● *Data Registers:* These are the registers that contain the data being input or output, and so they may be read-write or separate read and write.
In order to conserve memory space it is common practice for more than one register to occupy the same address. For example, a status register and a control register may be at the same address; the one that appears at that address is determined by whether you are reading or writing to it. Similarly, an input data register and an output data register may share an address.

The 6820 PIA contains six registers and occupies four consecutive bytes of memory space. The chip actually contains two independent ports, each of which uses three registers. The peripheral side of the chip has eight data lines and two control lines for each port. The two control lines would be connected to appropriate control lines on the peripheral so that they can be used to determine status. Control line 1 is for incoming control signals only, but control line 2 can be programmed to receive or send control signals.

The three registers are:

● A data register, which can function for both input and output, since each bit can be independently set.
● A data direction register, each bit of which is used to set the corresponding bit in the data register as input (0) or output (1).
● A combined control/status register.

The data direction register and the data register share the same address. The state of one of the bits in the control register determines which of these appears at this address. The table in the margin gives the offset from the base address of the chip for the addresses of each of the registers.

Register	Offset
Data A	0
Data Direction A	0
Control/Status A	1
Data B	2
Data Direction B	2
Control/Status B	3

The bits in the control/status register are assigned as follows:

Bit	Function
7	Status bit for control line 1; set to one when a control signal is received and automatically cleared to zero when the data register is read
6	Status bit for control line 2; as bit 7
5	Determines whether control line 2 is used for input (zero) or output (one)
4	Determines the nature of the control signal on line 2
3	If control line 2 is set for input then a one here enables interrupts from bit 6; if output then it helps determine the nature of the signal
2	Selects between data (1) and data direction (0) registers
1	Determines the nature of the control signal on line 1
0	A one here enables interrupts from bit 7

For the moment we shall not be considering the use of interrupts, nor shall we be concerned with the detailed effects of bits 1 and 4. Note that when writing to the register to set the control bits it is impossible to affect bits 6 and 7.

The first of our example programs sets up and uses a 6820 chip to control a printer via a standard Centronics interface. The latter specifies a large number of control lines as well as the eight data lines. We are not concerned with the detail of these except to note that one control line (called the *strobe*) is used to signal to the printer that a character is on the way. This should be connected to control line 2, which must be set for output. Another control signal (termed the *acknowledge*) is used by the printer to indicate that it is ready for the next character to be sent. This should be connected to control line 1. The eight data lines should clearly be connected to the eight data outputs from the PIA port.

To set up the port we must select the data direction register and program all eight bits for output, then select the data register, and set control line 2 for output. To use the chip we continually read the control/status register until a one appears in bit 7, indicating that the printer is ready for a character. We can then write a character to the data register, which automatically sends a control signal out on control line 2. Bit 6 will be set to one when the character has been transmitted. We must then read the data register to clear bits 6 and 7 and repeat the process until the last character has been transmitted. The process of sending and receiving control signals between the processor and the peripheral is known as *handshaking*.

We shall assume that the base address of the PIA is given in a table of addresses located at $3000. On entry to the printing subroutine, processor register A contains the index into this table, and Y contains the address of the string to be printed. The string is stored in the normal format; that is, length byte first. There are two subroutines, one to set up the port and one to print the string.

6850 ACIA

The 6850 ACIA is a UART (universal asynchronous receiver/transmitter) that is used for serial communication, normally using the RS232 protocol and possibly a modem. It has four registers and occupies two addresses. There are five connections to the chip on the peripheral side: one line is for transmitted data, one is for received data, and three control lines are for handshaking, if this is required. Two of these are for incoming control signals — DCD (Data Carrier Detect) and CTS (Clear To Send) — and one is for outgoing signals — RTS (Request To Send). The uses of these lines should be fairly obvious from their names, and they may be connected to the similarly named lines on a standard RS232.

The four ACIA registers are given in the margin. In the control register, the most significant bit (bit 7) is used to enable interrupts for receiving data. Bits 5 and 6 are used to enable or disable transmission interrupts and to determine the nature of the control signal sent out on the RTS line. Bits 2, 3 and 4 are used to determine the size of the 'package' that is actually transmitted. When a byte is transmitted over a serial link there are usually at least 10 bits sent, beginning with a start bit, which is detected by the receiver so that it knows that data is following. The actual data itself

Register	Offset
Control Register	0
Status Register	0
Transmit Data	1
Receive Data	2