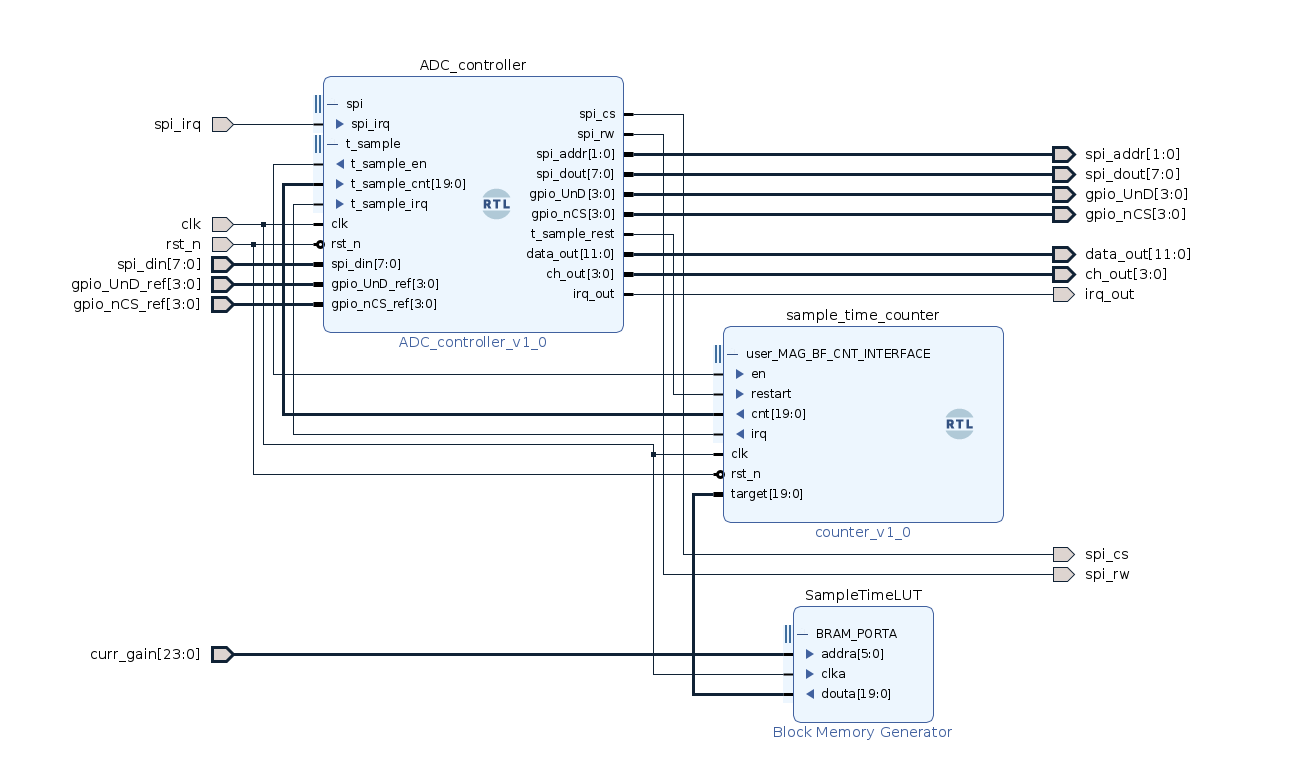
# Tests of ADCControl Submodule Functionality

The aim is to verify that the submodule is working as expected. The tests are divided into simulation tests and physical tests.

## Intended Functionality

The intended functionality of the ADCControl submodule is

1. To sequentially exchange data with each of the magnetometer ADCs, one SPI transmission per ADC, continuously as per the FSM, and that the data sent to the SPI module matches config, DL, DH, cmd, DL, DH, cmd, etc. Additionally, the first 4 data cycle is configuration values for the ADCs and then changes to data
2. To read data from each transmission to an internal register and to generate an internal irq whenever the read axis matches the expected axis, regardless of magnetometer
3. To have a timer running which generates an irq when the next sample should be taken and to let the timer irq time be dependent on the gain (ROM loaded with artificial data)
4. To stop the timer upon timer irq and to start it again as soon as a sample from magnetometer 0 is received
5. To sequentially propagate the next sample to the output ports upon timer irq once for each magnetometer
6. To switch the expected axis to the next axis when all magnetometers have delivered a sample upon timer irq
7. To command each ADC to switch to the next axis as soon as the timer irq has been registered
8. When the ref GPIO values change, command the respective ADC to change the corresponding GPIO value
9. Update the GPIO output port three data cycle after the ADC has been commanded to change the value of the respective GPIO pin

## Simulation

Simulation tests are carried out to test each individual functionality above and finally to test the full submodule functionality.

### Simulation 1

The test tests functionality point 1, that data is exchanged with each magnetometer ADC sequentially, one SPI transmission per ADC, as per the FSM, and that the configuration phase executes correctly followed by the data phase.

The ADC\_controller module is simulated in isolation to verify this behavior.

The module is taken through the configuration phase. This is done by letting the module write to all registers of the SPI module, for then to wait 2560 ns (the SPI transmission time 10ns\*16clkdiv\*16bits), and then to read the received data. It is expected that the module shifts between the magnetometers in a sequential fashion after each transmission. The SPI CS address should changes accordingly. One cycle consisting of all of the four magnetometers is repeated 4 times (the full configuration phase). The config pointer is monitored as it should first increment from 0 and then settle on 4 for each magnetometer after the configuration phase. Then, four cycles of the data phase is executed to verify that the configuration phase is over and the expected data is written to the SPI module (manual mode, ch 0 select, as curr\_axis=0, not changed during this test).

The module is concluded to correctly behave as described under functionality point 1.

The test can be reproduced with tcl script ADControl\_test1.tcl sourced from inside the simulation window with ADC\_controller.vhd as top.

### Simulation 2

The test tests functionality point 2, that data from the SPI module is read to an internal register, and that the internal irq is generated when the received ch matches the current expected axis, regardless of the magnetometer.

The ADC\_controller module is simulated in isolation to verify this behavior.

The module is first taken through the configuration phase. Then the module enters the data phase. The module is taken through the data phase where fabricated data is given to the module from the SPI interface to simulate received data. In the first data cycle, the simulated read channel from the SPI message does not current axis (0, not changed in this test), and the module should not generate an internal irq. In the second cycle, the channel matches for all magnetometers the current expected axis 0, and the module is expected to generate an internal irq for all magnetometers. In the first data cycle, the data and channel varies for the magnetometers. In the second data cycle, the channel is 0 for all magnetometers, but the data varies. The data is expected to propagate to the internal registers of the module in both data cycles for all magnetometers.

The module is concluded to correctly behave as described in under functionality point 2.

The test can be reproduced with tcl script ADControl\_test2.tcl sourced from inside the simulation window with ADC\_controller.vhd as top.

### Simulation 3

The test tests functionality point 3, to have a timer running which generates an irq when the next sample should be taken and to let the timer irq time be dependent on the gain (ROM loaded with artificial data).

The functionality is tested with testbench ADCControl\_test3 which consists of the counter module configured as the sample\_time\_counter module in ADControl.

The module is initially configured with en=0, restart=0 clk=clk rst\_n=1, target=X”FFFFF” and state=0. The following procedure is then followed:

1. state=1. 10 clock cycles are passed verifying that the module doesn’t run when en=0
2. state=2. Then en=1, target=X”0001F”, the module runs for 0001F+ clock cycles, irq is monitored. An irq should be generated while the counter should not auto restart
3. state=3. Restart is triggered for one clock cycle, the cnt should restart immediately. The module runs for A clock cycles. Then target=X”0000F”, the module runs for 5+ clock cycles, irq is monitored to assert when cnt hits target, cnt should not auto restart.
4. State=4. Restart is triggered for one clock cycle, the cnt should restart immediately. The module runs for A clock cycles. Then target=X”00008”, the module should generate irq immediately.
5. State=5. Target=X”FFFFF”, restart is triggered for one clock cycle. The module runs for F clock cycles, then en=0. It is monitored that cnt stops incrementing. Then en=1, cnt should continue. Then target=X”0001F”. The counter continues, should generate irq when cnt hits target.

The module is concluded to correctly behave as described under functionality point 3.

The test can be reproduced with tcl script ADCControl\_test3.tcl sourced from inside the simulation window with ADCControl\_test3.bd as top.

### Simulation 4

The test tests functionality point 4, to stop the timer upon timer irq and to start it again as soon as a sample from magnetometer 0 is received.

The functionality is tested with ADC\_controller in isolation.

The module is taken through the configuration phase. Then the first cycle of the data phase is run until mag2. The timer irq is asserted. It is observed whether the enable pin of the the timer is toggled low and whether it is enabled again upon receiving a message from magnetometer 0. The timing is monitored. The data phase is continued for 1.5 cycles. The timer irq is asserted right after receiving a mag 0 message. Similar observations are made. Finally, the timer irq is asserted right before receiving a mag0 message.

The module is concluded to follow functionality point 4.

The test can be reproduced with tcl script ADCControl\_test4.tcl sourced from inside the simulation window with ADC\_controller.vhd as top.

### Simulation 5

The test tests functionality point 5, to sequentially propagate the next sample to the output ports upon timer irq once for each magnetometer.

The functionality is tested with ADC\_controller in isolation.

The same procedure as in simulation 4 is executed, however now constructed SPI data is added and the output ports data\_out, ch\_out and irq\_out are monitored to ensure that the correct data is written out.

Works according to functionality.

The test can be reproduced with tcl script ADCControl\_test5.tcl sourced from inside the simulation window with ADC\_controller.vhd as top.

### Simulation 6

Functionality point 6, to switch the expected axis to the next axis when all magnetometers have delivered a sample upon timer irq – this has already been verified in simulation 5.

### Simulation 7

This test tests functionality point 7, to command each ADC to switch to the next axis as soon as the timer irq has been registered.

The functionality is tested with ADC\_controller in isolation.

The same procedure as in simulation 5 is executed, however now the data written to the ADCs is monitored along with the current axis.

Works according to functionality.

The test can be reproduced with tcl script ADCControl\_test7.tcl sourced from inside the simulation window with ADC\_controller.vhd as top.

### Simulation 8

This test tests functionality point 8, that when the ref GPIO values change, the module must command the respective ADC to change the corresponding GPIO value.

The functionality is tested with ADC\_controller in isolation.

The same procedure as in simulation 7, however, now the reference GPIO values are varied during the simulation and the output values for the ADCs are monitored for GPIO.

Works according to functionality.

The test can be reproduced with tcl script ADCControl\_test8.tcl sourced from inside the simulation window with ADC\_controller.vhd as top.

### Simulation 9

This test tests functionality point 9, that the module must update the GPIO output port three data cycle after the ADC has been commanded to change the value of the respective GPIO pin.

The same procedure as in simulation 8, however, now the output GPIO values are monitored.

Works according to functionality.

The test can be reproduced with tcl script ADCControl\_test9.tcl sourced from inside the simulation window with ADC\_controller.vhd as top.

### Simulation 10

This test tests the full ADCControl hierarchy in order to verify expected behavior.

The test bench ADCControl\_test10 is simulated. The module is simulated for a long period in order to monitor the output behavior. Keeping the gain constant, the sample time should be constant. It is verified that 1) the timer generates interrupts when reaching the target, 2) that the timer automatically restarts, 3) that the timer disables until receiving a sample from mag 0, and 4) that data is outputted from each magnetometer when the irq occurs.

Works according to functionality.

The test can be reproduced with tcl script ADCControl\_test10.tcl sourced from inside the simulation window with ADCControl\_test10\_wrapper.vhd as top.