Report: Logic

CLEMENT Andy

I/Introduction

In our analogue/digital converter project, the logic will process the various data received by the circuit elements (R2R, Sample & Hold, etc.) and control all the elements so that they all work in synchronization. To do this, the architect has defined several steps that need to be carried out. Firstly, for each bit of the R2R, the following tasks must be carried out:

- Set the test bit to 1**(MSB**)
- Replace the MSB value with Vcomp (Sample & Hold)
- Store the result of the comparison (0 or 1)

Before each cycle, all the bits should be initialized to 0 and the Sample value saved to ensure a stable voltage throughout the conversion.

II/Counter

We need to process 9 elements in turn (8 for each bit, 1 for Sample & Hold). To do this, we need an **LPM_counter** encoded on 4 bits with a set to 1 to count from 1 to 10 (we exclude 0 because the clock time is not the same in this state). To decode all the states properly, we use an **LPM_decode**. Each time we go to 10 (state10), the counter restarts from 1.

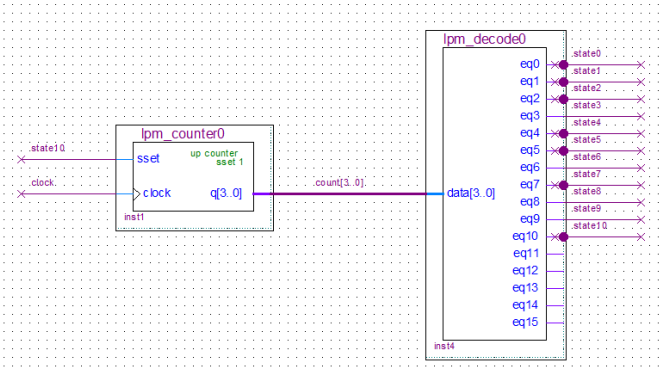

Figure 1 : Counter and lpm_decode logic diagram

This timeline is then obtained in Quartus simulation.

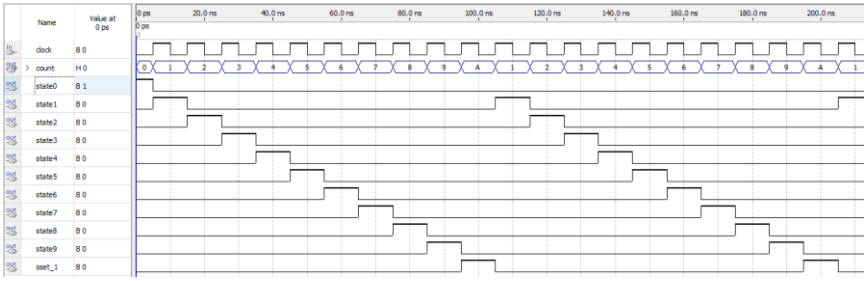

Figure 2 : Counter timeline

Note that the clock time is different in state 0 and that the counter returns to 1 after state 10.

III/Storing

The **data gating** method is used to store the Vcomp value on each bit. The components used are :

1. A Multiplexer**(21MUX**): selects the flip-flop return value or the Vcomp value according to the state value.
2. A D flip-flop**(LPM_FF**): stores the state 1 or 0 of the bit by means of feedback.
3. An OR gate : Transmits the recorded value to the rest of the circuit.

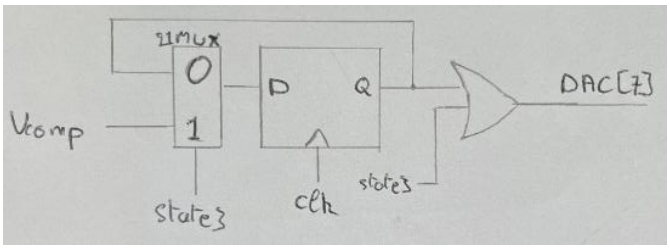

Figure 3 : Data gating logic diagram

With this arrangement, if state3 = 1 then the Vcomp value is stored, otherwise it is memorized. This device is then connected to each bit of the converter.

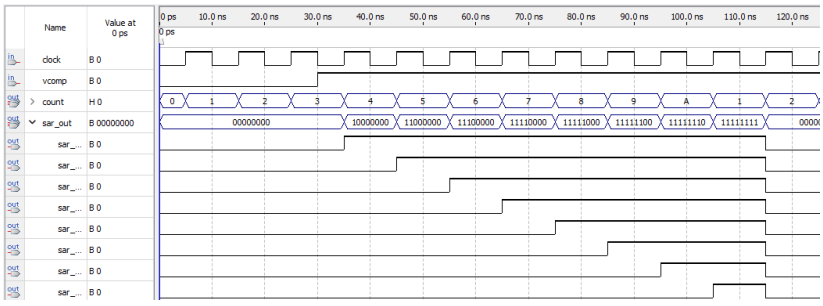

Figure 4Timeline obtained after data gating

Looking at the timeline, we can see that when Vcomp is high, the output bits**(sar_out[7..0]**) are recorded high. When A is reached, the counter resets to 1 and the sar_out[7..0] outputs are reset 1 clock stroke later.

IV/Signal transmission

Remember that only one wire is available for sending data, so each bit must be sent 1 by 1 (at each clock stroke) to be received correctly. To send the value of the bits that have just been retrieved by the data gating device, we store the states of each bit in another **LPM_FF**, which will free up the previous flip-flops so that they can store future states again. The bits are output on the **adc_out[7..0]** output **.**

Once recorded, the data must be transmitted using the **RS232** protocol, generating a start bit (high state), a start bit (low state), 8 bits of transmission and finally the stop bit (high state). Transmission then takes place over 10 clock strokes. This is done using a **161MUX** multiplexer which selects the data bit to be sent at each clock stroke.

At the output of the multiplexer, a **DFF** is added to absorb the delays caused by the logic in real conditions. Adding this flip-flop ensures that all the bits are sent at the desired clock stroke. We end up with the following diagram:

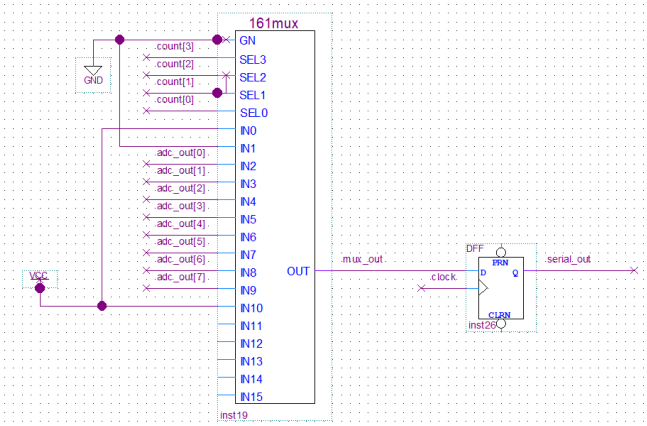

Figure 5 : Transmission block logic diagram

This configuration requires the data to be sent one clock stroke later than the end of the conversion. Note that the **serial_out** output is offset by one clock stroke relative to **mux_out**, because of the **DFF**, so transmission begins at the same time as the Sample.

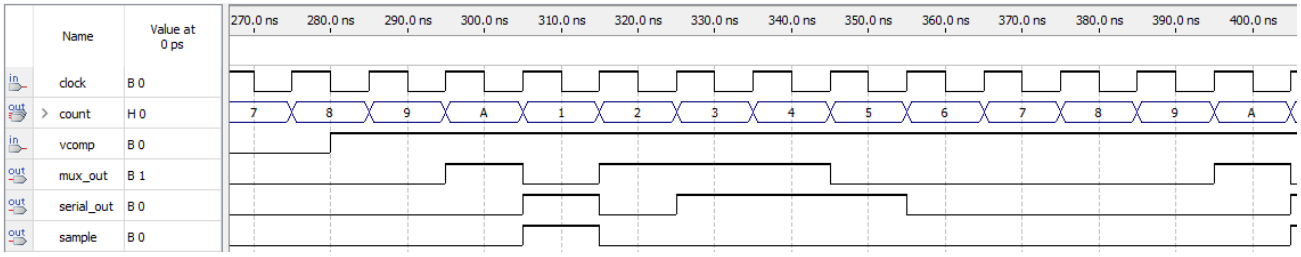

*Figure 6 : Transmission timeline with **mux_out** and **serial_out***

As the logic works well, we program the board with **MAX7000S** technology and inject the logic into the board. Testing the converter in real life shows that it behaves as expected.

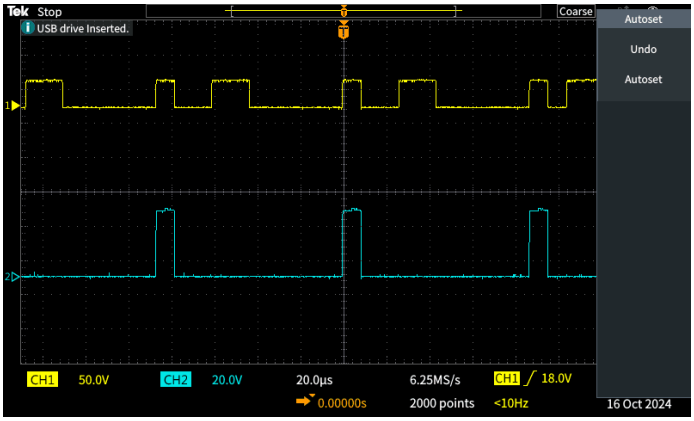

Figure 7 : Oscilloscope display of the signal obtained

The signal in yellow is **serial_out** and the signal in blue is the signal from the Sample. Note that the behaviour of the 2 signals corresponds to the simulation.