COMPUTER ORGANIZATION   Voluntary practice 2021-22

SQUID GAME

This practice is based on the TV series of the same name, and to be more specific, in the first game. Basically, if the “Green light” is ON, you can run (by pressing the button 1) and you have to stop if the red light is ON. If we run when the red light is ON, then you will be eliminated. There are two players, each one with a different button.

Once your program is executing, the green led furthest to the right should light up (this led will be called “the green light” from now on), and it is the reference to go ahead. To go ahead the player 1 has to press the button 1 in such a way that the next led will turn ON (and the previous one will turn OFF). To be specific, the patter is -,-,-,-,-,G (starting point), button 1 → -,-,-,-,G,G, button 1 → -,-,-,Y,-,G, button 1 → -,-,Y,-,-,G button 1 → -,R,-,-,-,G, button 1 → R,-,-,-,-,G. Thus, after pressing the button 1 five times, the red led at the end will be reached and you have passed the test and you are safe. If this happens, all leds will flash with a cadence of 200 ms ON, 200ms OFF. The game is over.

After the first player has pressed the button 1 for the first time, the second player can press the button 2 at any time. If this happens, the green light will turn OFF and the red led at the end will turn ON for 0.5 seconds (we call “red light” to this led from no won). After these 0.5 seconds, the previous state should be recovered if the player 1 did not press the button 1. The player 1 does not have to press the button 1 is the red light is ON at any time; otherwise, he is eliminated and a steady 250 Hz sound should be output.

On the other hand, there is a timeout of 5 second from the first time that the button 1 is pressed to the player reaches the extreme red led (in other words, you have 5 seconds to go through).

This exercise involves 2 points out of 10 of the final grade of chapter 4.

EXTRA POINTS (up to 2): add some melodies to the game, whatever you want. For example at the beging of the game and/or the end of the game, or hear a sound when the red light is ON … (suggestion: make the original song of the series, for example when the doll is playing …)

Ticks and suggestions

1. To prevent button bouncing you can add a delay of 200ms when you detect that a button is pressed. For example

```
x:        ldr r0,=GPBASE
          ldr     r1, [r0,#GPLEV0]
          tst     r1,#0b00100
          bne x
pressed: bl wait200
@         Continue here with the operation required after pressing
```

   where wait200 is a routine for waiting 200 ms.

2. You can create macros to facilitate the editing task. Macros are not subroutine calls but pieces of code that are repeated and that are replaced by some symbolic name at

the editing level. For example, a macro to turn on the red led (position 0x200) can be this (you can add it to the symbolic.inc file):

```
.macro red1ON
        push {r0,r1}
        ldr r0,=GPBASE
        ldr r1, =0x0200
        str r1,[r0,#GPSET0]
        pop {r0,r1}
.endm
```

(you can create a macros for turning OFF the red led in the same way and for the rest of the leds). In this way, instead of writing the instructions to turn on the red led you can write red1ON.

3. For the timeout of 5 s. you can insert a call to a routine that tests the time just in the middle of every checking button loop. To do that, once you detect the the button 1 is pressed for first time, then read the timer (port STCLO) and set a final count on a variable or register with the amount just read plus 5 seconds (5000000 microseconds). This variable or register will be tested in the aforementioned routine to check if we have reached the timeout.