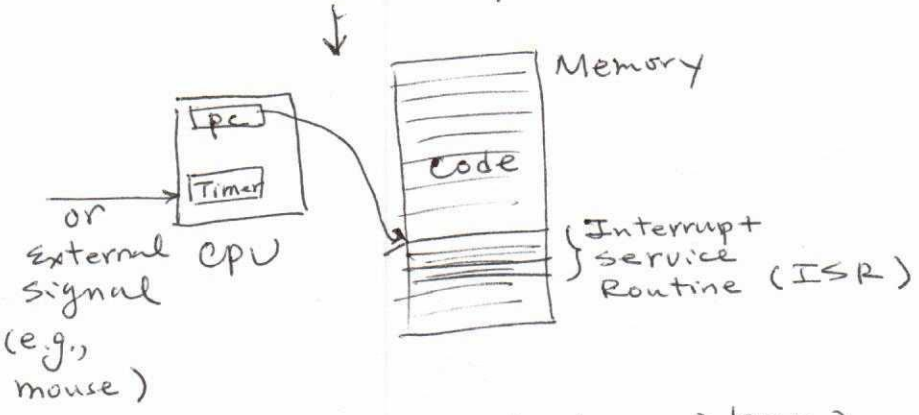
Hardware Interrupt

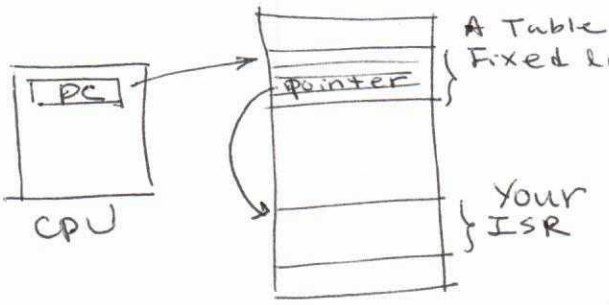


Q. How does the CPU (hardware) know the address of the Interrupt service Routine (ISR)?

Ans. The logical solution (answer) is that the manufacturer of the CPU assigns a fixed address for the ISR.

Q. This solution works. However, it significantly restricts the flexibility of the developer. Any better modification?

A.



Q. The pointer ~~that is at a~~ in the table needs to point to the ISR location. Who places that pointer to the table?

A. You, the developer, place that pointer to the table, since it is your ISR and you know the ISR's location.

Q. I am not sure I know the ISR location. Am I wrong?

A. Although you don't know the exact address (location) of your ISR. YOUR compiler/loader knows. You can tell your compiler/loader

Q. to find out that address for you. How?

A. It is called assembly language directive. (Search the Key word ~~Assembly~~ Directive)

Q. Do I, as a developer, know the location of the table?

A. Yes. The manufacturer of the CPU needs to tell you. (See MC13xxx RM)

Q. Do we have a name for that table?

A. Vector Table or Exception Table or Jump Table

Q. How come we need to have a table (not just a row/single entry)?

A. There are other types of interrupt. (For example Reset, Undefined Instruction)

Q. How do I place a pointer to one entry of the Vector/Jump/Exception Table?

A. As a developer, you use assembly directives to

1. find the address of your ISR
2. locate the table address
3. Place (1) in the table

Q. I got the concept. May I see a real world example?

A. See our group project step 3 - A real world timer interrupt example.