AD5791 20-bit DAC notes
Date: 6 Nov 2020
Author: Giorgos PAPADOPOULOS

After powering up the evaluation board with a single 3.3V power supply we can program the DAC with SPI link. I am using an Arduino Mega for the moment and the output works well, and the goal is to be able to use a Raspberry Pi 4, in order to test it before using it to control an ADC.

The DAC communicates with the user through a 24-bit information. In practice, using SPI this is achieved with $\overline{SYNC}$, SCLK, SDIN

- $\overline{SYNC}$: it is the equivalent name for Slave Select (SS) or Chip Select (CS). I will write SYNC from now to make my life easier.
- SCLK: Serial Clock.
- SDIN: this is the input of the DAC or from the Arduino/Raspberry point of view it is the MOSI (Master Out – Slave In) since the master is the Arduino/Raspberry and the slave is the DAC.

While in idle mode, SYNC must be held high. To start writing the 24-bit word to the DAC, the SYNC should change from high to low. Then, 3 8-bit packets will be send to the DAC. The word is composed by the two signals SCLK and SDIN. SCLK is going to provide 3 packets of 8 pulses each where at the falling edge of each pulse the SDIN will be high or low to define the bit as 1 or 0 respectively. The 24-bit word will be written in the ***input shift register*** of the DAC before it is transferred to the right address. The possible addresses are:

- DAC register: where we basically send the 20-bit word to set the DAC output value
- Control register: *not sure*
- Clearcode register: an option to set the DAC output at a predefined fixed value
- Software register: to program the DAC and its different configurations

I am using in practice only the DAC and control registers. The control register must be sent in the very beginning before start operating the DAC. I use a 24-bit word that works well with what I want to do in this is:

0010 0000 0000 0000 0001 0010
1st 2nd 3rd bytes

For more details and what every bit does check the reference documentation of AD5791.

- Before sending the 24-bit word one needs to change the SYNC from high to low and after sending the word it should go from low back to high. This is true for any message to send to the DAC (when writing in the control register or write to or read from the DAC register)
- Inside the DAC, the values will be written at the input shift register at the falling edge of each clock (SCLK) with respect of the value (high or low) of SDIN.

Write at DAC: After this one can set the output value of the DAC (**write** at the DAC) by sending a 24-bit word where the first 4 bits must be **0001**. The rest 20 bits, from 0 to 1048575, correspond to the desired output value. In this configuration the 0 corresponds to the negative reference of the DAC and the 1048575 to the positive reference. Operating the evaluation board of the AD5791, there is a daughter board producing the +/- voltage references at +10/-10 V respectively.

Read from DAC: In order to **read** what is written in the DAC register (through SDO or in the Arduino/Raspberry point of view MISO (Master In Slave Out)), the user must send a 24-bit word where the first 4 bits should be **1001** and the rest 20 can be 0 or whatever in my understanding. With the first 4 bits we enter the read mode (from the DAC) so the last 20 bits will not be written in the DAC register. The SDO signal will provide the information which can be saved and converted back to an actual binary input code to the DAC.