Summary - AGC Complex

Name	agc_complex
Worker Type	Application
Version	v1.1
Release Date	March 2017
Component Library	ocpi.training.components
Workers	agc_complex.hdl
Tested Platforms	xsim, isim, Matchstiq-Z1(PL)(Vivado 2017.1 and ISE 14.7)

Functionality

The Automatic Gain Control (AGC) Complex component inputs complex signed samples, drives the amplitude of both I and Q input rails to a reference level, and outputs complex signed samples.

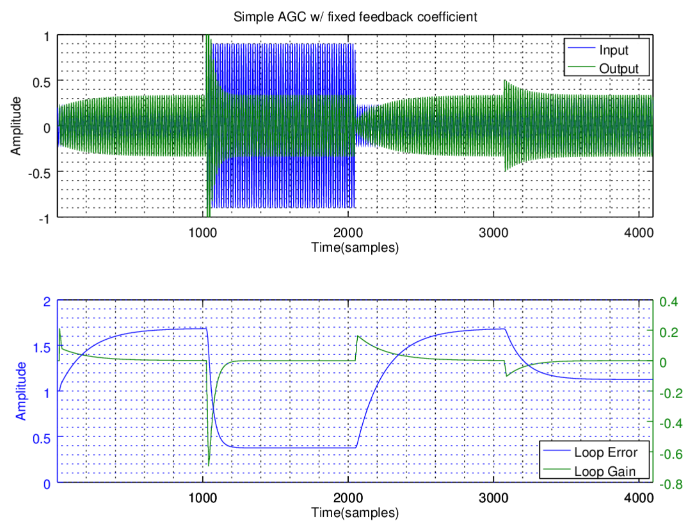


Figure 1: MATLAB AGC implementation with ref=0x1B26 and mu=0x144E

Worker Implementation Details

agc_complex.hdl

The response time and output level of the circuit are programmable, as is the ability to update/hold the gain differential used in the feedback loop. The size of the averaging window used for peak detection is build-time programmable using the `AVG_WINDOW_p` parameter, which is recommended to be a power-of-two to enable hardware division implementation with shift registers.

The `ref` property controls the desired output amplitude, while the `mu` property controls the AGC time constant, thus determining the response time of the circuit.

This implementation uses three multipliers per I/Q rail to process input data at the clock rate - i.e. this worker can handle a new input value every clock cycle. This circuit will produce output one clock cycle after each valid input, but the input-to-output latency is actually three valid clock cycles.

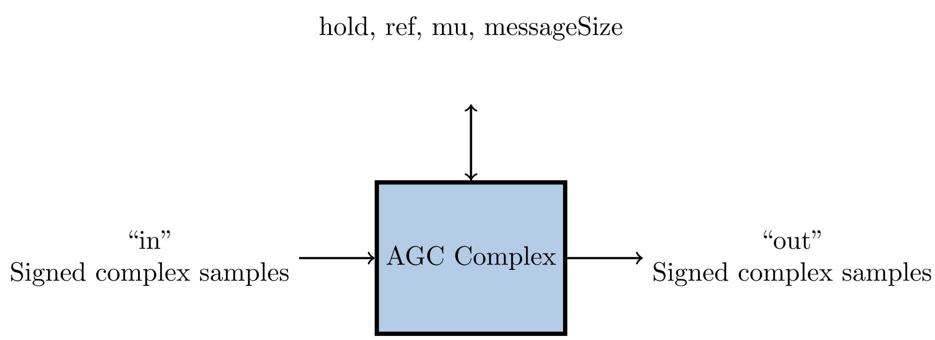
The AGC Complex worker utilizes the OCPI *iqstream_protocol* for both input and output ports. The *iqstream_protocol* defines an interface of 16-bit complex signed samples. The `DATA_WIDTH_p` parameter may be used to reduce the worker's internal data width to less than 16-bits.

Theory

The circuit is based upon Richard G. Lyons' "Understanding Digital Signal Processing, Third Edition" Automatic Gain Control (AGC) circuit found on Page 783. The text may also be found online here: DSP-Tricks-A simple way to add AGC to your communications receiver design. Lyons' circuit in Figure 13-76a implements the AGC function with a feedback loop on y(n) that consists of a magnitude operation to remove the sign, a comparator against the reference level "ref", a multiplier that uses "mu" to control the amplitude of the feedback signal (and thus the response time), and finally an accumulator. This implementation uses a peak detector in place of Lyons' simple magnitude operation. From Lyons: "The process is a nonlinear, time-varying, signal-dependent feedback system. As such, it's highly resistant to normal time-domain or z-domain analysis. This is why AGC analysis is empirical rather than mathematical ..."

Block Diagrams

Top level



State Machine

No finite-state machines (FSM) are implemented by this worker.

Source Dependencies

agc_complex.hdl

- training_project/components/agc_complex.hdl/agc_complex.vhd
- training_project/hdl/primitives/prims/prims_pkg.vhd
  training_project/hdl/primitives/prims/agc/src/agc.vhd

Component Spec Properties

N/A

Worker Properties

agc_complex.hdl

Type	Name	Type	SequenceLength	ArrayDimensions	Accessibility	Valid Range	Default	Usage
Property	`DATA_WIDTH_p`	UShort	-	-	Readable, Parameter	1-16	16	Worker internal non-sign-extended data width
Property	`AVG_WINDOW_p`	UShort	-	-	Readable, Parameter	4-256	16	Length of the averaging buffer; should be a power of two
Property	hold	Bool	-	-	Readable, Writable	Standard	false	Hold disables the gain differential feedback circuit, thus maintaining the current gain
Property	ref	UShort	-	-	Readable, Writable	1 to $2^{\mathtt{DATA\_WIDTH\_p}} - 1$	0x3FFF	Desired output amplitude expressed in percentage of full scale expected peak value in rms
Property	mu	UShort	-	-	Readable, Writable	1 to $2^{\mathtt{DATA\_WIDTH\_p}} - 1$	N/A	Feedback coefficient used to control the response time of the circuit; expressed as mu*fullscale
Property	messageSize	UShort	-	-	Readable, Writable	8192	8192	Number of bytes in output message

Component Ports

Name	Producer	Protocol	Optional	Advanced	Usage
in	false	iqstream_protocol	false	-	Signed complex samples
out	true	iqstream_protocol	false	-	Signed complex samples

Worker Interfaces

agc_complex.hdl

Type	Name	DataWidth	Advanced	Usage
StreamInterface	in	32	-	Signed complex samples
StreamInterface	out	32	-	Signed complex samples

Control Timing and Signals

The AGC Complex worker uses the clock from the Control Plane and standard Control Plane signals.

Performance and Resource Utilization

agc_complex.hdl

Table entries are a result of building the worker with the following parameter set:

- `DATA_WIDTH_p`=16
- `AVG_WINDOW_p`=16

Device	Registers	LUTs	Fmax	Memory/Special Functions	Design Suite
Zynq XC7Z020-1-CLG484	567	667	229.779 MHz	DSP48E1 = 6	Vivado 2017.1

Test and Verification

A single test case is implemented to validate the AGC Complex component. An input file is generated with a single tone at Fs/16 Hz, where Fs = 100 Hz, but applies 20% of the maximum amplitude to the first quarter of the file, 90% maximum amplitude to the second quarter of the file, 20% maximum amplitude to the third quarter of the file, and 30% maximum amplitude to the fourth quarter of the file. The complex waveform is then scaled to fixed-point signed 16-bit integers. Time and frequency domain plots may be viewed in Figures 2 and 3 below, respectively.

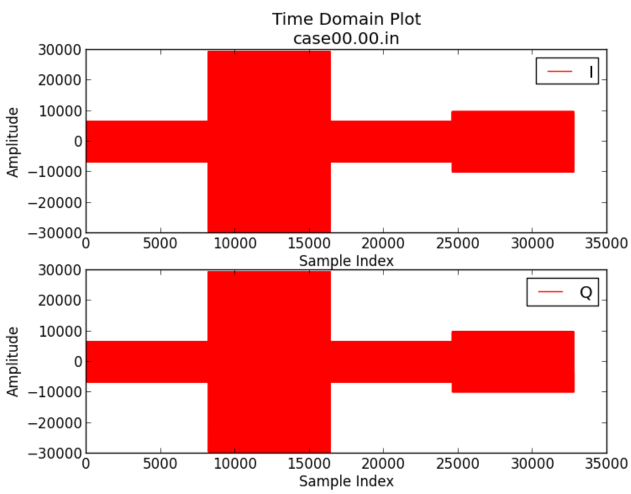


Figure 2: Time Domain Tone

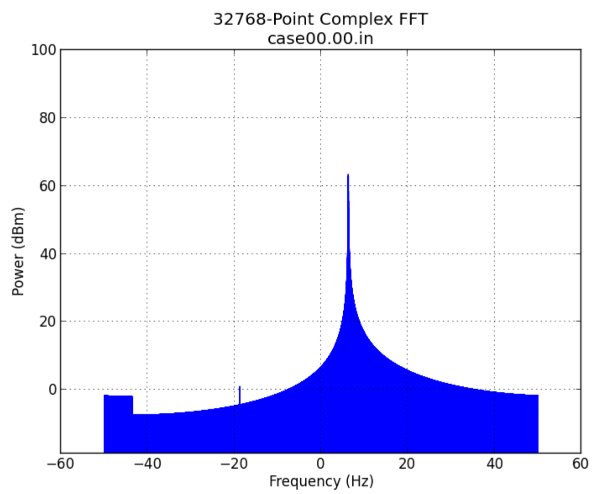


Figure 3: Frequency Domain Tone

For verification, the output file is first checked that the data is not all zero, and is then checked for the expected length of 32,768 complex samples. Once these quick checks are made a floating-point python implementation of the AGC is performed on the input data, which is then compared sample-by-sample to the output data. Figures 4 and 5 depict the output of the AGC Complex worker.

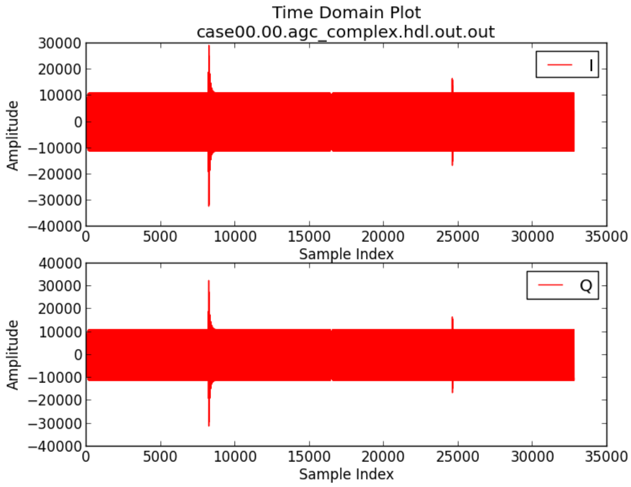


Figure 4: Time Domain Tones with AGC

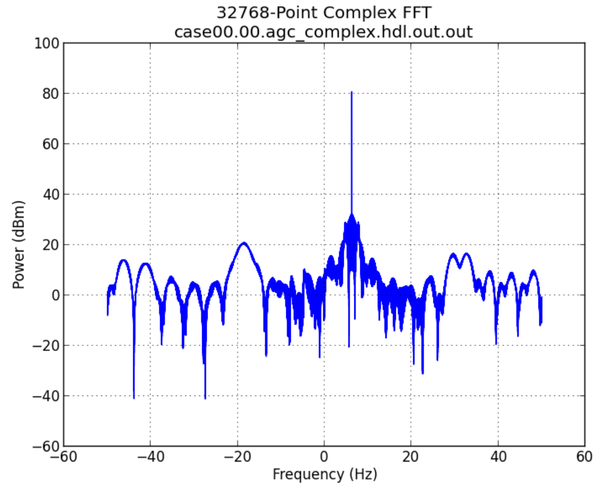


Figure 5: Frequency Domain Tones with AGC

References

(1) Richard G. Lyons. *Understanding Digital Signal Processing.* Third Edition. Pearson Education, Inc., Boston. 2001.

(2) Richard G. Lyons. (2011, March 29). *A simple way to add AGC to your communications receiver design.* Retrieved from http://www.embedded.com/design/other/4214571/A-simple-way-to-add-AGC-to-your-communications-receiver-design-.