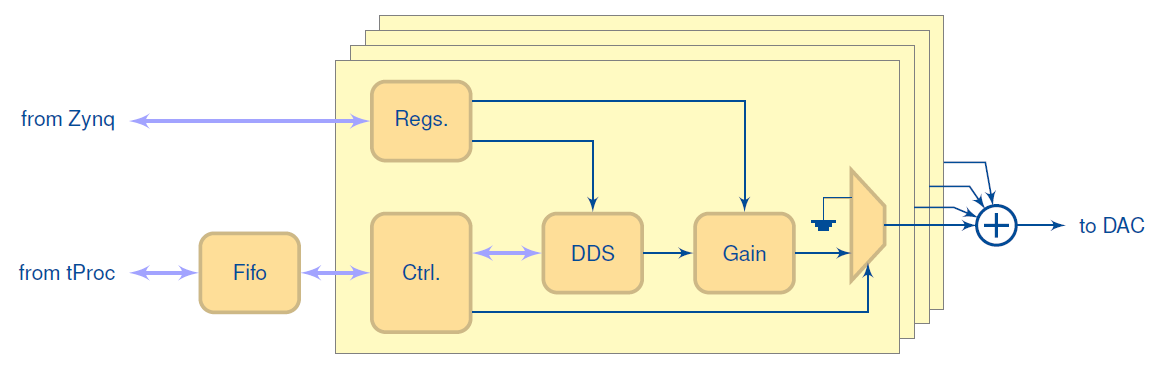
**axis\_sg\_mux4\_v2**

Introduction

This ip is often used to play multiple readout pulses to multiple resonators to readout the state of multiple qubits at the same time, all using one DAC. This ip is often used in combination with axis\_pfb\_readout\_v2 (also introduced in this thesis) to demodulate the pulses coming back from resonator.

Some properties



* Four channels. Their outputs are summed and output through a DAC.
  + Frequency, phases and gains of the four channels can be independently controlled.
  + Play time of the four pulses cannot be independently controlled.
* No envelope memory.
* Available waveform style: const (square wave).
* Nyquist zone: 1 or 2.
* Phase1: 0 to 360 degree, resolution is 32-bits (step 360/232).
* Frequency (fs is sampling rate of DAC, and fdds = fs/4):
  + DDS of each channel2: from - fdds/2 to fdds/2, resolution is 32-bits (step fdds/232­­­).
  + RFDC’s mixer (Fine mode) 3: 0 up fs, resolution is 48-bits (step fs/248­­­).
  + DAC’s output: refer to *Sampling & re-construction* section.
* waveform length: no limit.

1: relevant qick functions: QickConfig::deg2reg(),

2: relevant qick functions: QickConfig::freq2reg(), QickConfig::freq2int(), QickSoc::set\_mux\_freqs(), AxisSgMux4V2::set\_freq(). Relevant parameters: gencfg['b\_dds'] and gencfg['f\_dds'] in QickConfig::freq2reg().

3: relavant qick functions: AbsSignalGen::set\_mixer\_freq(), RFDC::set\_mixer\_freq(). Relavant codes: “fstep = fs/2\*\*48” in RFDC::set\_mixer\_freq().

How to get started using it

We design a simple experiment on zcu216 to test this ip core. The experiment is as follow: We use one DAC (DAC2) to send out four frequencies, and feed the four frequencies into one ADC (ADC0), following the ADC we use an axis\_pfb\_readout\_v2 ip that is also introduced in this thesis to demodulate the four frequencies, as shown in the figure below.

|  |
| --- |
|  |
| Use one DAC2 to send out four frequencies, and feed the four frequencies into one ADC0 |

Next, we let DAC2 play four pulses at the same time with different frequencies, and then let the ADC0 acquire the signal. Following shows the acquired pulses:

|  |
| --- |
|  |
|  |

We see that the pulse’s shape looks good. The codes to reproduce the above results are here:

<https://github.com/Ri-chard-Wu/thesis/tree/master/codes/axis_sg_mux4_v2-test-216>

how to include it in firmware

IP core settings:

* N\_DDS=4

RFDC ADC settings:

* Interpolation Mode: 4x
* Samples per AXI4-Stream Cycle: 8
* Datapath Mode: DUC 0 to Fs/2
* Mixer Type: Fine
* Mixer Mode: I/Q->Real

RFDC ADC tile clocking settings:

Wirings: