Overview

In this unit your students should:

- learn how to make a PIC change the state of its output port at fixed frequency
- learn how to implement test-and-skip loops for the input port
- understand how to use registers as loop counters in programs
- be able to program a PIC to act as a logic system
- analyse flowcharts and describe the behaviour that they impose on a PIC

This should not require more than 5 hours of class time.

Hour	Suggested Activity
1	Launch students straight into the **Oscillators** practical. In adapting the intial flowchart, they will learn how to use the pause, processing and decision instructions. Note that the pause instruction uses decimal notation, not hex as in the text book. Steps 5 and 6 are extension activities. Ask them to answer question 1 of the **Software** exercises before the next session.
2	Discuss their answers to question 1 of the **Software** exercises. Then get students started on the **Testing inputs** practical. Not all students will get as far as the extension activity of step 6. Ask them to answer questions 2 and 3 of the **Software** exercises before the next session.
3	Discuss their answers to the questions 2 and 3 of the **Software** exercises. Students should spend the rest of this session on the **Multipulse monostables** practical. In order to do step 5, students will have to realise that the number of flashes is set up in register b2 at the start of the program. Ask them to answer question 1 on page 156 of the text book before the next session.
4	Students should spend this session on the **Logic systems** practical. Steps 6 and 7 are extension activities which could be completed in a later session. Ask them to answer question 2 on page 156 of the text book before the next session.
5	Students should start this session by working through questions 3 and 4 on pages 157 and 158 of the text book. Those who finish with time to spare could resume the **Logic systems** practical. Ask students to revise **Microcontrollers** for a formal test next session.

Model Answers

1 (a) The binary word 1000 0000 is placed in the register S4.

(b) The voltage at the adc input is represented as a byte which is copied to register S0.

(c) 1F is 0001 1111.

(d) The byte in register S7 is copied to the output port and frozen onto Q_7 to Q_0.

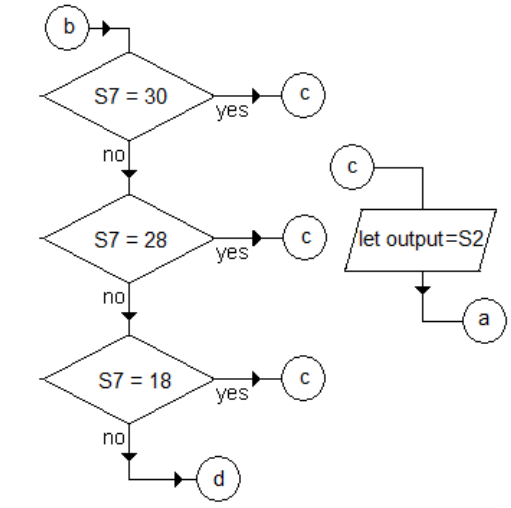
(e) While the voltage at adc is at or below 0.31 V, only the green LED glows.
Only the red LED glows when the voltage at adc is above 0.31 V.

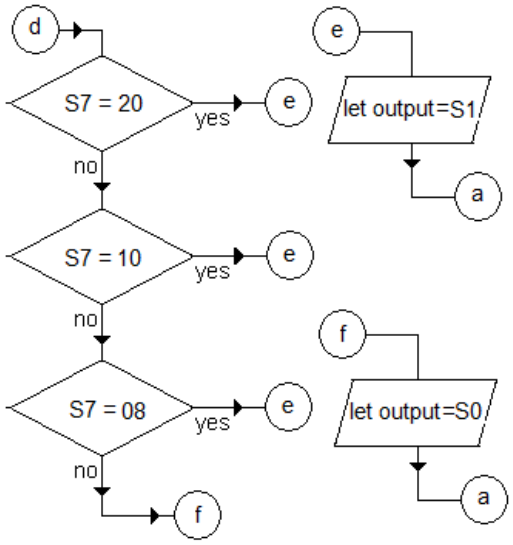
2 (a)

I5	I4	I3	binary	hex
open	open	open	00000000	00
open	open	closed	00001000	08
open	closed	open	00010000	10
open	closed	closed	00011000	18
closed	open	open	00100000	20
closed	open	closed	00101000	28
closed	closed	open	00110000	30
closed	closed	closed	00111000	38

(b) The state of the input port is compared with each of the eight bytes shown in the table, and the appropriate register (S0, S1, S2 or S3) copied to the output port to make the LEDs glow accordingly.

(c) This section tests the contents of S7 three times to find out if only two of the switches are being pressed. If so, program control passes to **c** and S2 is copied to the output port, making two LEDs glow.
When program control passes to **d**, the contents of S7 are tested another three times to find out if only one of the switches are being pressed. If so, program control passes to **e** and S1 is copied to the output port, making one LED glow.
Should program control pass to **f**, no switches are pressed, so S0 is copied to the output port and no LEDs glow.
Whatever byte is copied to the output port, the process starts over again as program controll passes to **a** ...

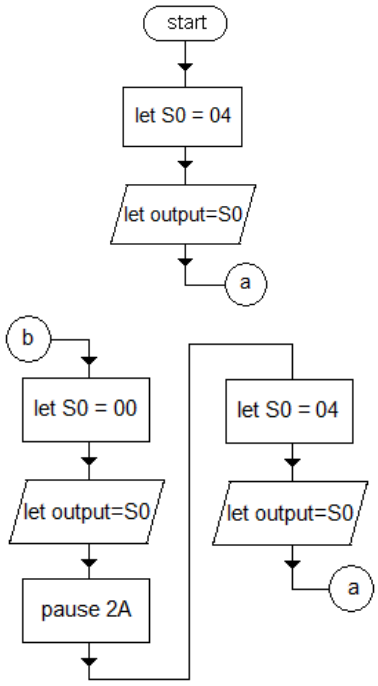




3 (a)

(b) The input port is copied to S6. If S6 contains 24 (00100100), the program control passes to **b**, otherwise, the input port is copied to S6 again and the test repeated ... until DCBA is 1001.

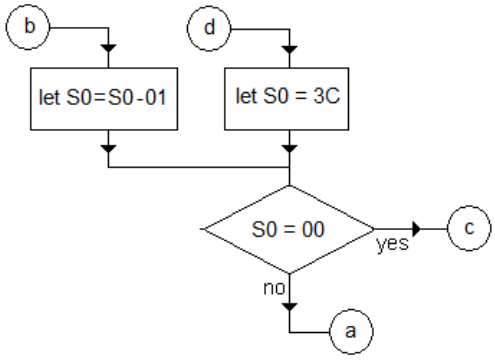
(c)



4 (a) The word 1011 0000 is copied to the output port for 42 ms. The word 1011 1000 is then copied to the output port for another 42 ms.

(b) The net effect is to make Q_3 do one cycle of an oscillation whose period is 84 ms, so it has a frequency of $1/84 \times 10^{-3}$ = 12 Hz.

(c)



(d)