A device driver process carries out character I/O via a Universal Asynchronous Receiver/Transmitter (UART).

(i) Why is hardware–software synchronisation needed? \( \) [1 mark]

(ii) Describe polled operation. \( \) [2 marks]

(iii) Describe interrupt-driven operation. \( \) [2 marks]

(iv) Draw a state transition diagram for the device-driver process. Indicate the events that cause each transition and in each case explain the effect on the device driver’s process descriptor and the operating system’s scheduling queues. Assume interrupt-driven software. \( \) [7 marks]

The device driver process fills/empties a buffer of fixed size in an I/O buffer area. A process carrying out application requests reads and writes data in variable-sized amounts from the buffer.

(i) Why must mutually exclusive access to the buffer be enforced? \( \) [2 marks]

(ii) Why is condition synchronisation needed? \( \) [2 marks]

(iii) What is wrong with the following pseudocode fragment from the device-driver’s specification, where:

- \texttt{buffer-lock} is a semaphore initialised to 1,
- \texttt{space} is a semaphore initialised to the buffer size in bytes,
- \texttt{data} is a semaphore initialised to 0?

on input: on output:
\begin{align*}
\text{WAIT(buffer-lock);} & \quad \text{WAIT(buffer-lock)}; \\
\text{if buffer is full then WAIT(space);} & \quad \text{if buffer is empty then WAIT(data)}; \\
\text{write a character into the buffer;} & \quad \text{read a character from the buffer;} \\
\text{SIGNAL(buffer-lock);} & \quad \text{SIGNAL(buffer-lock)};
\end{align*} \( \) [4 marks]