HW7

9.2 9.6 9.10 9.14 9.26

9.2

> Why is a ready bit not needed if synchronous I/O is used?

If synchronous I/O is used, the CPU can predict after how many cycles the input will appear and when it'll be output, and he only needs to read/write it at regular intervals.
Though the ready bit is the simplest form of *synchronization*, if the CPU can know the time to read, it's already synchronic, then it doesn't need to use the more time-consuming *polling* to check.

9.6

> What problem could occur if a program does not check the ready bit of the KBSR before reading the KBDR?

If KBSR[15] is 0, the data contained in the KBDR has been read by the processor. Thus, if the keyboard hardware does not check the KBSR before reading the KBDR, the same character will be read again and again.

9.10

> What problem could occur if the **CPU** does not check the DSR before writing to the DDR?

initial version: DDR can't be written by the display hardware.

change to *CPU*:

If DSR[15] is 0, it means the contents of DDR[7:0] are being processed and hasn't finished yet. So if write new data into DDR at this time, the previous data (which is being processed) may lost.

9.14

> An LC-3 Load instruction specifies the address xFE02. How do we know whether to load from the KBDR or from memory location xFE02?

It's decided by the organization of memory. Addresses xFE00 to xFFFF do not correspond to memory locations at all. That is, the last address of a memory location is xFDFF.
Meanwhile, when getting the address xFE02 from MAR, **the Address Control LOGIC** will give out control signal to enable KBDR work and let the INMUX choose KBDR.

9.26

> The following program is supposed to print the number 5 on the screen. It does not work. Why? Answer in no more than ten words, please.

`JSR B` reset R7, can't jump to `OUT`, go loop.