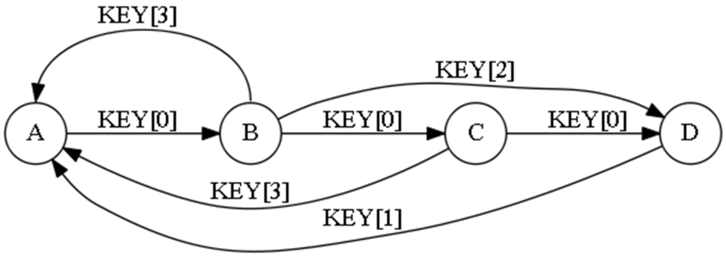
TÉCNICO LISBOA

Projecto e Controlo em Lógica Digital

3rd Laboratory

Work #1

Implement the state machine represented by the following diagram. Use a “standard” construction of finite state machines in Verilog.



When the FSM is on the “D” state, all green LEDs must be OFF.

Use the res LEDs to indicate the actual state of the machine.

Work #2

Implement a Finite State Machine for the control of an Oscilloscope.

A digital Oscilloscope has, tipically, several subsystems performing diferente tasks. Among the most important are the analog-to-digital conversion subsystem providing a continuous flow of data, the trigger subsystem, analyzing online the data flow to select interesting events, a temporary memory subsystem for buffering and a control subsystem responsible for the global management of the system which is typically implemented using a Finite State Machine.
It is intended to implement a State Machine for the control of an oscilloscope.

Functionality Description

The oscilloscope will start in an **idle** state. After receiving an **arm** command given by pressing **KEY[0]** the oscilloscope will enter in a **pre-trigger** mode in which it signals the memory subsystem to fill a buffer memory. After filling the buffer memory the oscilloscope will **wait for a trigger signal** from the trigger subsystem. During this time the buffer memory will be refreshed to contain the latest set of data by storing each new data value while deleting the oldest one. **After triggering** an order is given to the memory subsystem to finish filling the memory with post-trigger data. Then the oscilloscope will stay **waiting for readout** after which should return to **idle**.

Foresee a global reset that should bring the oscilloscope to the **idle** state.
Mimic the other subsystems by using switches, pushbuttons, etc. to emulate the control lines from the other subsystems, as, e.g., the trigger line, the memory state, etc.
Design the state diagram and then implement the state machine using a standard construction in Verilog.