

Do we need local CPU in multi-serial card ?

1. Confused:

We can have faster and faster CPU in PC. Right now we always have GHz grade CPU in PC and we just have MHz grade CPU in multi-serial card. Do we still need to have local CPU in multi-serial card?

2. How to work in one PC:

In standard serial port we need to transmit data in one serial port. The motherboard's main CPU need to check UART controller and send data to such UART's register. All these actions are I/O access. We can not loss one I/O command or let such I/O command sequence in wrong order. So we must wait every I/O command one by one finished. Even though we have GHz grade CPU, we still need to wait such MHz grade I/O controller to finish their work. So whole PC system's performance will be degraded by such condition. In our experience the system performance for one PC to handle serial port is about 200K --300KByte/sec, even though we have very fast CPU. For one 9600bps serial port we need to transmit one byte of data. Generally we need one start bit, eight data bit, no parity bit and one stop bit. It will be 10bit transmission for one byte data. So we need about $9600/10=960$ byte/sec or about 1Kbyte/sec system process power. For one 115200bps serial port we may need about 12KByte/sec system process power. So you can imagine one condition for PC to handle 20 serial ports to transmit in 115200bps speed. In this condition whole CPU process power may be consumed in serial port tansmission. So there are no other CPU power left for other task. This is the first problem for one PC to hanle multi-serial card without local CPU.

The other problem for PC is IRQ confliction problem. From PCAT structure we do not have many IRQ available for add-on card. Some card's IRQ can be shared and other card's IRQ may not be shared. When one UART generate IRQ to main CPU, you must confirm this IRQ do not conflict with other card. Or you must confirm that this IRQ is shareable in other card. So you may have limitation in IRQ assigned. Different PC and different add-on card may have different condition to solve. Can we run without IRQ in multi-serial card?

Do we need local CPU in multi-serial card ?

3. How to work in multi-serial card with local CPU:

When multi-serial card with local CPU we can run without IRQ. So we can solve the problem in IRQ confliction. Generally we use dual port RAM structure for main CPU and local CPU to exchange data. Local CPU will handle all the task for data transmission in serial port. So the process power in PC will depend one multi-serial card's local CPU. Our product line can support 16KByte/sec upto 360KByte/sec in local CPU. You may have one question in this number. Yes, we say the PC process power is about 200--300KByte/sec. How can one GHz grade main CPU have lower process power than MHz grade CPU? This is due to the IRQ service routine structure. Our PC's main CPU do not have special structure in IRQ handling. Because they do not target in I/O handling. Some CPU may be major in I/O handling. So they have special structure in IRQ service. So that is why we can have local CPU with higher process power than main CPU. Due to all the transmission tasks are handled by local CPU, we just need to prepare the data in dual port RAM. So the access for main CPU in dual port RAM just like the local memory in motherboard. Main CPU just use a little time in data exchange task in dual port RAM. Because every multi-serial cards have local CPU, it means that more card with more process power in main CPU. This structure is similar as asymmetrical multi-processor system. We can have one main CPU to assign task for each dedicated CPU (the local CPU in multi-serial card). Multi-processor work simultaneously.

4. the limitation for multi-serial card with local CPU:

Even though the local CPU in multi-serial card can share the task in main CPU. But it is problem to synchronize both CPU. For example, the signal state change in serial port will be detected by local CPU and tell main CPU in dual port RAM. When main CPU get this condition and send command in dual port RAM. Local CPU can get this command in dual port RAM and do action. In this process period the signal state in serial port may be changed again. So the action in last command may be not suitable for this new condition.

So we suggest that the card with local CPU is suitable for data transmission only application. If user need to detect the real-time status changed condition, it is not suitable for card with local CPU. It is good for user to use card without local CPU. Every task are handled by main CPU. So there are no synchronized problem.