

the CPU, of the system timers and the CPU machine cycles. The machine cycles are for fetching codes and data from memory and then decoding and executing them at the processor and for transferring the results to memory.

For processing units, a highly stable oscillator is required and the processor clock-out signal provides the clock for synchronizing all system units with the processor.

1.3.3 System Timers and Real-time Clocks

A timer circuit is suitably configured as the system-clock, which ticks and generates system interrupts periodically; for example, 60 times in 1s. The interrupt service routines then perform the required operation.

A timer circuit is suitably configured as the real-time clock (RTC) that generates system interrupts periodically for the schedulers, real-time programs and for periodic saving of time and date in the system.

The RTC or system timer is also used to obtain software-controlled delays and time-outs. An RTC functions as driver for software timers (SWTs). [Sections 3.6 and 3.8]

Microcontrollers also provide internal timer circuits for counting and timing devices.

To schedule the various tasks and for real-time programming, an RTC or system clock is needed. The clock also drives the timers for various timing and counting needs in a system.

1.3.4 Reset Circuit, Power-up Reset and Watchdog-Timer Reset

The program counter (PC) holds the address from where the instruction is to be fetched for execution. In 80x86 processors, the instruction pointer (IP) holds that address. A code segment register (CS) holds the base address of the code memory segment. The CS address equals the code starting address when the IP = 0 at the start of a code segment. The IP increments when the program executes the codes.

Reset means that the processor begins the processing of instructions from a starting address. That address is one that is set by default in the processor PC (or IP and CS in x86 processors) on a power-up. From that address in memory, program-instructions are fetched following the reset of the processor. A program that is reset and runs on a power-up can be one of the following: (i) A system program that executes from the beginning. (ii) A system boot-up program. (iii) A system initialization program.

In certain processors, for example, 68HC11 and HC12, there are two start-up addresses. One is based on the power-up reset vector and the other on the reset vector after the reset instruction or after a time-out (for example, from a watchdog timer). The processor fetches the bytes for the PC from the first power-up reset vector on power-up. The processor fetches the bytes for the PC from the second reset vector on the watchdog timer timing out or on executing the reset instruction.

The reset circuit activates for a fixed period (a few clock cycles) and then deactivates. The processor circuit keeps the reset pin active and then deactivates to let the program proceed from a default beginning address. The reset pin or the internal reset signal, if connected to the other units (for example, the IO interface or the serial interface) in the system, is activated again by the processor; it becomes an outgoing pin to enforce a reset state in other sister units of the system. On deactivation of the reset that succeeds the processor activation, a program executes from a start-up address.

Reset can be activated by an external reset circuit that activates on power-up, on switching-on reset of the system or on detection of a low voltage (e.g. <4.5 V when what is required is 5 V on the system

supply rails). This circuit output connects to a pin called the reset pin of the processor. This circuit may be a simple RC circuit, an external IC circuit or a custom-built IC. Examples of ICs are MAX 6314 and Motorola MC 34064.

Alternatively, it can also be activated by any one of the following: (i) software instruction; (ii) time-out by a programmed timer known as a watchdog timer (or on an internal signal called COP in 68HC11 and 68HC12 families); (iii) a clock monitor detecting a slowdown below certain frequencies.

The watchdog timer is a timing device that resets the system after a predefined timeout. It is activated within the first few clock cycles after power-up. It has a number of applications. In many embedded systems reset by a watchdog timer is very essential because it helps in rescuing the system if a fault develops and the program gets stuck. On restart, the system can function normally. Most microcontrollers have on-chip watchdog timers. The watchdog timer device is described in detail in Section 3.7.

Consider a system controlling temperature. Assume that when the program starts executing, the sensor inputs work all right. However, before the desired temperature is achieved, the sensor circuit develops some fault. The controller will continue delivering the current nonstop if the system is not reset. Consider another example of a system for controlling a robot. Assume that the interfacing motor control circuit in the robot arm develops a fault during the run. In such cases, the robot arm may continue to move unless there is a watchdog timer control. Otherwise, the robot will break its own arm!

When a program executes the program counter increments or changes. An important circuit that associates a system is its reset circuit that can change the program counter to a power-up default value. A program that is reset and runs on a power-up can be one of the following: (i) A system program that executes from the beginning. (ii) A system boot-up program. (iii) A system initialization program.

The watchdog timer reset is a required feature in control applications.

1.3.5 Memory

In a system, there are various types of memory. Figure 1.4 shows a chart for various forms of memory that are present in systems. These are as follows:

1. Internal RAM of 256 or 512 bytes in a microcontroller for registers, temporary data and stack.
2. Internal ROM/PROM/E²PROM for about 4 kB to 64 kB of program (in the case of microcontrollers).
3. External RAM for the temporary data and stack (in most systems) or internal caches (in the case of certain microprocessors).

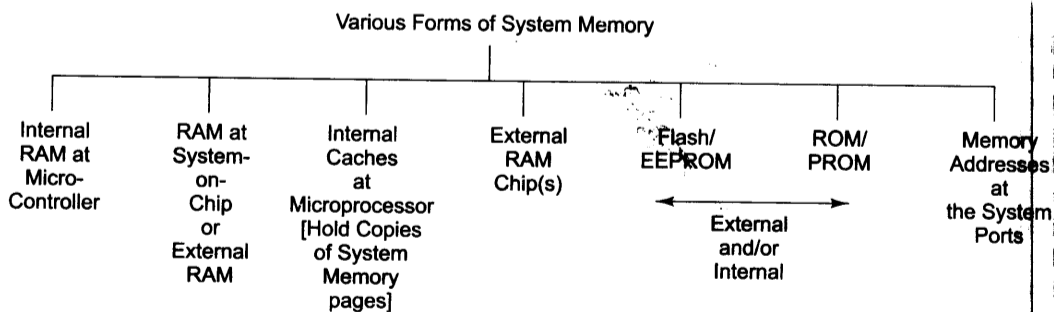


Fig. 1.4 The various forms of memories in the system