Computer Organization

Iván de Jesús Deras Tábora

January 29, 2020

Period 1 2020

Keypad and Timer API

Keypad and Timer API

In this project you will have to implement the following functions:

Function prototypes

```
void delay_ms(uint32_t ms);
void keypad_init();
uint8_t keypad_getkey();
```

You'll have to include the header file `keypad.h` in order to have access to any of this functions.

`delay_ms` function

This function makes a delay for the specified time in milliseconds. To implement this function you'll have to use the timer available in the processor, which is available at address `0xffff0008` using MMIO (memory mapped input/output). In simple terms everytime you read this address you get the number of milliseconds since the processor started. To make a delay you'll be reading this address until the specified amount of time has passed.

`keypad_init` function

This function will initialize any global variables needed for keeping the keypad state.

`keypad_getkey` function

To implement this function you have to understand how the keypad works. In simulation mode the keypad is mapped to the keys up, down, left, right, 'q', 'p', 'b' and space bar. To read the keypad state we'll use MMIO also. The memory address for this is `0xffff0004`, the leftmost 8 bits of the value returned by this address represents the state of the keys individually, 1 is the key is pressed and 0 is the key is relesed. The following table shows the mapping between bits and keys:

7	6	5	4	3	2	1	0
space	b	p	q	up	down	right	left

`keypad_getkey` returns 1 if the left key is pressed, 2 if the right key is pressed, and so on. If no key is pressed it returns 0, if two keys are pressed at the same time it returns the one with the least bit value. One important thing to take into account, when a key is pressed and `keypad_getkey` is called for the first time it will return 1, but if the function is called again while the same key is still pressed it will return 1, if and only if, the time since the last call is greater than 100 millisecond. To do this you'll to use timer, but don't call `delay_ms`, because this will block the program.