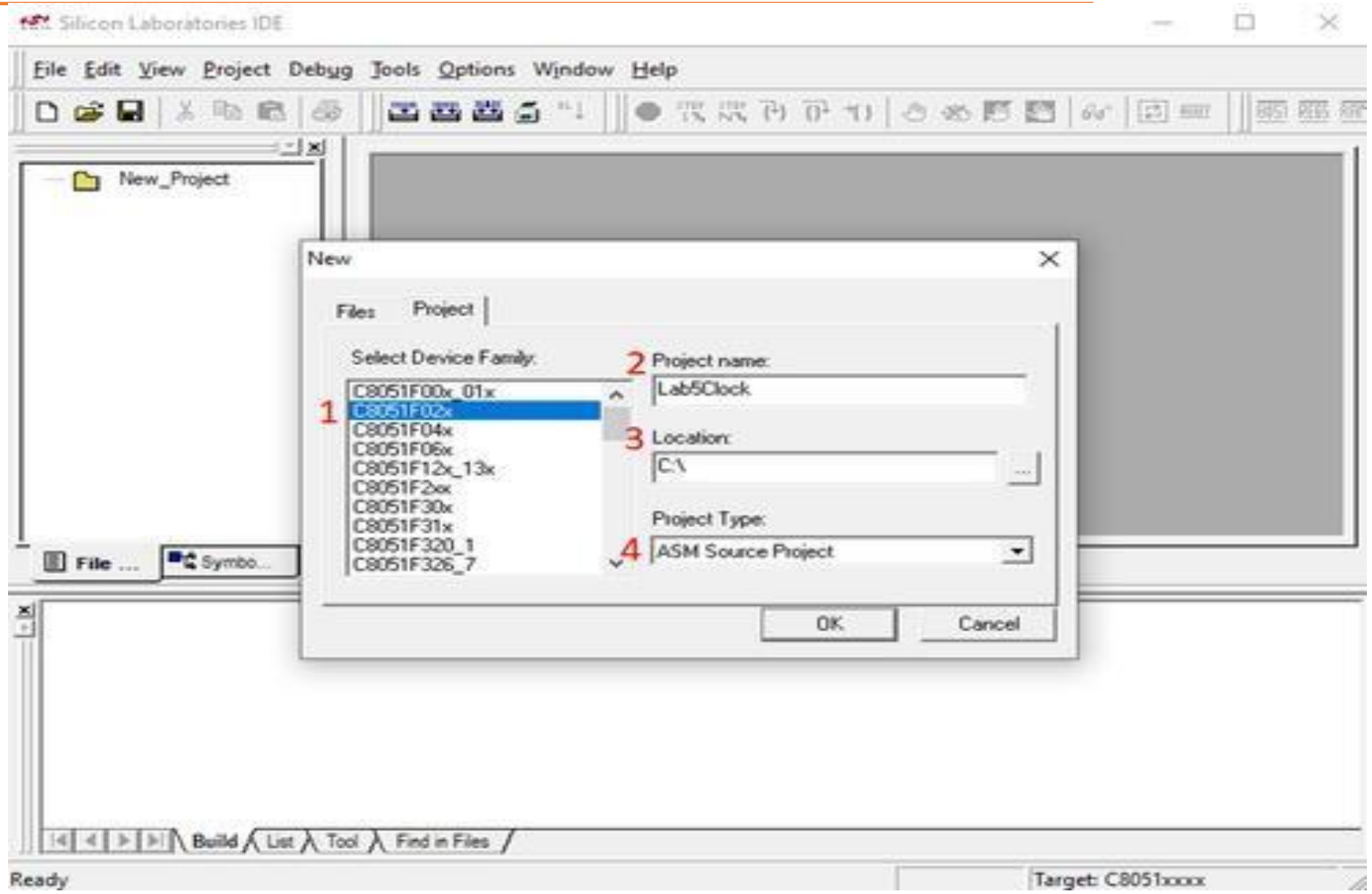




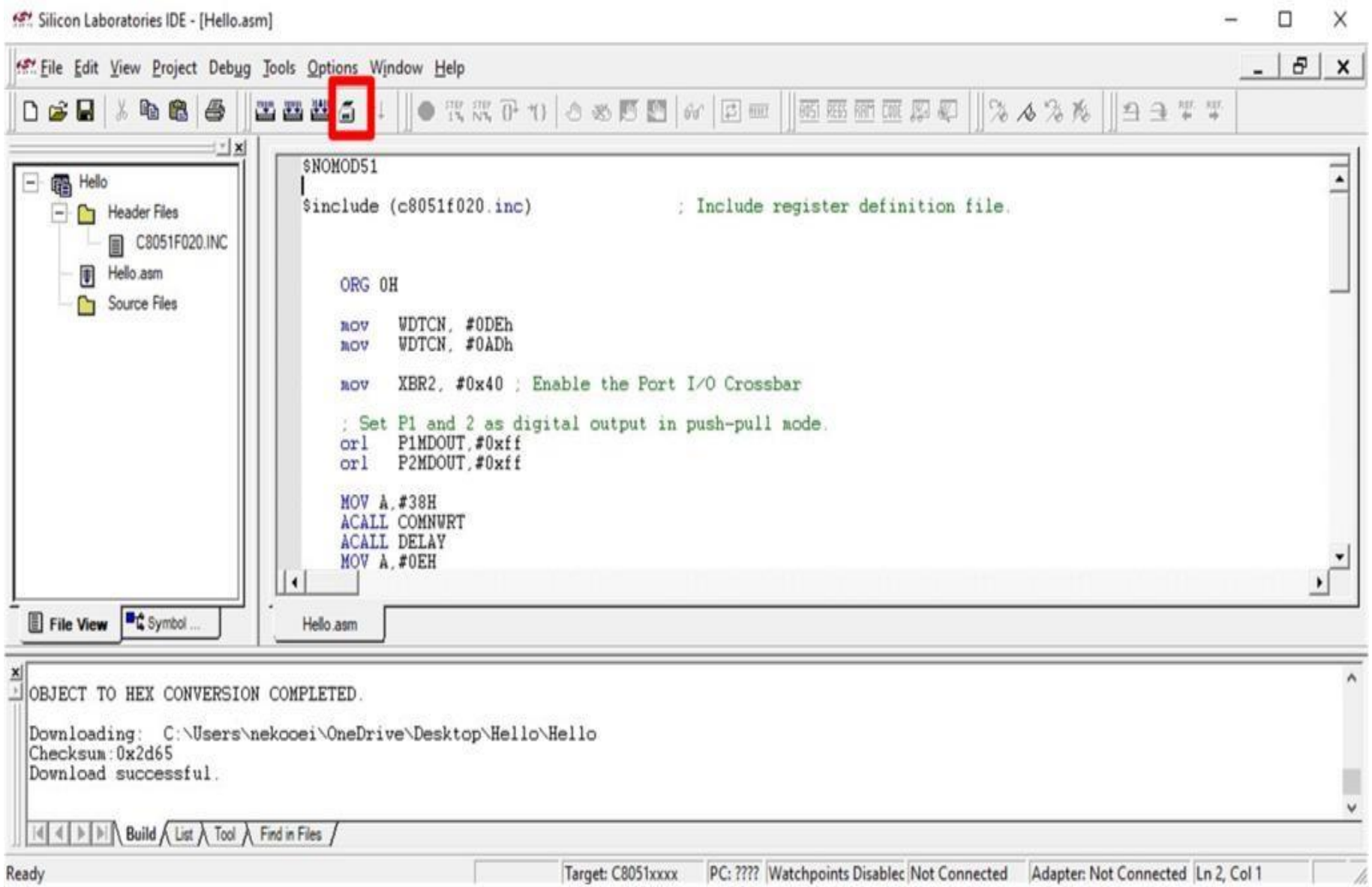
8051 Development kits



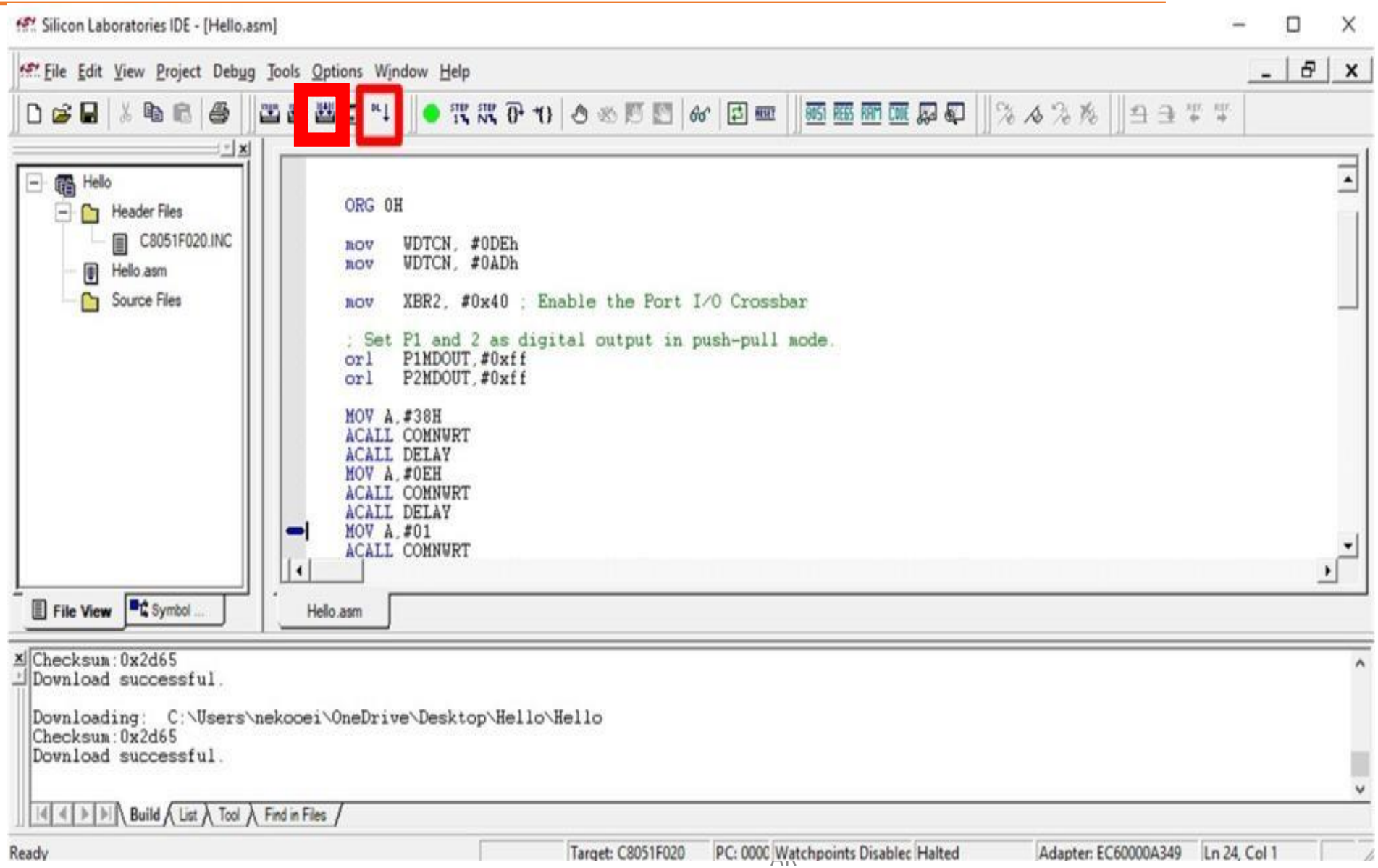
8051 Development kits



8051 Development kits



8051 Development kits



8051 Development kits