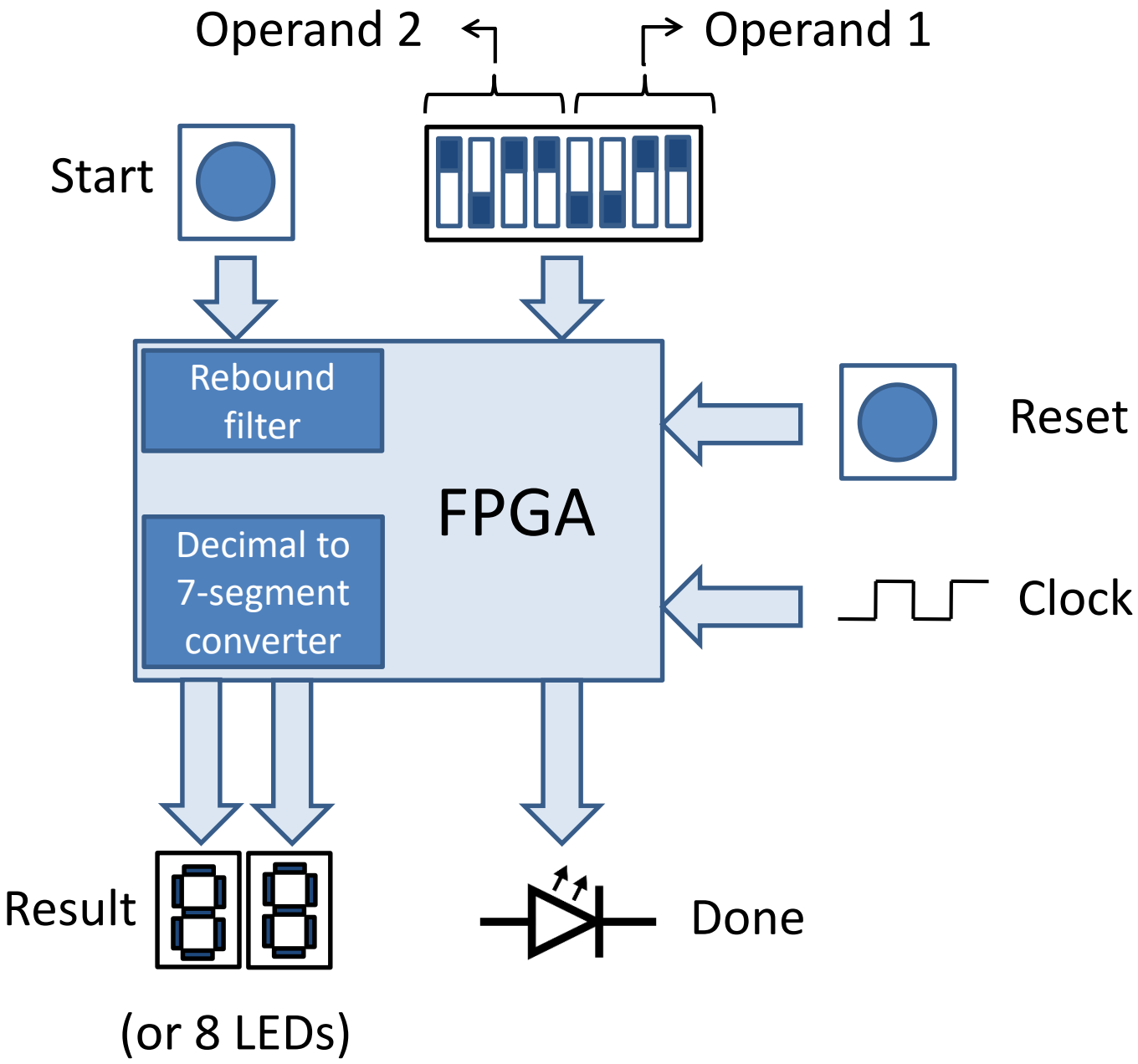
# Laboratory practice 4

Sequential 4x4-bit multiplier

# Objective

- Implement an iterative 4x4-bit multiplier:
  - To introduce the two operands we will use the switch bank.
  - We will use a ‘Start’ button and a ‘Done’ signal
  - The result of the multiplication will be shown in two 7-segment displays.

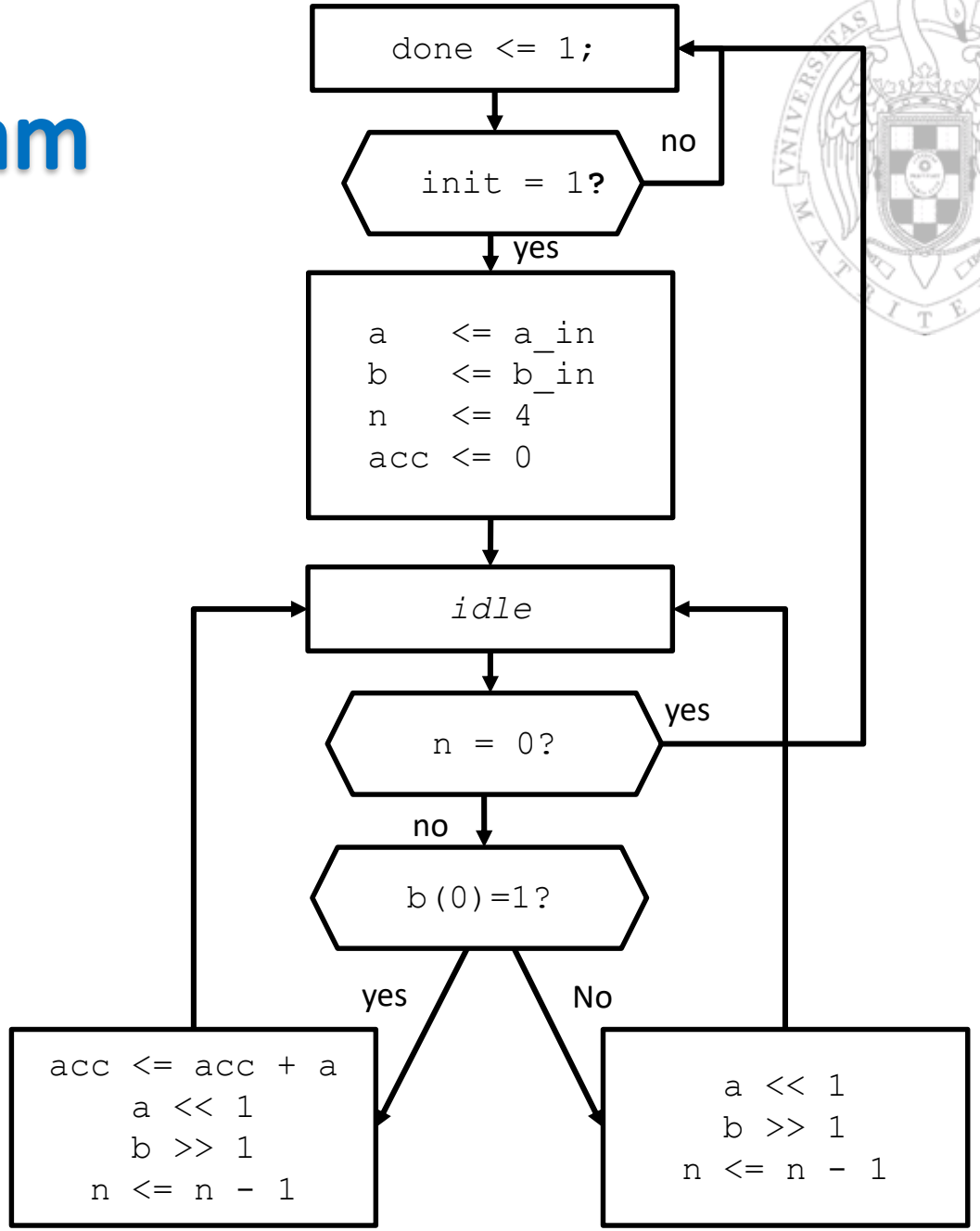

Operand 2
Operand 1
Start
Rebound filter
Decimal to 7-segment converter
FPGA
Reset
Clock
Result
Done
(or 8 LEDs)

# ASM diagram

- ASM multiplier:

```
a   = a_in;
b   = b_in;
n   = 4;
acc = 0;

while( n > 0 ){
  if( b(0) == 1 )
    acc = acc + a;
  a << 1;
  b >> 1;
  n --;
}
```



# Template for code

- I'm so generous that you can start the lab with a template available in the Campus Virtual.
- Files:
  - adder_std.vhd → Adder
  - adder_sub_std.vhd → Adder/subtractor
  - asynch_reg.vhd → PIPO register
  - definitions.vhd → Package with constants with indexes for accessing the control and status vectors
  - left_right_shift_reg.vhd → Shift register to the left/right
  - data_path.vhd → Datapath. To be completed
  - controller.vhd → Control unit. To be completed
  - multiplier.vhd → Top entity that instantiates datapath and controller
  - tb_mult.vhd → Testbench
- **Only data_path.vhd and controller.vhd should be modified. The rest of the files must remain unchanged.**
  - If you change the other files, the lab will not be valid.
- An additional Vivado project is available for displaying 4 values in the 4 7-segment displays, alternatively with low speed.
  - The 4 digits are entered by means of the 16 switches.
  - The reset is the central button (keep it pressed to make the system work).
  - You can adapt this to make the refresh faster (by considering contador_refresco (1 downto 0)) in order to make believe that the 4 displays are lit simultaneously.

# Grading

- Multiplier implemented and simulated as homework (use the test bench in the VC, **no modifications allowed**, even the UUT name and interface).
- Students must come to the laboratory with the **Data Path drawn (manuscript)** indicating clearly which are the **control, state and intermediate signals 0.15 pts**
- Students must show the multiplier working, and have to understand the implementation and functionality:
  - If it works properly in the FPGA, 0.15 pts
  - In case of just simulation working, 0.10 pts
- Advanced part 0.35 pts
  - If it works OK on the FPGA but the result is just shown in the LEDs, 0.2 pts
  - Additionally, if you use (and adapt) the module for using 2 7-segment displays to display the result, +0.15 pts
- Lab 4 is **NOT** recoverable