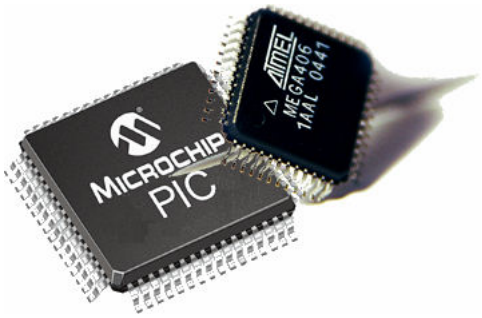
8051 Microcomputer Overview



8051 Microcomputer Overview

1.1 INTRODUCTION

Figure 1.1 shows a functional block of the internal operation of an 8051 microcomputer. The internal components of the chip are shown within the broken line box.

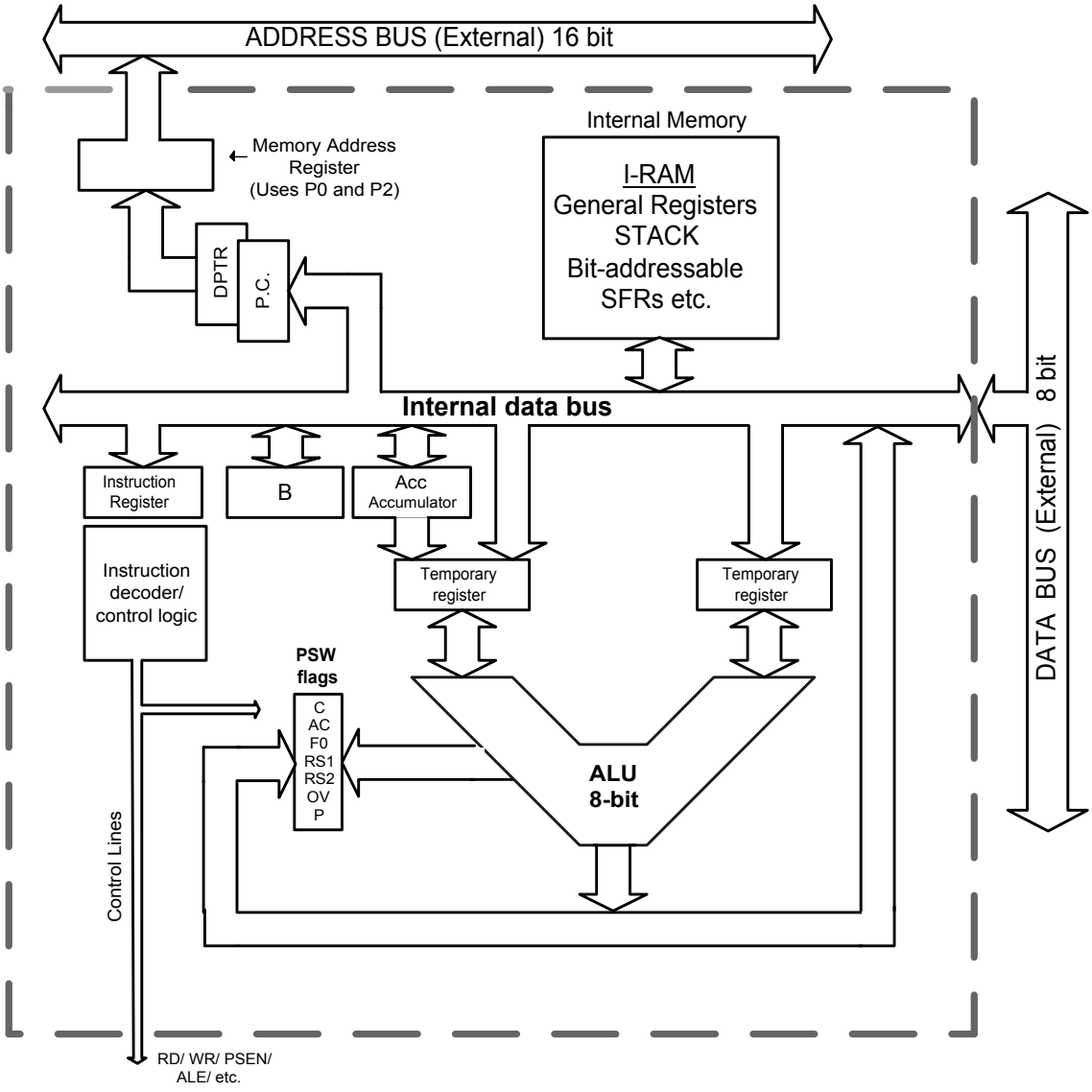


Figure 1.1 8051 functional block diagram.

Figure 1.2 shows the external code memory and data memory connected to the 8051 chip.

Note – part of the external code memory can be located within the chip but we will ignore this feature for now. Also, variants of the chip will allow a lot more memory devices and I/O devices to be accommodate within the chip but such enhanced features will not be considered right now.

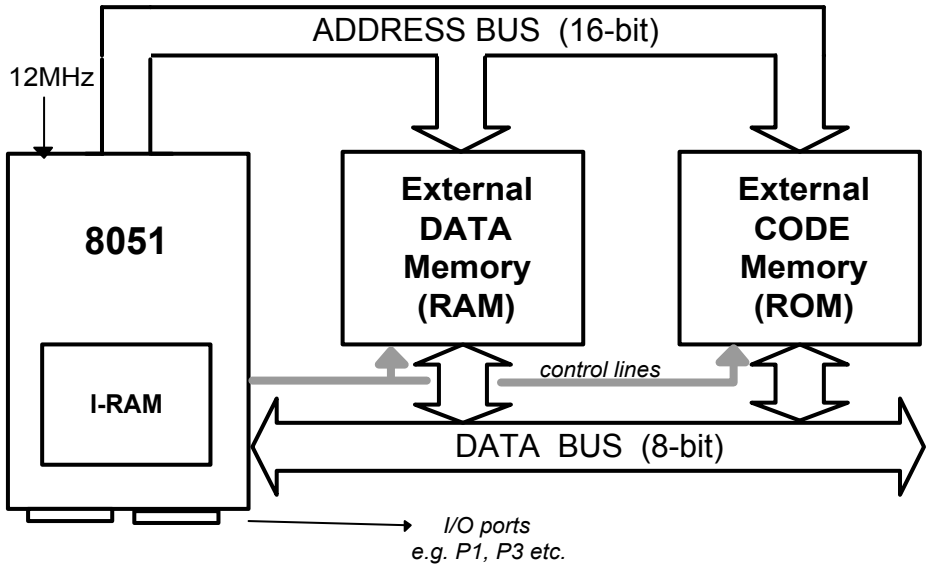


Figure 1.2 8051 chip with external memory

A quick comparison with the well known Pentium processor

A modern PC is powered by a Pentium processor (or equivalent), which is really a very powerful microprocessor. Where the 8051 microcontroller represents the low end of the market in terms of processing power, the Pentium processor is one of the most complex processors in the world. Figure 1.3 shows a simplified block diagram of the Pentium processor and a simple comparison between the 8051 and the Pentium is given in the table below.

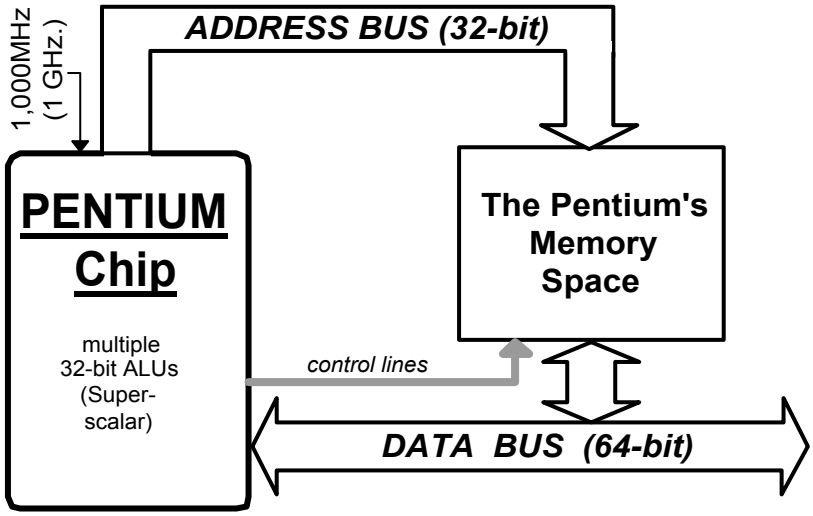


Figure 1.3 Simplified diagram of a Pentium processor

Simple comparison: Pentium vs. 8051

FEATURE	8051	PENTIUM	COMMENT
Clock Speed	12Mhz. typical but 60MHz. ICs available	1,000 MHz. (1GHz.)	8051 internally divides clock by 12 so for 12MHz. clock effective clock rate is just 1MHz.
Address bus	16 bits	32 bits	8051 can address 2^{16}, or 64Kbytes of memory. Pentium can address 2^{32}, or 4 GigaBytes of memory.
Data bus	8 bits	64 bits	Pentium's wide bus allows very fast data transfers.
ALU width	8 bits	32 bits	But - Pentium has multiple 32 bit ALUs – along with floating-point units.
Applications	Domestic appliances,	Personal Computers	

	Peripherals, automotive etc.	And other high performance areas.	
Power consumption	Small fraction of a watt	Tens of watts	Pentium runs hot as power consumption increases with frequency.
Cost of chip	About 2 Euros. In volume	About 200 Euros – Depending on spec.	

The basic 8051 chip includes a number of peripheral I/O devices including two t Timer/Counters, 8-bit I/O ports, and a UART. The inclusion of such devices on the 8051 chip is shown in figure 1.4. These I/O devices will be described later.

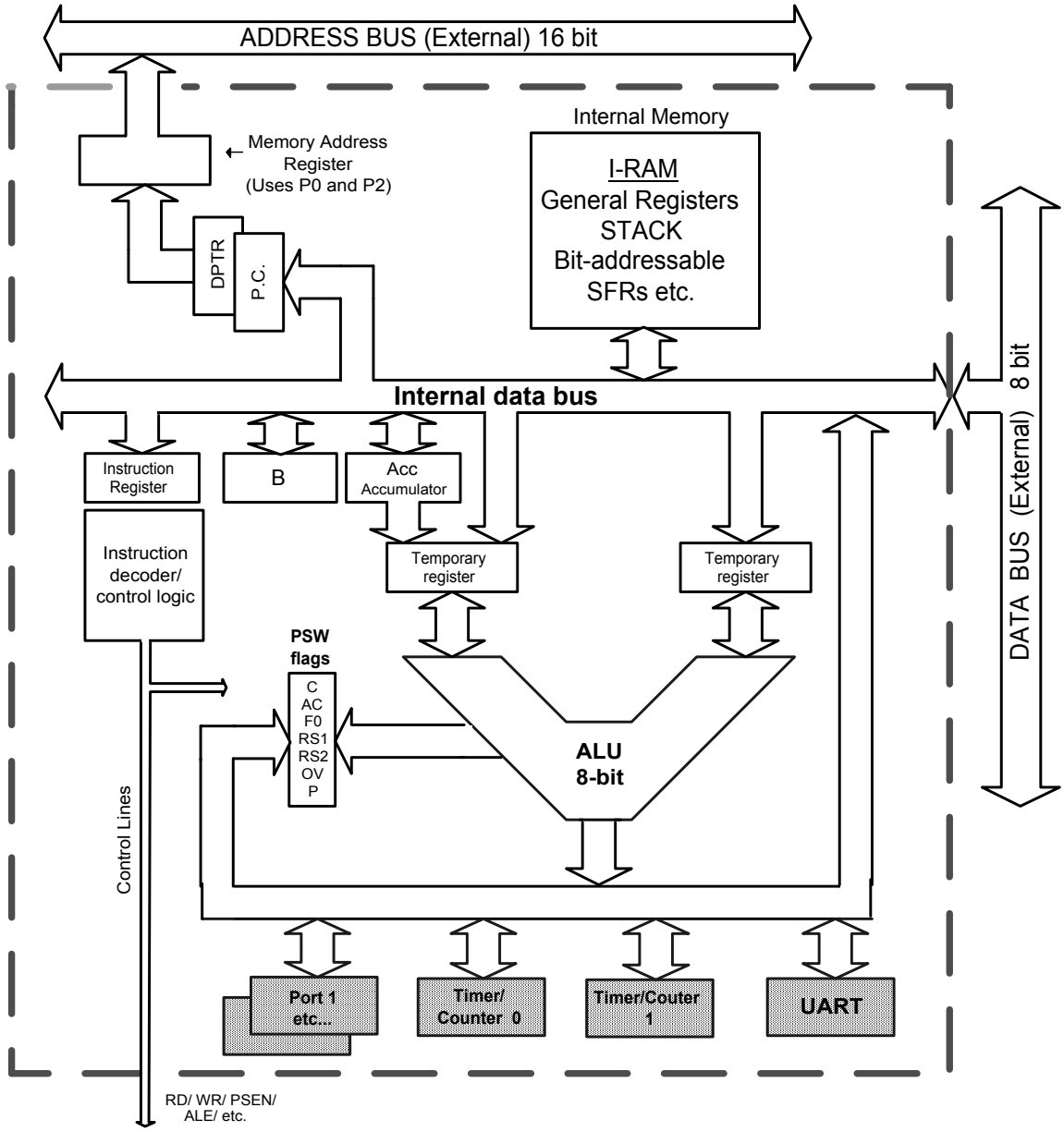


Figure 1.4 8051 showing the on-chip I/O devices

1.2 MEMORY AND REGISTER ORGANISATION

The 8051 has a separate memory space for code (programs) and data. We will refer here to on-chip memory and external memory as shown in figure 1.5. In an actual implementation the external memory may, in fact, be contained within the microcomputer chip. However, we will use the definitions of internal and external memory to be consistent with 8051 instructions which operate on memory. Note, the separation of the code and data memory in the 8051 architecture is a little unusual. The separated memory architecture is referred to as *Harvard* architecture whereas *Von Neumann* architecture defines a system where code and data can share common memory.

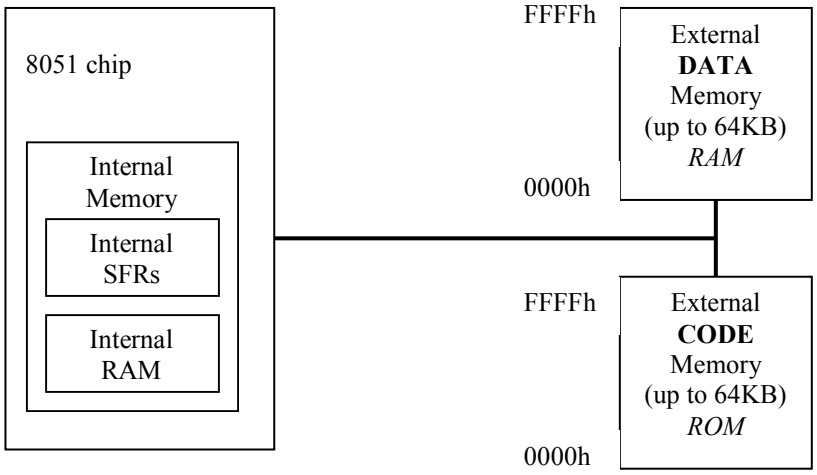


Figure 1.5 8051 Memory representation

External Code Memory
The executable program code is stored in this code memory. The code memory size is limited to 64KBytes (in a standard 8051). The code memory is *read-only* in normal operation and is programmed under special conditions e.g. it is a PROM or a Flash RAM type of memory.

External RAM Data Memory
This *is read-write* memory and is available for storage of data. Up to 64KBytes of external RAM data memory is supported (in a standard 8051).

Internal Memory
The 8051's on-chip memory consists of 256 memory bytes organised as follows:

First 128 bytes:	00h to 1Fh	Register Banks
	20h to 2Fh	Bit Addressable RAM
	30 to 7Fh	General Purpose RAM
Next 128 bytes:	80h to FFh	Special Function Registers

The first 128 bytes of internal memory is organised as shown in figure 1.6, and is referred to as Internal RAM, or IRAM.

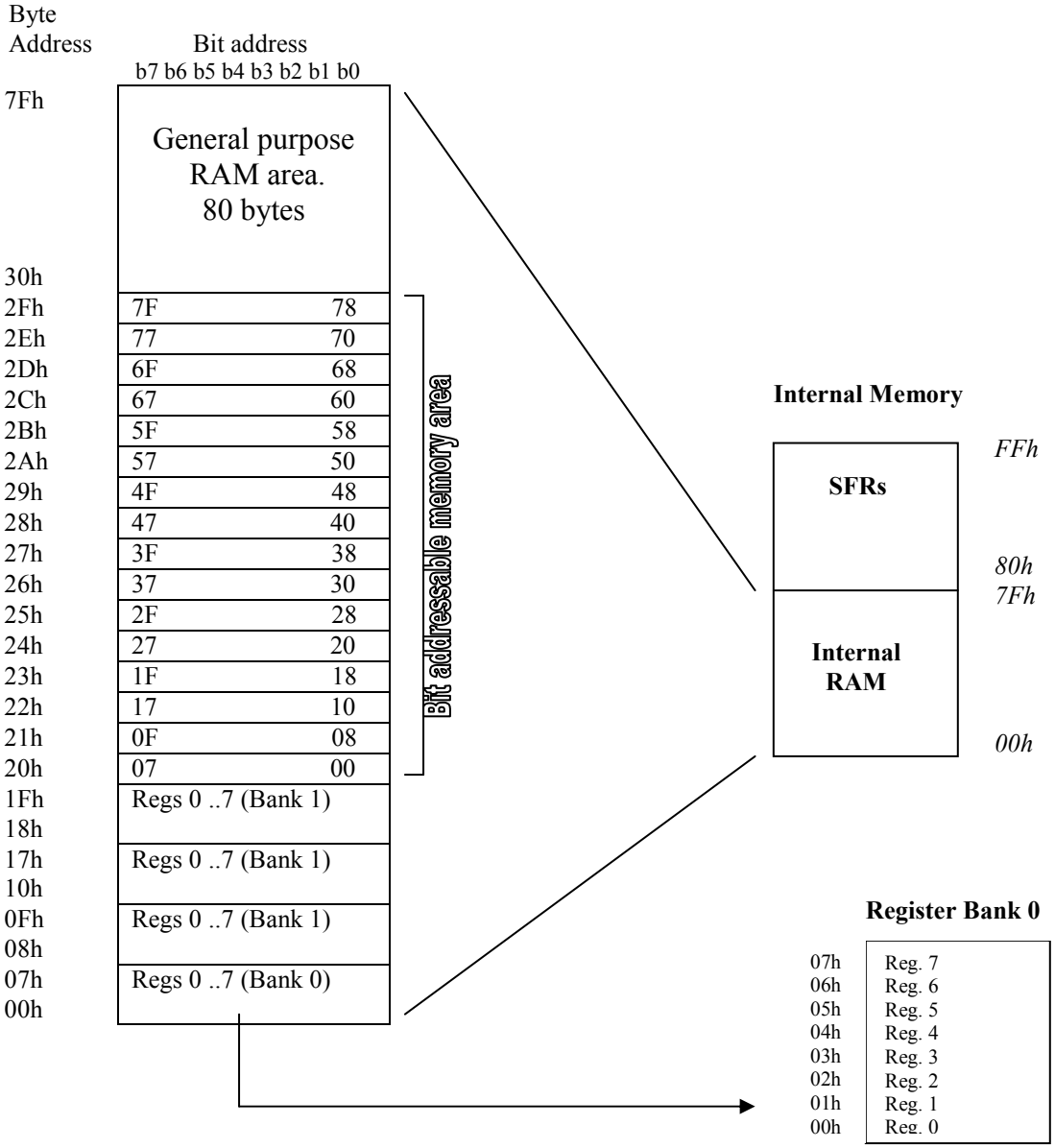


Figure 1.6 Organisation of Internal RAM (IRAM) memory

Register Banks: 00h to 1Fh

The 8051 uses 8 general-purpose registers R0 through R7 (R0, R1, R2, R3, R4, R5, R6, and R7). These registers are used in instructions such as:

ADD A, R2 ; adds the value contained in R2 to the accumulator

Note since R2 happens to be memory location 02h in the Internal RAM the following instruction has the same effect as the above instruction.

ADD A, 02h

Now, things get more complicated when we see that there are four banks of these general-purpose registers defined within the Internal RAM. For the moment we will consider register bank 0 only. Register banks 1 to 3 can be ignored when writing introductory level assembly language programs.

Bit Addressable RAM: 20h to 2Fh

The 8051 supports a special feature which allows access to *bit variables*. This is where individual memory bits in Internal RAM can be set or cleared. In all there are 128 bits numbered 00h to 7Fh. Being bit variables any one variable can have a value 0 or 1. A bit variable can be set with a command such as SETB and cleared with a command such as CLR. Example instructions are:

SETB 25h ; sets the bit 25h (becomes 1)

CLR 25h ; clears bit 25h (becomes 0)

Note, bit 25h is actually bit b5 of Internal RAM location 24h.

The Bit Addressable area of the RAM is just 16 bytes of Internal RAM located between 20h and 2Fh. So if a program writes a byte to location 20h, for example, it writes 8 *bit variables*, bits 00h to 07h at once.

Note bit addressing can also be performed on some of the SFR registers, which will be discussed later on.

General Purpose RAM: 30h to 7Fh

These 80 bytes of Internal RAM memory are available for general-purpose data storage. Access to this area of memory is fast compared to access to the main memory and special instructions with single byte operands are used. However, these 80 bytes are used by the system stack and in practice little space is left for general storage. The general purpose RAM can be accessed using direct or indirect addressing modes. Examples of direct addressing:

MOV A, 6Ah ; reads contents of address 6Ah to accumulator

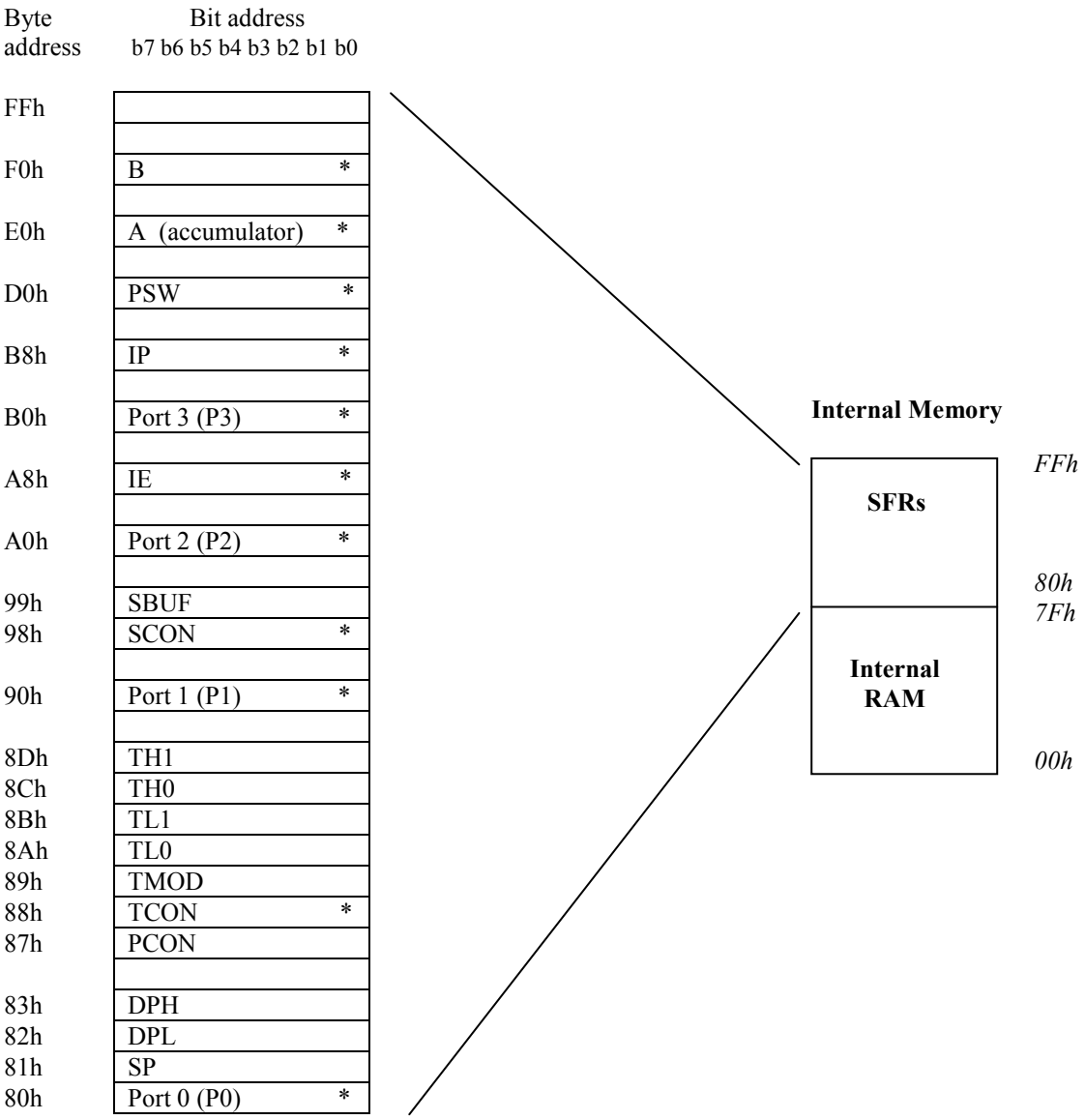
Examples for indirect addressing (**use registers R0 or R1**):

```
MOV  R1, #6Ah      ; move immediate 6Ah to R1
MOV  A, @R1        ; move indirect: R1 contains address of Internal RAM which
                     contains data that is moved to A.
```

These two instructions have the same effect as the direct instruction above.

SFR Registers

The SFR registers are located within the Internal Memory in the address range 80h to FFh, as shown in figure 1.7. Not all locations within this range are defined. Each SFR has a very specific function. Each SFR has an address (within the range 80h to FFh) and a name which reflects the purpose of the SFR. Although 128 byes of the SFR address space is defined only 21 SFR registers are defined in the standard 8051. Undefined SFR addresses should not be accessed as this might lead to some unpredictable results. Note some of the SFR registers are *bit addressable*. SFRs are accessed just like normal Internal RAM locations.



* indicates the SFR registers which are bit addressable

Figure 1.7 SFR register layout

We will discuss a few specific SFR registers here to help explain the SFR concept. Other specific SFR will be explained later.

Port Registers SFR
The standard 8051 has four 8 bit I/O ports: P0, P1, P2 and P3.

For example Port 0 is a physical 8 bit I/O port on the 8051. Read (input) and write (output) access to this port is done in software by accessing the SFR P0 register which is located at address 80h. SFR P0 is also bit addressable. Each bit corresponds to a physical I/O pin on the 8051. Example access to port 0:

```
SETB  P0.7    ; sets the MSB bit of Port 0
CLR   P0.7    ; clears the MSB bit of Port 0
```

The operand P0.7 uses the dot operator and refers to bit 7 of SFR P0. The same bit could be addressed by accessing bit location 87h. Thus the following two instructions have the same meaning:

```
CLR   P0.7
CLR   87h
```

PSW Program Status Word
PSW, the Program Status Word is at address D0h and is a bit-addressable register. The status bits are listed in table 1.1.

Table 1.1. Program status word (PSW) flags

Symbol Bit	Address	Description
C (or CY)	PSW.7 D7h	Carry flag
AC	PSW.6 D6h	Auxiliary carry flag
F0	PSW.5 D5h	Flag 0
RS1	PSW.4 D4h	Register bank select 1
RS0	PSW.3 D3h	Register bank select 0
0V	PSW.2 D2h	Overflow flag
	PSW.1 D1h	Reserved
P	PSW.0 D0h	Even Parity flag

Carry flag. C
This is a conventional carry, or borrow, flag used in arithmetic operations. The carry flag is also used as the 'Boolean accumulator' for Boolean instruction operating at the bit level. This flag is sometimes referenced as the CY flag.

Auxiliary carry flag. AC
This is a conventional auxiliary carry (half carry) for use in BCD arithmetic.

Flag 0. F0
This is a general-purpose flag for user programming.

Register bank select 0 and register bank select 1. RS0 and RS1
These bits define the active register bank (bank 0 is the default register bank).

Overflow flag. OV
This is a conventional overflow bit for signed arithmetic to determine if the result of a signed arithmetic operation is out of range.

Even Parity flag. P
The parity flag is the accumulator parity flag, set to a value, 1 or 0, such that the number of '1' bits in the accumulator plus the parity bit add up to an even number.

Stack Pointer
The Stack Pointer, SP, is an 8-bit SFR register at address 81h. The small address field (8 bits) and the limited space available in the Internal RAM confines the stack size and this is sometimes a limitation for 8051 programmes. The SP contains the address of the data byte currently on the top of the stack. The SP pointer in initialised to a defined address. A new data item is 'pushed' on to the stack using a PUSH instruction which will cause the data item to be written to address SP + 1. Typical instructions, which cause modification to the stack are: PUSH, POP, LCALL, RET, RETI etc.. The SP SFR, on start-up, is initialised to 07h so this means the stack will start at 08h and expand upwards in Internal RAM. If register banks 1 to 3 are to be used the SP SFR should be initialised to start higher up in Internal RAM. The following instruction is often used to initialise the stack:

```
MOV SP, #2Fh
```

Data Pointer
The Data Pointer, DPTR, is a special 16-bit register used to address the external code or external data memory. Since the SFR registers are just 8-bits wide the DPTR is stored in two SFR registers, where DPL (82h) holds the low byte of the DPTR and DPH (83h) holds the high byte of the DPTR. For example, if you wanted to write the value 46h to external data memory location 2500h, you might use the following instructions:

```
MOV     A,  #46h            ; Move immediate 8 bit data 46h to A (accumulator)

MOV     DPTR, #2504h        ; Move immediate 16 bit address value 2504h to A.
```

```
                        ; Now DPL holds 04h and DPH holds25h.

MOVX  @DPTR, A          ; Move the value in A  to external RAM location 2500h.
                          Uses indirect addressing.
```

Note the **MOVX** (Move X) instruction is used to access external memory.

Accumulator

This is the conventional accumulator that one expects to find in any computer, which is used to the hold result of various arithmetic and logic operations. Since the 8051 microcontroller is just an 8-bit device, the accumulator is, as expected, an 8 bit register.

The accumulator, referred to as ACC or A, is usually accessed explicitly using instructions such as:

```
INC A ; Increment the accumulator
```

However, the accumulator is defined as an SFR register at address E0h. So the following two instructions have the same effect:

```
MOV A, #52h         ; Move immediate the value 52h to the accumulator

MOV E0h, #52h       ; Move immediate the value 52h to Internal RAM location E0h,
                      which is, in fact, the accumulator SFR register.
```

Usually the first method, MOV A, #52h, is used as this is the most conventional (and happens to use less space, 2 bytes as oppose to 3 bytes!)

B Register

The B register is an SFR register at addresses F0h which is bit-addressable. The B register is used in two instructions only: i.e. MUL (multiply) and DIV (divide). The B register can also be used as a general-purpose register.

Program Counter

The PC (Program Counter) is a 2 byte (16 bit) register which always contains the memory address of the next instruction to be executed. When the 8051 is reset the PC is always initialised to 0000h. If a 2 byte instruction is executed the PC is incremented by 2 and if a 3 byte instruction is executed the PC is incremented by three so as to correctly point to the next instruction to be executed. A jump instruction (e.g. LJMP) has the effect of causing the program to branch to a newly specified location, so the jump instruction causes the PC contents to change to the new address value. Jump instructions cause the program to flow in a non-sequential fashion, as will be described later.

SFR Registers for the Internal Timer

The set up and operation of the on-chip hardware timers will be described later, but the associated registers are briefly described here:

TCON, the Timer Control register is an SFR at address 88h, which is bit-addressable. TCON is used to configure and monitor the 8051 timers. The TCON SFR also contains some interrupt control bits, described later.

TMOD, the Timer Mode register is an SFR at address 89h and is used to define the operational modes for the timers, as will be described later.

TL0 (Timer 0 Low) and TH0 (Timer 0 High) are two SFR registers addressed at 8Ah and 8Bh respectively. The two registers are associated with Timer 0.

TL1 (Timer 1 Low) and TH1 (Timer 1 High) are two SFR registers addressed at 8Ch and 8Dh respectively. These two registers are associated with Timer 1.

Power Control Register

PCON (Power Control) register is an SFR at address 87h. It contains various control bits including a control bit, which allows the 8051 to go to 'sleep' so as to save power when not in immediate use.

Serial Port Registers

Programming of the on-chip serial communications port will be described later in the text. The associated SFR registers, SBUF and SCON, are briefly introduced here, as follows:

The SCON (Serial Control) is an SFR register located at addresses 98h, and it is bit-addressable. SCON configures the behaviour of the on-chip serial port, setting up parameters such as the baud rate of the serial port, activating send and/or receive data, and setting up some specific control flags.

The SBUF (Serial Buffer) is an SFR register located at address 99h. SBUF is just a single byte deep buffer used for sending and receiving data via the on-chip serial port

Interrupt Registers

Interrupts will be discussed in more detail later. The associated SFR registers are:

IE (Interrupt Enable) is an SFR register at addresses A8h and is used to enable and disable specific interrupts. The MSB bit (bit 7) is used to disable all interrupts.

IP (Interrupt Priority) is an SFR register at addresses B8h and it is bit addressable. The IP register specifies the relative priority (high or low priority) of each interrupt. On the 8051, an interrupt may either be of low (0) priority or high (1) priority. .