11/08/2023

Tests to understand how Sync All command functions

I set an AD9546 eval board to generate OUT0A=1Hz.

I then set M3 pin to receive a 1.8 V CMOS clock of 1/8Hz at J604. I chose M3 pin because this is the only Mx pin that can go into the AD9546 dc coupled. On the AD9546 eval board, all the M0, M1, M2 are ac coupled because people wanted to use them as outputs, and M5 and M6 pins go into the USB IC because they are muxed with SPI lines. M4 pin could also be used.

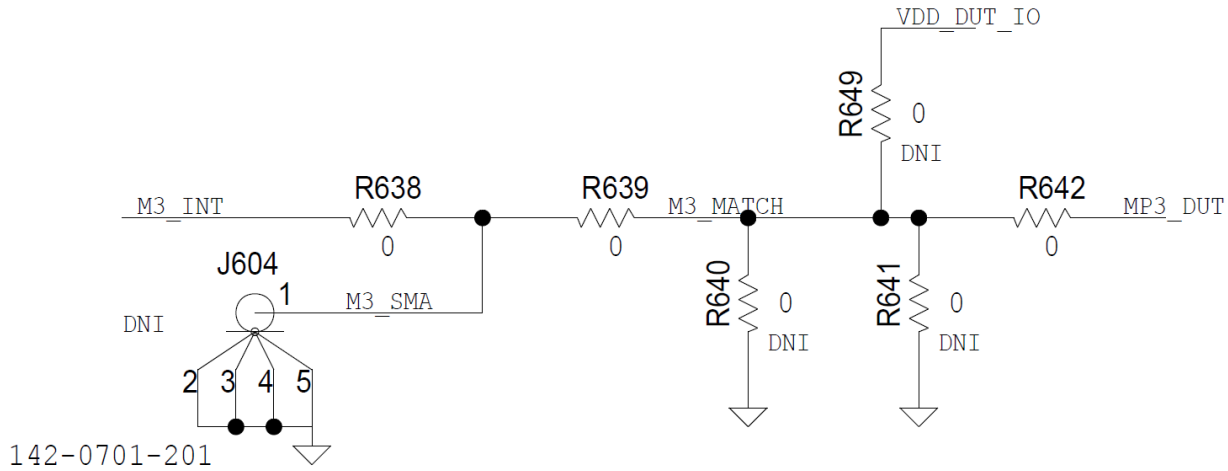

Figure 1. M3 pin schematic on the AD9546 eval board

I then set M3 pin as a control pin, active high, generating a Sync All command:

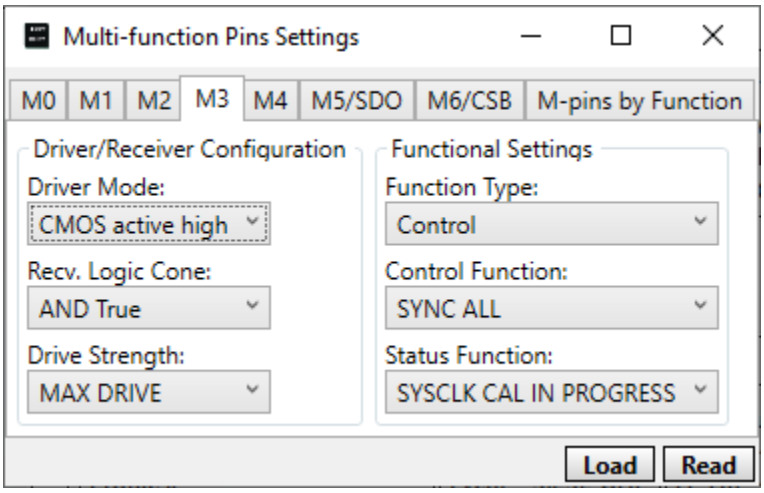

Figure 2. M3 pin setting in the eval software

On the oscilloscope, I was able to capture this:

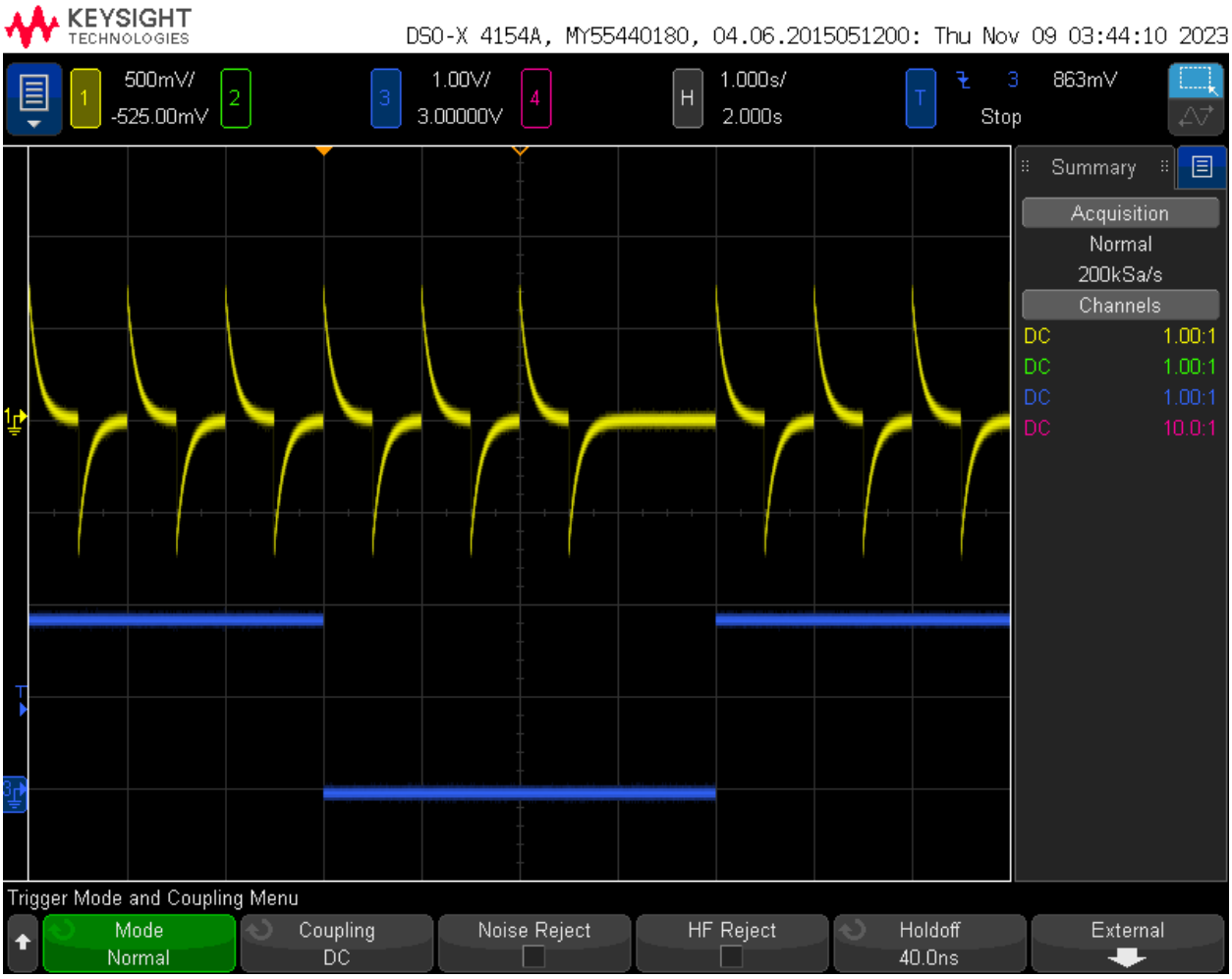

Figure 3. Channel 1, yellow is OUT0AP=1Hz. Channel 3, blue is the inverted pair of the clock provided at M3 pin

In Figure 3, when the clock at Channel 3 goes from high to low, the M3 pin is provided with a low to high transition. We can see this moment being equivalent with the bit 3 in register 0x2000 having a transition from 0 to 1. After three 1Hz clock cycles, OUT0A goes to 0V. When Channel 3 goes low to high, which means the M3 pin sees a clock going high to low, equivalent for the bit 3 in register 0x2000 having a transition from 1 to 0, OUT0AP starts to be generated.

Note that how we set the active level of M3 pin in Figure 2 matters. Not it is set to active high. If it was set to active low, the edges on which OUT0AP is manipulated would have changed.

Also note that the fact the clock goes to 0V is determined by the fact the output clock is ac coupled. If it was dc coupled, the OUT0AP would have gone to low level, and OUT0AN would have gone to the high level.

To verify the equivalence above with the state of the bit 3 in register 0x2000, I disabled M3 pin from doing a Sync All command:

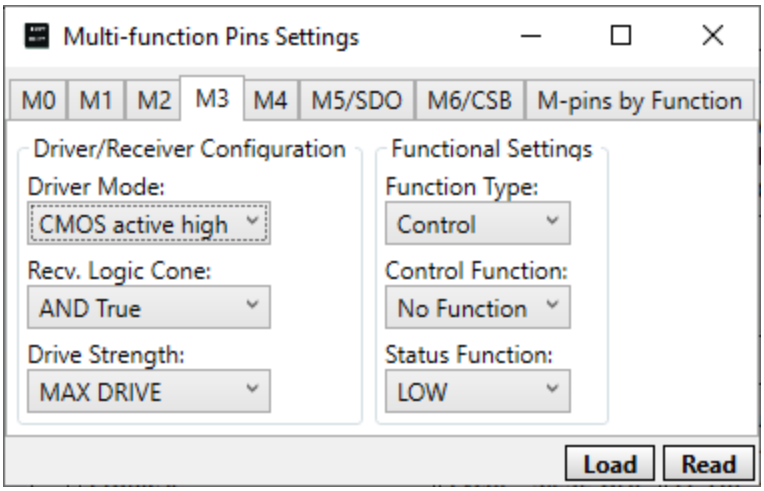

Figure 4. M3 pin is set as a control pin with no function

I then wrote using the Register debugger function in the eval software the value 0x08 into the register 0x2000. OUT0AP went to 0V.

I then wrote the value 0x00 into register 0x2000 and OUT0AP started to be generated.

Using a python script, I then manipulated bit 3 in register 0x2000 as follows:

- I wrote 0x08 into register 0x2000, followed by IO Update
- Waited 6 seconds
- I wrote 0x00 into register 0x2000, followed by IO Update

See behavior in Figure 5:

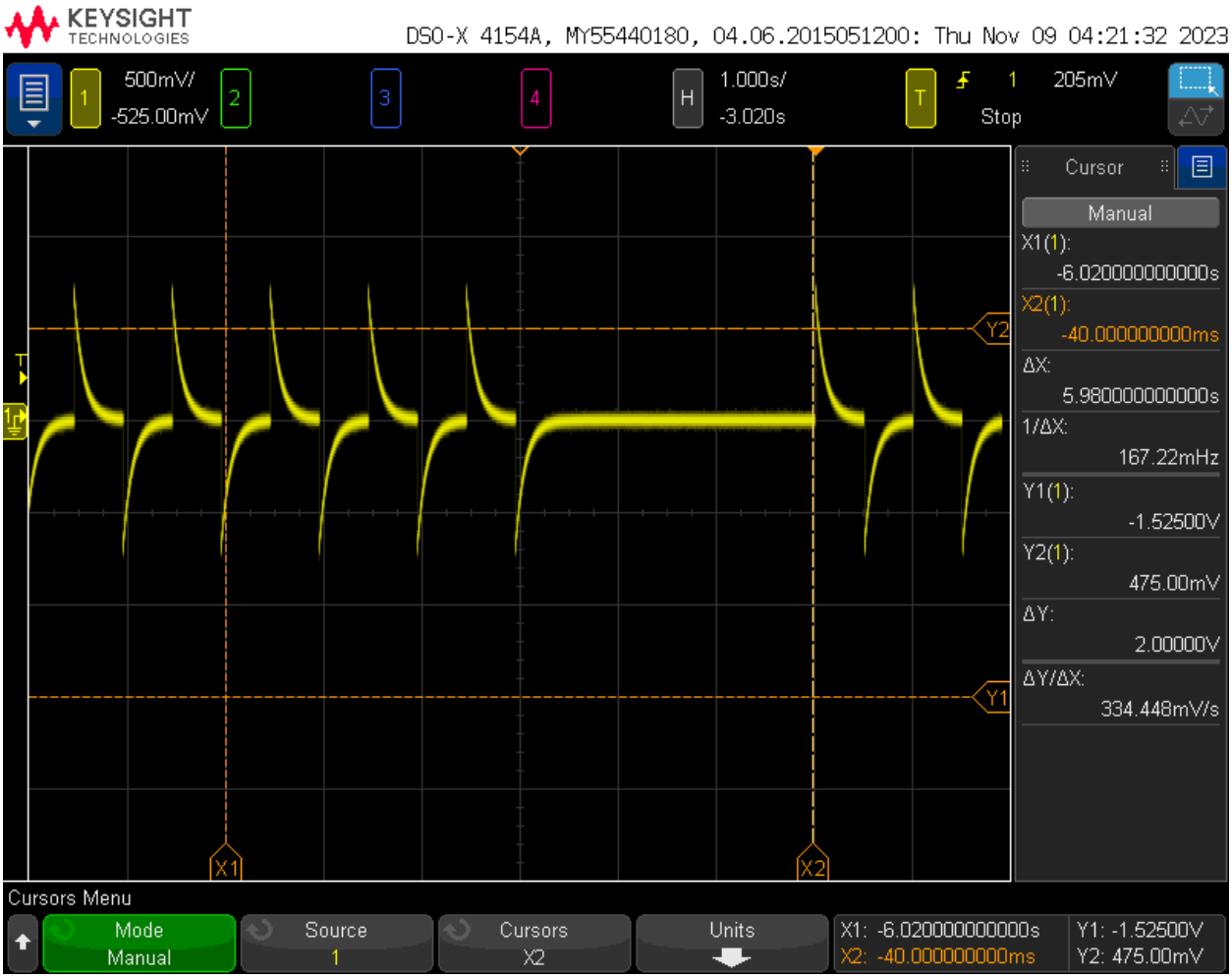

Figure 5. Channel 1, yellow is OUT0AP=1Hz

We know that when the bit 3 in register 0x2000 has the 1 to 0 transition, the OUT0AP is generated. Going back 6 seconds, we can identify the moment in which we set bit 3 in register 0x2000 to 1. We can see that OUT0AP goes to 0V three more than 3 clock cycles (or 3s) after the bit transitioned to 1.

To make sure the number of clock cycles after which the OUT0A goes to 0V is determinant, and not the time in which this occurs, I changed OUT0A to be 10Hz. I then enabled M3 pin to be the Sync All control pin.

I captured these figures:

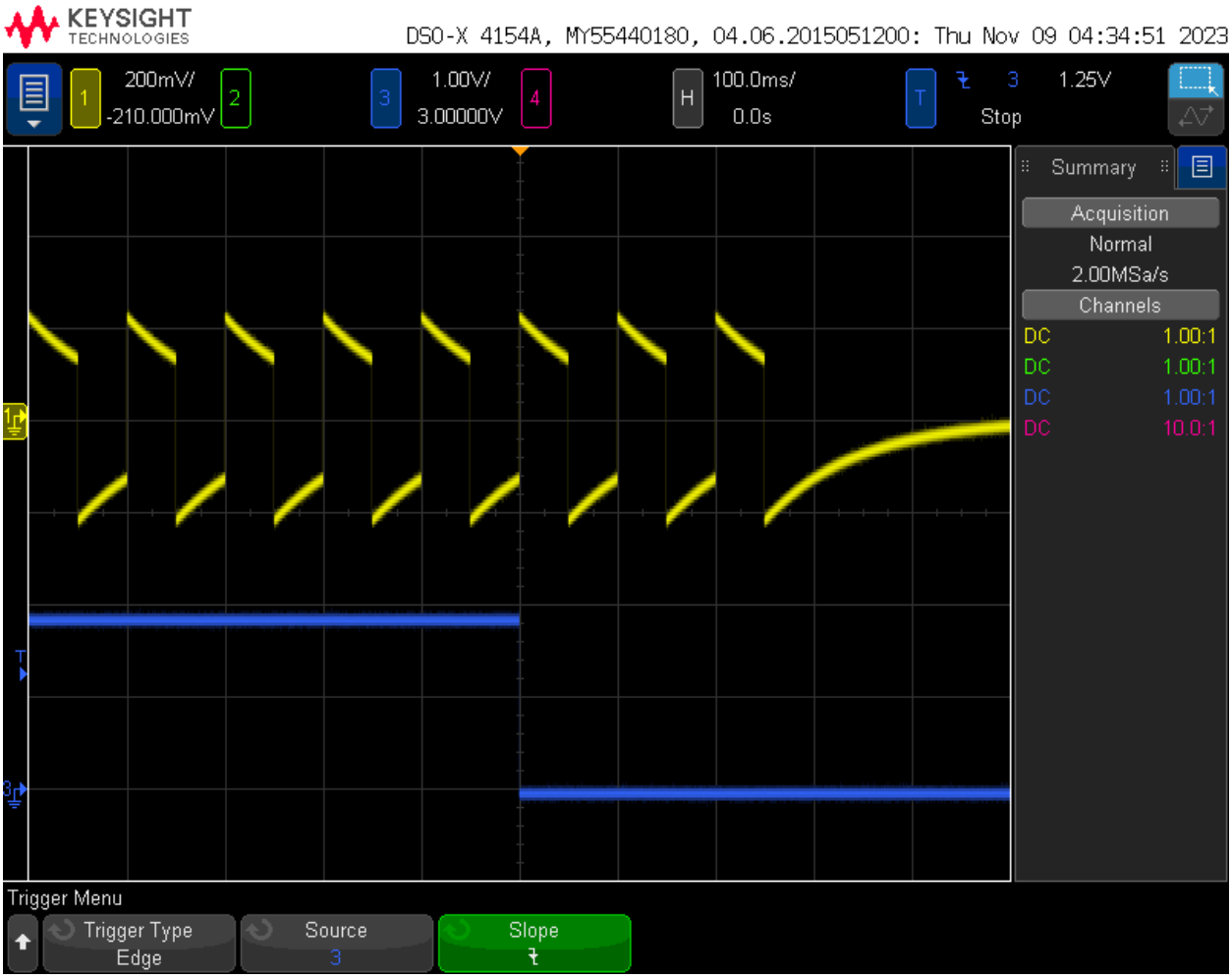

Figure 6. Channel 1, yellow is OUT0AP=10Hz. Channel 3, blue is the inverted pair of the clock provided at M3 pin, which means M3 pin went from low to high at the AD9546 input

Figure 6 shows that when M3 pin goes high (equivalent to bit 3 in register 0x2000 becoming 1), OUT0AP goes to 0V after approximately 3 clock cycles. This demonstrate that it is the number of clock cycles that matters, not the frequency of the clock.

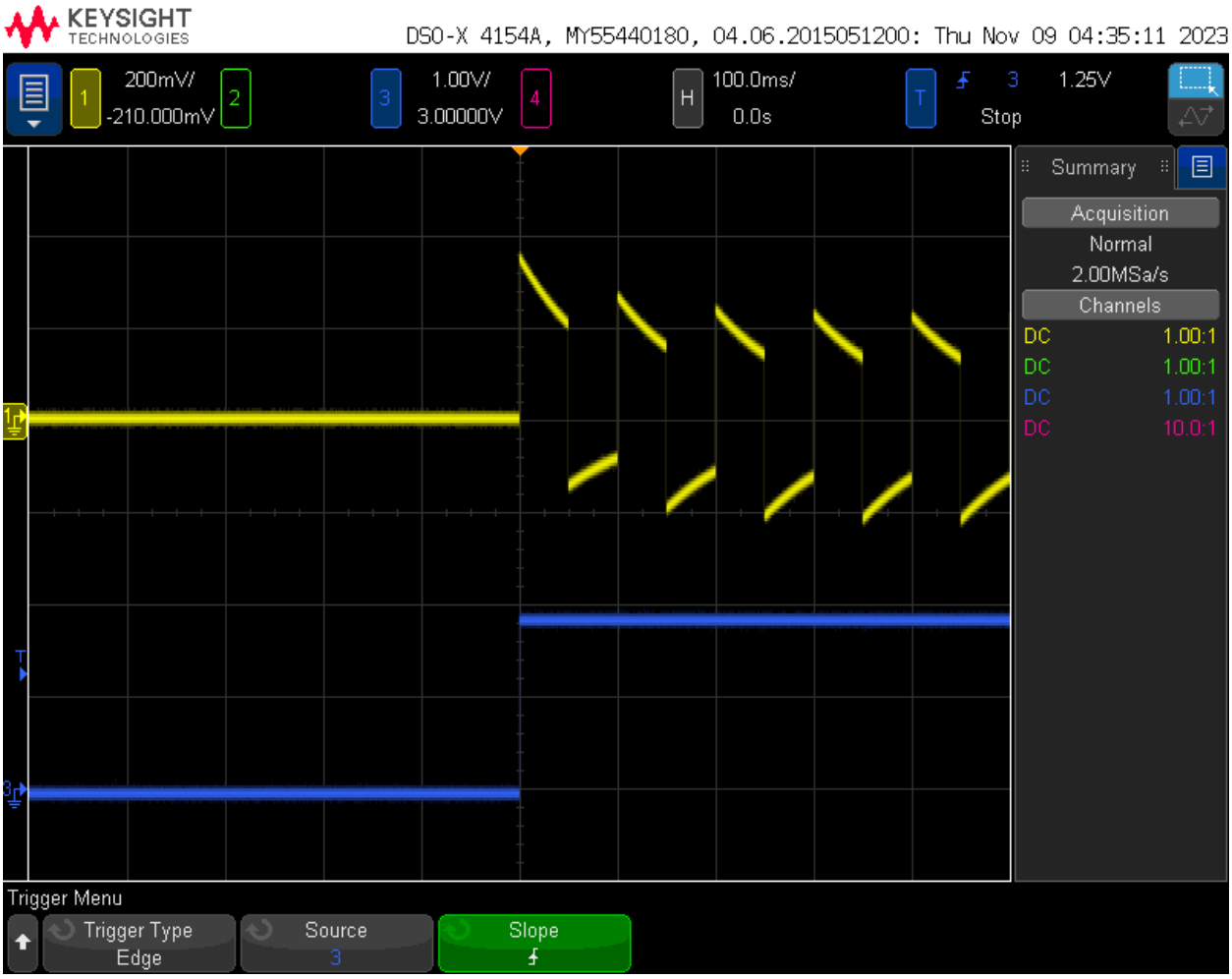

Figure 7. Channel 1, yellow is OUT0AP=10Hz. Channel 3, blue is the inverted pair of the clock provided at M3 pin, which means M3 pin went from high to low at the AD9546 input

Figure 7 shows that when M3 pin goes from high to low (equivalent to bit 3 in register 0x2000 becoming having a transition from 1 to 0), OUT0AP starts to be generated.

Conclusions

The bit 3 of register 0x2000 manipulates the Sync All command. It requires first setting this bit to 1, which stops the generation of the output clock after at least 3 clock cycles. The positive line of the output clock goes to the low level, while the negative line of the output clock goes to the high level. Then it requires clearing the bit to 0, which immediately starts the clock generation with a positive edge first.