Project J: Pulse generator

- Specifications

In Fig. 1 there is the first draft sketch of a synchronous sequential machine to generate a periodic and programmable train of pulses. Each frame consist of a pattern of 16 configurable pulses and a time delay. The timing diagram in Fig. 2 represents how the frame is generated after triggering the start/stop button *ST*. At the end of each pulse array and delay time, there is an end-of-frame (*EoF*) pulse signal to indicate the end of a cycle and the beginning of a new one if the signal is set periodical. A single frame generation is possible when *Td* = “000”. For other *Td* values the repetition cycle is set between 1 s (“001”) and 7 s (“111”). It also may be interesting to represent de programmed *Td* delay using a 7-segment digit.

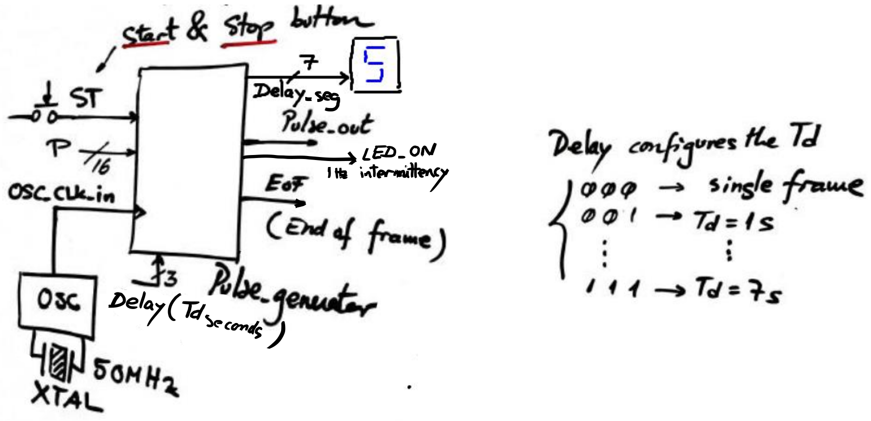

Fig. 1
Symbol of the pulse generator and the way to program the delay time *Td* that follows each frame by means of the *Delay* input vector.

When running continuously in a periodic mode, another *ST* pulse stops de machine when the frame ends. This feature means that if the button *ST* is clicked while the machine is generating the pulse sequence or delaying time, the process continues until it ends.

A pulse frame CLK of 800 Hz (CLK_F_SQ) controls the sequence of the 16 pulses, thus each bit duration is 1.25 ms.

The machine runs at a system CLK of 100 kHz (CLK_SYS_SQ).

To interface the start/stop key a debouncing filter is required.

A LED_ON generates a 1 Hz intermittency when the machine is running.

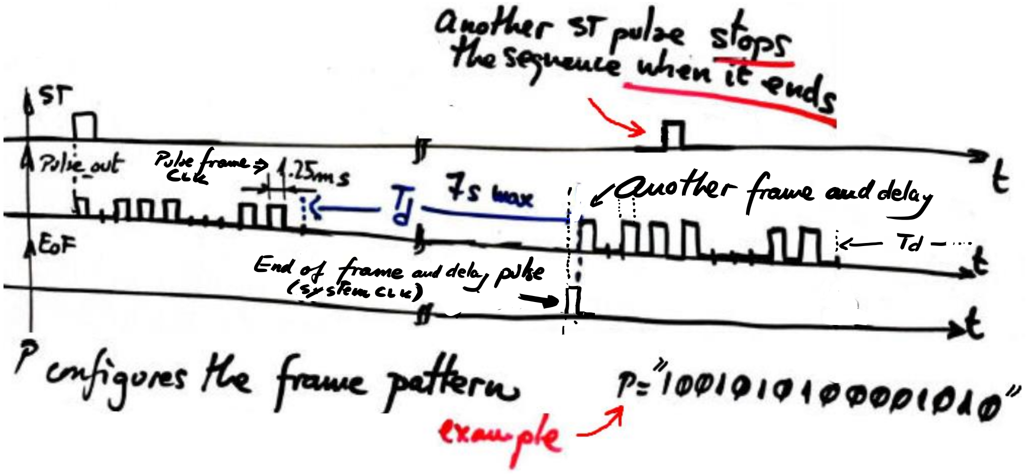
another ST pulse stops the sequence when it ends
ST
Pulse_out
Pulse frame CLK
1.25ms
Td
7s max
another frame and delay
t
EoF
Td
t
End of frame and delay pulse (system CLK)
t
P configures the frame pattern
P= "1001010100001010"
example

Fig. 2
Example of a timing diagram showing circuit’s activity. Start is a synchronous single pulse derived from the output of a debouncing filter.

1. Why a circuit like a *debouncing* filter is required to interface the *ST* start/stop push-button? Why a *RS_FF* may also be required? In which way may be possible to count real-time like *Td* = 3 s or *Td* = 7 s? Why a shift register may be used to organise the pulse sequencing?

- ❖ Planning (dedicated processor, plan C2)

In Fig. 3 there are examples of components that can be used when planning a dedicated processor architecture to complete the specifications. However, many other architectures using combinational and sequential components are feasible. It is also convenient to establish several design phases adding a new feature at a time.

2. Invent an architecture for the circuit. Describe the operation of the datapath proposed and the function of each module. Describe the control signals for the datapath, status signals from the datapath, external signals for the control unit, input data, output data, etc.

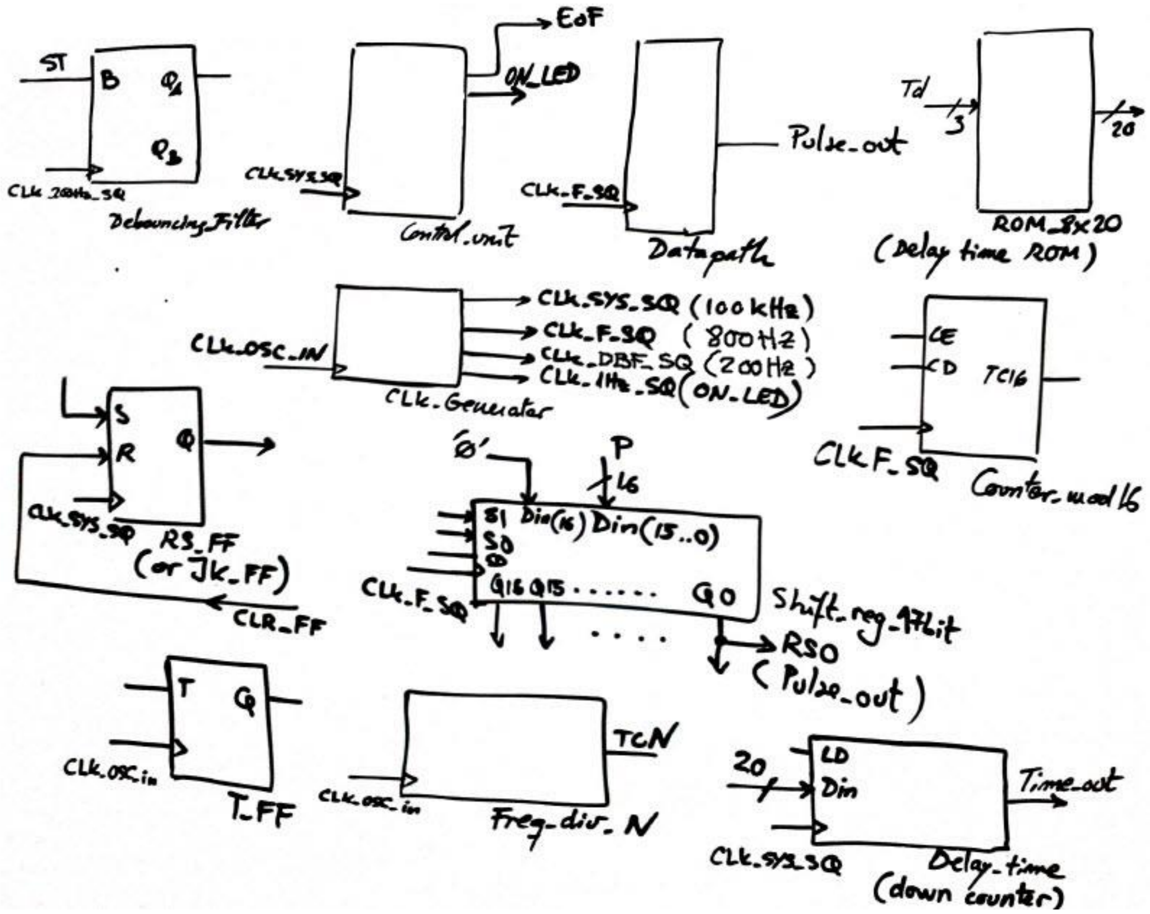

Fig. 3 Examples of components that can be used in the internal architecture of the system.

3. If the circuit uses a 50 MHz *OSC_CLK_in* from a quartz crystal oscillator, invent the *CLK_Generator* component using the generic architecture in P8 to produce the required squared signals. For instance: CLK_SYS_SQ = 100 kHz for the system CLK, CLK_F_SQ = 800 Hz for the frame pulses, CLK_DBF_SQ= 200 Hz for the debouncing filter, and CKL_1Hz_SQ = 1 Hz for the ON LED intermittency. How many VHDL files may it contain?

4. Describe the operation and internal architecture of the programmable delay time. How a binary down counter can be used for delaying time or a programmable timer?

5. How does the debouncing filter works? Suppose that we use the circuit designed in the P6 tutorial.

6. Planning the control unit.

   A. Draw the state diagram for the control unit FSM.

   B. Customise the general architecture of a FSM to the problem.

   C. Draw the CC2 truth table and its flowchart interpretation, so that it can be coded in VHDL in the usual CSD style.

   D. Draw the CC1 truth table and its flowchart interpretation, so that it can be coded in VHDL in the usual CSD style.

   E. Draw the internal circuit of the state register. How many data flip-flops (DFF) are required to implement it if we encode the machine in binary sequential or in Johnson?

7. Determine the number of D_FF used in the top circuit *Pulse_generator*.

- ❖ Development

8. Synthesise the complete circuit. Inspect and annotate the RTL and the technology views. Check that the number of D_FF used is correct.

- ❖ Testing

9. Write down the VHDL test bench translating approximately the inputs signals in the Fig. 2 diagram. Run a functional simulation to verify the circuit's operation.

10. Measure the maximum frequency of operation of this circuit for a given target chip.

Note: This application is for demonstrating the design possibilities of CSD circuits, as a practical way to finish Chapter 2. It is not necessary to finish it completely, but devoting the usual study time.

This kind of complex circuits may require several additional design phases to debug and eliminate errors, and they can be enhanced with many more features out of the scope of this introductory course.

Furthermore, the same circuit can be solved again in Chapter 3 as an application of microcontrollers. For instance using a PIC18F4520 and programming the source code in C language. Peripheral timers may be used to replace the down counter for delay timing and for pulse sequencing; external interrupts may detect the push-button clicks, etc.