

SISTEMI EMBEDDED

AA 2012/2013

Programming the DE2 Basic Computer:
playing with parallel ports

Putting into practice (1)

- Create a new “Nios II Application and BSP from Template” using DE2_Basic_Computer as target Nios processor (nios_system.sopcinfo)
 - Update DE2_Basic_Computer_Starter_Kit
 - Start coding from the template **seven_segments_template.c** available on the course webpage
- Basically, the template provides 2 functions:
 - *HexToSevenSeg()* showing a 4-hex digit number on a 4x 7-seg display
 - *Wait_ms()* generating a delay w/ ms resolution using a *for* loop to let the time pass

Putting into practice (2)

- Write a program that shows on the 7-seg display HEX3-HEX0 the sizes in number of bytes of *long*, *long*, *short* and *char* integer data types
- Do they match with the fixed-size integer type definitions in `alt_types.h`?

Putting into practice (3a)

- Write a program that turns on a single green led among LEDG7-LEDG0 and makes the position of the on-LED rotate with a period of around 500 ms. Make the activation and direction of the rotation controllable by the pushbuttons KEY3-KEY1, as follows:
 - KEY2 stops rotation, KEY3 and KEY1 activate rotation clockwise and counterclockwise respectively
 - The program must be sensitive to the edges originated by the pressure of the pushbuttons KEY3-KEY1

Putting into practice (3b)

- Hints:
 - Recognize pushbutton activations through the EVENT register of the relevant Parallel Port
 - Store the LEDG7-LEDG0 status on a 8-bit unsigned variable
 - Use <<, >> for left and right rotation (be careful to manage the all-zero situation)
 - Use a finite state machine (Moore model) to:
 - update the rotation state according to the KEY3-KEY1 events
 - generate the new LEDG7-LEDG0 status through a *switch* instruction that scans the rotation state
 - Use the *Wait_ms()* function to generate the rotation period
 - What does it happen if the LEDG7-LEDG0 status is stored in a signed variable?

Putting into practice (4)

- **Fast click game:**

- Detect which of KEY1 and KEY3 is pressed first after the turning on of one of the GREEN LEDS
- Make the interval time between two consecutive switching on of the LED random
- Make also the GREEN LED position random
- Signal which KEY has been pressed first using two different RED LEDs
- Display the number of times KEY3 has been pressed first on HEX3-HEX2 and KEY1 on HEX1-HEX0
- Use one SLIDER to start/stop the game and reset the scoring