EXAMPLE BUTTONS BOX

OBJECTIVE
This example has the following objectives:

- Review the use of parallel ports
- Illustrate the use of Port C for input and output
- Introduce the concept of shifting register and memory content
- Practice the use of the THRSim11 IO box for input and output

PROGRAMS
This example demonstrates the use of Port B for output and of Port C for both input and output. Port C is configured to have the pins PC0, PC2, PC4, PC6 for output. This is achieved using the Data Direction Register for port C (DDRC). Since the default setting for Port C is input, the pins that have to be set to output have their image set to 1 in DDRC. In our case, the bits 0, 2, 4, and 6 of DDRC are set to 1. Thus, the Port C pins PC0, PC2, PC4, PC6 become output, while pins PC1, PC3, PC5, PC7 remain input.

The program loops continuously, reading data from the some Port C pins, and sending data back to the other Port C pins. Simultaneously, the data is also sent to Port B.

i) PROGRAM BUTTONS BX

Instructions

(i) Load REGBAS in reg. X
(ii) Clear port B by storing zero in it

Select odd Port C pins for output

(iii) Load %01010101 in accA
(iv) Store accA to DDRC

Main loop

(v) Load Port C to accA
(vi) Shift accA to the right
(vii) Store accA to Port C and Port B
(viii) Branch back to (v)
(ix) SWI

FLOWCHART AND CODE
The program flowchart is not provided, since is simpler than in previous examples. The students are encouraged to use the space provided for filling in their own flowcharts of the program. The essential code for this program is shown to the right of the flowchart. The file Buttons_bx.asm is the result of incorporating this code in the standard template.
EXECUTION
a) Before you run the program, verify the Port C pins functionality. In the THRSim11 IO box, toggle up the switch PC3. You will see that the corresponding bit in Port C register, bit 3, becomes 1 immediately. Your screen looks like this:

![THRSim11 IO Box Screenshot]

b) Toggle down the switch PC1 in the IO box.

c) Run the program. The program will loop continuously between label BEGIN and opcode BRA.

d) With the program running, toggle up the switch PC3 in the IO box. Immediately, bit 1 in PORTC will switch to one. Simultaneously, pin 0 in PORTB will become 1, and the light PB2 in the IO box will lit. Press the stop button and capture the screen. Your screen should look like this:
e) Try to toggle other PC switches in the IO box. You will notice that now only the odd switches (PC1, PC3, PC5, PC7) are producing the setting of the corresponding bits in PORTC register. The other switches (PC0, PC2, PC4, PC6) can be toggled, but with no effect on the PORTC register. This illustrates how DDRC controls PORTC register.

f) Restart the program by pressing the Run button. With the program running, try to toggle the other switches. You will notice that the toggling of the odd switches (PC1, PC3, PC5, PC7) has the effect of switching on the even Port B lights (PB0, PB2, PB4, PB6). This is the result of the data from the odd Port C pins is loaded in accA, right shifted, and sent back to Port B.

WHAT YOU HAVE LEARNED

In this example, you have learned:

- The use of PORTC for input and output
- The concept of shifting a register
- The use of the THRSim11 IO box for both input and output
- New words and expressions: toggle switch; shift a register, shift data.