

TESTING AND VERIFICATION OF A RF POWER AMPLIFIER CONTROL CIRCUIT

by

David Ciampi

A thesis submitted in the partial fulfillment
of the requirements for the degree of
Bachelor of Applied Science
in the School
of
Engineering Science

© David Ciampi 2002
Simon Fraser University
August 19, 2002

All rights reserved. This work may not be
reproduced in whole or in part, by photocopy
or other means, without the permission of the author.

Approval

Name David Ciampi
Degree Bachelor of Applied Science
Title Of Thesis Testing and Verification of a RF Power Amplifier Control Circuit

Dr. John Jones
Director
School of Engineering Science, SFU

Examining Committee

Academic Supervisor

Dr. Ljiljana Trajkovic
Associate Professor
School of Engineering Science, SFU

Technical Supervisor

Mike Schwartz
Conexant Systems, Inc.

Committee Member

George Austin
School of Engineering Science, SFU

Date Approved

Abstract

This thesis investigates the testing and verification of a digital power amplifier controller (PAC). The power amplifier controller is an instantaneous, closed-loop, handset transmit power controller. The PAC provides the control voltage to the power amplifier (PA) when a handset needs to transmit information. The verification comprises of testing the functionality of each sub-block of the loop and performance testing of the entire loop.

An integrated digital PAC versus the current analog PAC allows for many advantages. The digital PAC is more flexible for better power control over fabrication and temperature. The saturation characteristic of a digital integrator can be better controlled. Saving physical space and reducing part count provides benefit.

In this thesis, I performed tests to verify block functionality, system performance, and system stability. I used a test methodology for creating a test platform that provides reproducible test results with respect to functionality and performance. Using the ideas supporting this methodology, I was able to create a test platform that had the ability to create verification configurations from reusable models with minimal effort and maximum consistency. The test methodology I used to create test cases for each individual block was to isolate the block itself and to keep other system parameters constant.

Acknowledgements

I would like to thank my technical supervisor, Mike Schwartz, for his everyday support, comments and suggestions. He was always available for any type of questions that I had.

I am very grateful for the time and commitment my Academic supervisor, Ljiljana Trajkovic, for her support. Assisting and supervising a thesis from a long distance is difficult and her help was much appreciated. My thanks extend to George Austin for assisting me in my efforts to complete my thesis work. Different points of views can be combined to form quite diverse work.

My gratitude also extends to Ken Miller for all his efforts in getting me down to beautiful sunny California. I took many months of red tape but he pulled everything together, like always. “If you want to run with the big dogs, get off the porch.”

My appreciation and thanks to David Fisher for allowing me the opportunity to work on this aspect of the Air Force project. Assigning me the important responsibility of investigating the digital power amplifier controller gave me the confidence I needed to see the project through.

My fellow co-worker, Tim Irving, was always available to help whenever problems came up is definitely an unsung hero. Technically and socially he helped make each day quite an adventure.

Finally, my utmost appreciation goes to my parents, Felice and Rosaria, my two loving sisters, Antonella and Donatella, for their loving support throughout my studies.

Table of Contents

Approval	ii
Abstract	iii
Acknowledgements	iv
Table of Contents.....	v
List of Figures	vii
List of Tables.....	x
Abbreviations	xi
1 Introduction	1
1.1 Conexant Systems, Inc.....	1
1.2 Air Force Chip Set	1
1.3 Power Amplifier Controller	3
1.3.1 Analog Power Amplifier Controller	4
1.3.2 Digital Power Amplifier Controller	4
1.4 Road Map.....	5
2 Overview of the Testing System	6
2.1 Personal Computer System.....	7
2.2 Micro Controller	7
2.3 Digital Signal Processor	8
2.4 Integrated Analog	9
2.5 External Power Amplifier Controller	9
2.6 General Purpose Input / Output Bus.....	10
2.7 Internal Power Amplifier Controller	10
2.7.1 Intra-Frame Sequencer.....	11
2.7.2 Ramp Store.....	12
2.7.3 LPF / Integrator	14
2.7.4 Integrator with Limiter	15
2.7.5 Digital to Analog Converter Gain	15
2.7.6 Power Shaper	16
2.7.7 Digital to Analog Converter	17
2.7.8 RC Low Pass Filter	18
2.7.9 Power Amplifier.....	18
2.7.10 Coupler and Detector	19
2.7.11 Analog-To-Digital Converter.....	19
2.7.12 ADC Gain.....	19
3 Test Platform Development.....	20
3.1 Modular Verification Environment.....	20
3.1.1 Stimulus generator.....	21
3.1.2 Transaction Verification Module	21

3.1.3	Response Checker	21
3.1.4	Modularity Benefits	22
3.2	PAC Test Platform.....	22
4	Testing and Verification.....	24
4.1	Functional Specifications.....	24
4.1.1	Intra-Frame Sequencer.....	24
4.1.2	Digital-To-Analog Converter	25
4.1.3	Ramp Store.....	26
4.1.4	Analog-To-Digital Converter	28
4.1.5	LPF/Integrator	29
4.1.6	DAC Gain.....	35
4.1.7	Power Shaper	37
4.1.8	ADC Gain.....	38
4.1.9	Integrator with Limiter	39
4.2	Performance Specifications.....	40
4.2.1	Output Power	40
4.2.2	Output Burst Timing.....	43
4.2.3	Output RF Spectrum Due To Modulation.....	44
4.2.4	Output RF Spectrum Due To Switching Transients.....	46
4.3	Loop Stability.....	48
5	PAC Controller Program	52
5.1	Overview	52
5.2	GUI Interface.....	53
5.3	Command Object	54
5.3.1	Read Command.....	55
5.3.2	Write Command.....	56
5.3.3	Masked Write Command	57
5.4	Confirm Message Object.....	58
5.5	Debugging Capability	60
5.6	Test Saving Ability.....	60
6	DAC Power On Transient.....	61
6.1	Problem.....	61
6.2	Investigations	62
6.3	Solution	63
7	Conclusion And Future Work.....	64
7.1	Conclusion	64
7.2	Future Work	66
	References	67

List of Figures

Figure 1: The GSM/GPRS system solution is comprised of the CX815 baseband processor, CX20505 integrated analog device, CX20460 power management integrated circuit, CX74017 direct-conversion transceiver, and CX77304 tri-band, multi-slot power amplifier. Connections to ports, services, battery, speaker, microphone, memory, antenna and subscriber identity module (SIM) are shown with directional arrows that indicates data flow [2].2

Figure 2: The power amplifier controller is a closed-loop system. The forward path consists of an integrator, gain block, power amplifier, and coupler. The feedback path consists of a detector and gain block. Pout is the amount of power outputted by the power amplifier. An example of a simplified version of the actual controller.4

Figure 3: The system contains four major sub-systems with sub-blocks. The personal computer, micro controller, DSP, and integrated analog (IA) device are the major sub-systems. The arrows indicate data flow direction between blocks. The integrated analog device can select two power amplifier controllers. The internal PAC is discussed in this thesis.6

Figure 4: The RFTTestDOS message format consists of message header and message body. The message header consists of the message type, tag, and message length. The message body contains the data. The number of lines in the message body is related to the message length minus one.....7

Figure 5: The DSP to integrated analog control interface consists of a clock, control, control data, and request data lines. The control data line has a specific format that is configured differently for reading and writing [4].....8

Figure 6: The digital closed-loop PAC is slightly more complex than the generalized version shown in Figure 2. The IFS is not connected to any of the blocks because it controls the base timing of the system. The IFS provides no other purpose than timing of system. The PA, coupler, and detector in this figure are the same as in Figure 3. The dotted line shows the division between the elements that are found inside and outside the IA device.10

Figure 7: An example IFS control timing output. The actual output of the control signals that can be used to drive internal or external circuitry. An output of zero corresponds to a zero voltage level and an output of one corresponds to a positive rail voltage.12

Figure 8: An example Ramp Store timing output. The actual output of the Ramp Store used to drive the LPF/Integrator portion of the control loop.14

Figure 9: The Power Shaper transfer function is piecewise as shown. The forward path signal is shaped according to the transfer function to allow for high gains at higher input levels and no additional gain at lower input levels.....17

Figure 10: The Power Amplifier transfer function is described by the figure. The input to the PA is a control voltage ranging from approximately 0.9 volts to 2.2 volts. The PA saturates at the voltage of 2.2 volts. The output of the PA is power measured in the figure by dBm....18

Figure 11: The Modular Verification Environment consists of the creation of a cause and observing the effect.20

Figure 12: The PAC Test Platform based on the methodology described in Section 3.1.....22

Figure 13: IFS stimulus sequence containing the three PAC control signals Tx Enable, PAC Enable, and Ramp Enable.....24

Figure 14: DAC output with PAC enable signal. The DAC output is zero volts when the PAC enable signal is low. The DAC output is 0.3 volts when the PAC enable signal is high and the digital input is zero. The DAC output can vary from 0.3 volts to 2.4 volts depending on the digital input value.	25
Figure 15: Ramp Stimulus sequences for the two test cases. The first sequence tests the magnitude levels and the second sequence test durations.	27
Figure 16: LPF/Integrator verification setup uses digital loop to send actual digital values to the feedback path.	29
Figure 17: Impulse response for the LPF test case using a 250 nit impulse input with varying coefficient A of the LPF/Integrator block. The y-axis is the value in output value in nits and the x-axis the number of samples. Four curves for the four corresponding coefficient values are described in the legend.	31
Figure 18: Step response for the LPF test case using a 250 nit step input with varying coefficient A of the LPF/Integrator block. The y-axis is the value in output value in nits and the x-axis the number of samples. Four curves for the four corresponding coefficient values are described in the legend.	32
Figure 19: Impulse response for the integrator test case using a 250 nit impulse input with varying coefficient A of the LPF/Integrator block. The y-axis is the value in output value in nits and the x-axis the number of samples. Four curves for the four corresponding coefficient values are described in the legend.	33
Figure 20: Step response for the integrator test case using a 250 nit step input with varying coefficient A of the LPF/Integrator block. The y-axis is the value in output value in nits and the x-axis the number of samples. Four curves for the four corresponding coefficient values are described in the legend.	34
Figure 21: DAC Gain verification setup assumes the Ramp Store, ADC and LPF/Integrator block are functional.	35
Figure 22: Output Power verification setup requires the use of HP8922M GSM/DCS Test Set. The connection from the Device to the HP8922M is a standard RF cable.	42
Figure 23: Output burst timing example. The timing process of the transmit signal may cause interference into other user time slots if slot timing is not accurate.	43
Figure 24: Transmit burst time mask that output power must fit to pass performance testing [11].	44
Figure 25: Output RF spectrum due to modulation example. The modulation process may cause interference into other user channels and only specific levels of power are allowed according to the GSM standard.	45
Figure 26: Spectrum measurements test setup with the device and HP8922M Test Set. The Level Shifter provides the trigger for the Spectrum Analyzer. The RF Splitter provides the input signal to both the HP8922M and the Spectrum Analyzer, and is also terminated with a 50 Ohm connector.	45
Figure 27: Output RF spectrum due to switching transients example. The switching process of the transmit signal may cause interference into other user channels and only specific levels of power are allowed according to the GSM standard.	47
Figure 28: Loop stability verification setup. The diagram on the left illustrates the block diagram of the signals and gain blocks. The diagram on the right illustrates the actualization of the block diagram using an operational amplifier and resistors. The value of the resistors (R) is	

10K Ohms. The value of V_t sweeps for 0 mV to 50 mV. V_{out} and V_{in} will be the points of measurement.	49
Figure 29: Gain Margin value measurement. The value of the gain margin is determined by measuring the gain in dB when the phase is at zero degrees. The value of gain margin is 19.5dB.	50
Figure 30: Phase Margin value measurement. The value of the phase margin is determined by measuring the phase in degrees when the gain is unity. The value of phase margin is 92.15 degrees. The value of the loop bandwidth is 228kHz.....	51
Figure 31: PAC Controller Program GUI interface screen. The interface resembles the actual control loop structure.....	54
Figure 32: Ramp Store and IFS Store control interface screen. These two interface screens allow direct access to control the Ramp Store and IFS Store functions.....	54
Figure 33: Command Object properties and methods. The command object controls the request messages to the system.	55
Figure 34: Read command flow diagram. The read command is used when a register value needs to be retrieved for verification or monitor purposes.....	55
Figure 35: Write command flow diagram. The write command is used when a register value needs to be changed.	56
Figure 36: Masked Write command flow diagram. The masked write command allows for specific bits in a register to be changed.	57
Figure 37: Command Object properties and methods. The confirm message object is responsible for deciphering messages from device.	58
Figure 38: Confirmation Message format. The format of the message includes two header lines and any number of data lines up to the maximum allowed.	59
Figure 39: Confirm Message process flow diagram. The confirm message function extracts the message information to be processed and used.	59
Figure 40: DAC and Power Amplifier interface with control signal to clamp the DAC output to a value of zero volts.	61
Figure 41: Secondary control signal logic generator. A simple flip-flop is used to generate the control signal.	62
Figure 42: Output switch control logic. If the output switch is clamped the DAC output is held to a zero volt level and when the output switch is not clamped the DAC output is passed to the PA.	63
Figure 43: DAC power-on transient resolution results. The input control voltage is shown in the before and after diagram indicating a large reduction in overshoot amount. The resolution is based on delaying the control signal from turning on during the DAC power on sequence.	63

List of Tables

Table 1: An example IFS control sequence. The example shows the control of four of a total 16 possible signals and a total of five of the available 32 states. The duration of State 1 is the longest and the duration of State 4 is the shortest.	11
Table 2: An example set of IFS control store programmed values. The example shows the actual programmable values that are required to implement the example sequence in Table 1. Note the duration of State 5 includes the high reset bit to restart the sequence.	12
Table 3: An example Ramp Store sequence. The example shows use of seven ramp signal states with the largest magnitude and longest duration in State 5.	13
Table 4: An example set of Ramp Store programmed values. The example shows the actual programmed values that are required to implement the example in Table 3.	13
Table 5: LPF/Integrator block has four possible configuration values for the A coefficient. Each coefficient setting has a corresponding time constant for the LPF.	15
Table 6: DAC Gain block has eight possible configuration values.	16
Table 7: Power Shaper block has seven configuration values. The shape of the transfer curve can be adjusted by four gain settings and three threshold values.	17
Table 8: ADC Gain block has eight configuration values for the ADC Gain block.	19
Table 9: DAC verification values with the corresponding test case to actual voltage level.	26
Table 10: ADC verification values with the corresponding test case to actual output reading. ..	28
Table 11: DAC Gain verification setup values.	36
Table 12: DAC Gain verification values with the corresponding gain value to expected slope value.	36
Table 13: Power Shaper gain verification setup values.	37
Table 14: Power Shaper gain verification values with the corresponding gain value to expected slope values.	38
Table 15: ADC Gain verification setup values.	38
Table 16: ADC Gain verification values with the corresponding gain value to expected slope values.	39
Table 17: Integrator with Limiter verification setup values.	39
Table 18: Integrator with Limiter verification values with the corresponding test case to expected voltage level values.	40
Table 19: Typical configuration values for performance testing. The values for each block are the typical values for regular operation and assumed to provide stable operation. The latched value in the Ramp Profile will last for 1184 tics, which coincides to the first falling edge of the Ramp Enable signal.	41
Table 20: GSM and DCS Power Levels with corresponding output power and level. The normal tolerance is the limit for normal voltage and temperature operating levels. The extreme tolerance is the limit for extreme voltage and temperature operating levels [10].	42
Table 21: Modulation spectrum power limits with the corresponding frequency range [12].	46
Table 22: Switching transients spectrum power limits with corresponding frequency range [13].	48

Abbreviations

ADC	Analog to Digital Converter
ARFCN	Absolute Radio Frequency Channel Number
CODEC	Coder - Decoder
DAC	Digital to Analog Converter
DCS	Digital Communications System
DSP	Digital Signal Processor
DUV	Device Under Verification
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
IA	Integrated Analog
IFS	Intra-Frame Sequencer
LPF	Low Pass Filter
NIT	Magnitude of the digital value
PA	Power Amplifier
PAC	Power Amplifier Controller
PMIC	Power Management Integrated Controller
RC	Resistor Capacitor
RF	Radio Frequencies
SIM	Subscriber Identity Module
TIC	Duration length of the digital value
TVM	Transaction Verification Module

1 Introduction

Currently, the wireless communications industry is greatly expanding and its demand is equally growing. The need in the market for smaller, cheaper phones has grown to a level that forces chip manufactures to reduce part counts and create chip sets that use less power. In order to create these chip sets, better power management control and integration of discrete functionality is necessary. Converting functionality to the digital world, allows for better control over temperature and process, reduced part count, and saved physical space on the final product.

One step to meet the need for a smaller phone is to digitally implement the existing external power amplifier controller. The digital controller eliminates discrete parts that were previously used. A digital implementation also allows better control over the loop characteristics.

1.1 Conexant Systems, Inc.

Conexant Systems, Inc. provides semiconductor products and system solutions for a wide variety of communications electronics. Conexant delivers semiconductor integrated circuit products and system-level solutions for a broad range of communications applications. These products facilitate communications worldwide through wire line voice and data communications networks, cordless and cellular wireless telephony systems, personal imaging devices and equipment, and emerging cable and wireless broadband communications networks. The company has two business segments: the Personal Networking and the Internet Infrastructure business.

1.2 Air Force Chip Set

The Air Force System is a multi-slot Global System for Mobile Communications (GSM) / General Packet Radio Service (GPRS) system solution with a Direct-Conversion Transceiver. GSM is a globally accepted telecommunications standard. GPRS is an extension of GSM that allows high speed packet data to be transmitted, similar to packets transmitted on the Internet. Multi-slot enabled systems allows for the service providers to give users more speed if required by allowing a user to consume from one to eight of the possible time slots available in the GSM standard. The chip set provides all the necessary components for building a multi-band

GSM/GPRS handset including baseband, power management, integrated analog device, direct-conversion transceiver, and power amplifier. All the components descriptions are specified in [1]. Figure 1 provides a block diagram of all the components contained in the Air Force chip set solution.

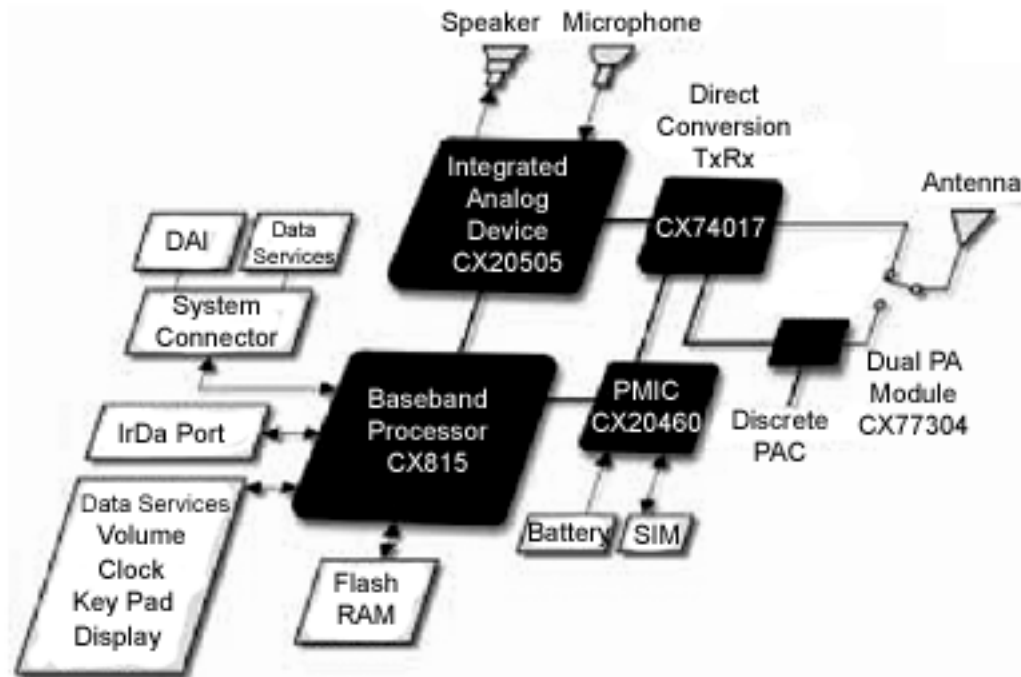


Figure 1: The GSM/GPRS system solution is comprised of the CX815 baseband processor, CX20505 integrated analog device, CX20460 power management integrated circuit, CX74017 direct-conversion transceiver, and CX77304 tri-band, multi-slot power amplifier. Connections to ports, services, battery, speaker, microphone, memory, antenna and subscriber identity module (SIM) are shown with directional arrows that indicates data flow [2].

The CX815 baseband processor contains an ARM7TDMI Thumb processor, digital signal processor (DSP), and all the interface logic required for the handset. The CX815 provides support for all the voice coding algorithms including full-rate, enhanced full-rate, and half-rate. Built in cache memory on the processor removes the requirement for external memory. All functions contained in the processor can be individually managed for power consumption.

The CX20505 is an integrated mixed-signal device that manages timing controls, interfacing between the baseband and radio frequencies. The mixed-signal device also handles all the CODEC functions.

The CX20460 power management controller (PMIC) handles all the battery charging and power supply functions. The interface to the controller consists of a serial interface that allows for simple programmability. The controller handles the voltage switching, conversion and regulation. The subscriber identity module is interfaced to phone through the controller.

The CX74017 is a Direct Conversion Transceiver that utilizes a direct down-conversion architecture that eliminates any intermediate frequency components. The receiver contains low noise amplifiers, quadrature demodulator, baseband filters, high frequency voltage controlled oscillator, and direct current offset correction sequencer. The transmitter contains a phase-frequency detector, charge pump, offset mixer, high-power voltage controller oscillators and buffer for the radio frequency output and intermediate frequency inputs. The transceiver is fully integrated with a synthesizer suitable for multi-slot GSM/GPRS.

The CX77304 Tri-Band, Multi-Slot Power Amplifier supports EGSM900, DCS1800, and PCS1900 with up to a four-slot transmission for Class 12 GPRS. The inputs and outputs are matched to 50 ohms.

1.3 Power Amplifier Controller

The power amplifier controller (PAC) is an instantaneous, closed-loop transmit power controller. The power amplifier controller controls the CX77304 Tri-Band, Multi-slot Power Amplifier for transmitting radio frequency (RF) signals. The transmitter of cellular phone is critical portion of the system. Maintaining spectral considerations while optimizing signal quality and battery life is essential to a quality phone call and overall phone performance. Proper and reliable control of the power amplifier is crucial. Without the power amplifier, all the other information processing of the phone renders useless.

The integrated PAC resides on the mixed-signal device. Figure 2 shows the generalized closed-loop PAC block diagram. The control loop in Figure 2 is a very simplified version of what is required for power amplifier control. The DSP provides the profile for the output power. The integrator sums the error signal and the system will respond by increasing or decreasing the output accordingly. The forward gain and feedback gain blocks provide gain control for adjusting system stability and response time. The power amplifier (PA) converts a control

voltage to an associated output power level. The coupler redirects a small amount of the power amplifier output to the detector. The detector converts the radio frequency energy to a corresponding voltage level to be used in the feedback path for closed loop control. There are two approaches to realizing this controller, an analog version and digital version. The current version in use is the analog controller and the current version in development, testing and in discussion in this thesis is digital version.

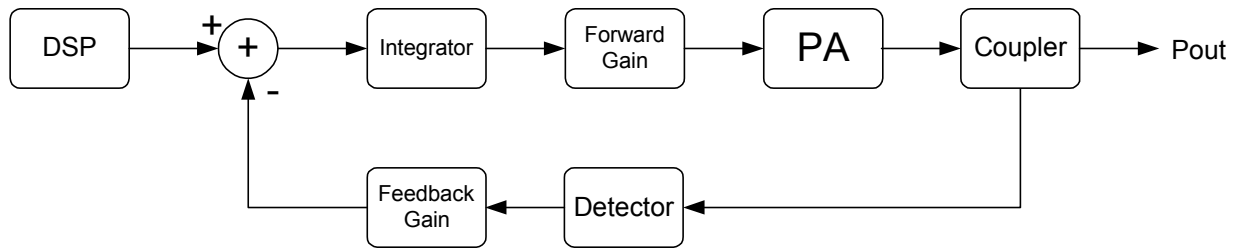


Figure 2: The power amplifier controller is a closed-loop system. The forward path consists of an integrator, gain block, power amplifier, and coupler. The feedback path consists of a detector and gain block. Pout is the amount of power outputted by the power amplifier. An example of a simplified version of the actual controller.

1.3.1 Analog Power Amplifier Controller

The current power amplifier controller in the system is an analog controller. This system has been in use since the conception of the chipset solution. The analog controller is the industry standard and has proven sufficient against the test of time. Even though the analog controller has many benefits and is fairly reliable, there is always room for improvement. Discrete parts, board space, and parameter control are desired when creating chipsets that will be manufactured by the hundreds of thousands. Currently, there is no control of the saturation characteristics of the analog integrator that can pose problems. The gain values in the forward and reverse path are set and cannot be changed during normal operation. There is no built in flexibility for better control over process and temperature. One problem that can occur is if the power amplifier saturates, the analog controller cannot compensate and the loop exhibits open loop characteristics.

1.3.2 Digital Power Amplifier Controller

Converting the analog controller to a digital controller can solve many of the analog controller issues. First and foremost board space will be reduced since the mixed signal device will contain the digital controller. Controllability has been increased ten fold since the micro controller can

modify gain values and other controller characteristics during normal operation. Temperature will become less of a factor since the signals are digitized and are less affected by extreme temperature conditions. Since the integrator will be a digital integrator, the saturation characteristics are fully controllable. All these factors will increase the power amplifier controller's reliability.

The digital power amplifier controller was achieved by incorporating all the digital blocks in the mixed signal device. The solution was developed and incorporated into the design of the mixed signal device. Full parameter controls are integrated in to the micro controller and can be controlled via a serial communication interface. The software used to control the device through the serial port can be used to initialize and control the power amplifier controller. The actual digital control of the controller allows for higher levels of testability and in the final product allows for greater product verification. For compatibility purposes the mixed signal device can be set to external mode, where the external power amplifier controller can be used. This allows for backward compatibility while the digital controller is still in the testing phase.

1.4 Road Map

The work supporting this thesis has contributed to the area of testing and verification of the digital power amplifier controller. The major contributions are testing and verifying the functionality of each block of power amplifier controller, testing and verifying the performance of the power amplifier control loop, development of a high-level GUI program that allows control over the PAC settings. In addition, an investigation and resolution of a power on transient that causes an unwanted voltage overshoot in the power amplifier.

In Section 2 of this thesis, the entire system is broken down into its basic components. The entire system is described as a whole and each subsystem is described individually along with information indicating the protocol between each block. Section 3 describes the test platform development approach used to test the PAC. Section 4 discusses the test stimulus generation and response check for block functionality and performance in the PAC. Section 5 describes in detail the development of the PAC controller program that provides efficient testability. Section 6 describes the resolution to the power on transient that causes unwanted voltage overshoot. Section 7 concludes with a summary of the work and future considerations.

2 Overview of the Testing System

The system consists of a personal computer, serial connection, micro controller, DSP, and integrated analog device. This system allows for infrastructure to control the PAC and other device through a high-level interface. Figure 3 shows the interconnections and sub-blocks of each sub-system.

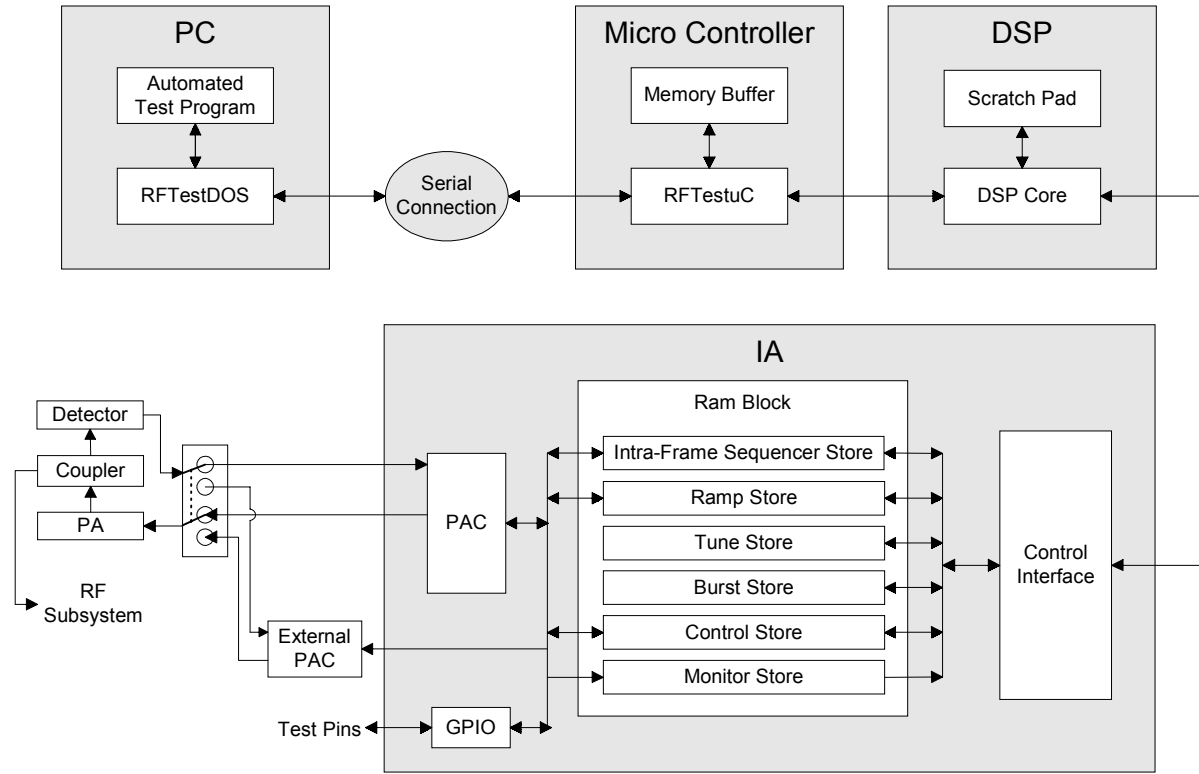


Figure 3: The system contains four major sub-systems with sub-blocks. The personal computer, micro controller, DSP, and integrated analog (IA) device are the major sub-systems. The arrows indicate data flow direction between blocks. The integrated analog device can select two power amplifier controllers. The internal PAC is discussed in this thesis.

The following sub-sections describe in detail the major sub-systems in Figure 3. The external power amplifier and general purpose input / output (GPIO) bus are also described. The internal PAC, noted as the PAC in Figure 3, is described in full and complete detail. In Section 2.7, there is a complete break down of all the sub-components of the PAC and are described in detail.

2.1 Personal Computer System

The personal computer system operates the automated test program and a DOS program called RFTestDOS. RFTestDOS is used to communicate back and forth to the device. The automated test program allows the high-level graphical user interface (GUI) to control the different settings in the PAC. This level of parameter control allows for great ability to run function and performance tests. The communication connection between the PC and the micro controller is a serial cable at 57600 bits/second (no handshaking, 8 data bits, 1 stop bit).

The device can interpret several different types of commands. The basic types of commands are read and write requests to the different devices in the system. The devices all have RAM that can be accessed using the RFTestDOS protocol. The format of the message consists of a message header and a message body. Figure 4 illustrates the message format.

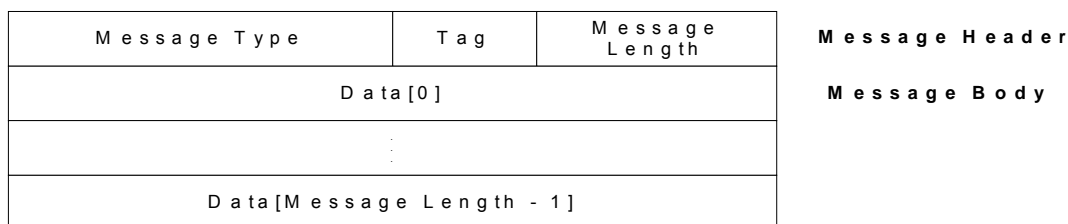


Figure 4: The RFTestDOS message format consists of message header and message body. The message header consists of the message type, tag, and message length. The message body contains the data. The number of lines in the message body is related to the message length minus one.

The message header contains the message type, tag, and message length. Each different command has a unique message type number. A message with only a header has a message length of zero. The tag is a special purpose field, which is not used. Depending on the type of message, different lengths of data are sent with a maximum of 16 data values.

2.2 Micro Controller

The micro controller runs the program called RFTestµC that interprets the command messages sent from the PC and acts upon those messages accordingly. The same message format is used if a confirmation message is sent back to the PC. The micro controller itself has its own RAM memory that it uses as a buffer to accept a set of messages and then execute them in order. The micro controller is ARM7TDMI™ THUMB processor. The ARM processor is a low power 32-

bit microprocessor with excellent code density for minimal system memory and cost [3]. The micro controller is the first interface between the high-level commands and actual PAC parameter manipulation.

2.3 Digital Signal Processor

The Digital Signal Processor (DSP) is a highly efficient Conexant designed digital signal processor. The DSP also contains a scratch pad memory buffer and the means to communicate with the micro controller. The DSP communicates to the IA via a Control Interface. The only command that is supported is the reading and writing of IA registers from the DSP. Figure 5 shows the control interface signals required to send and retrieve information.

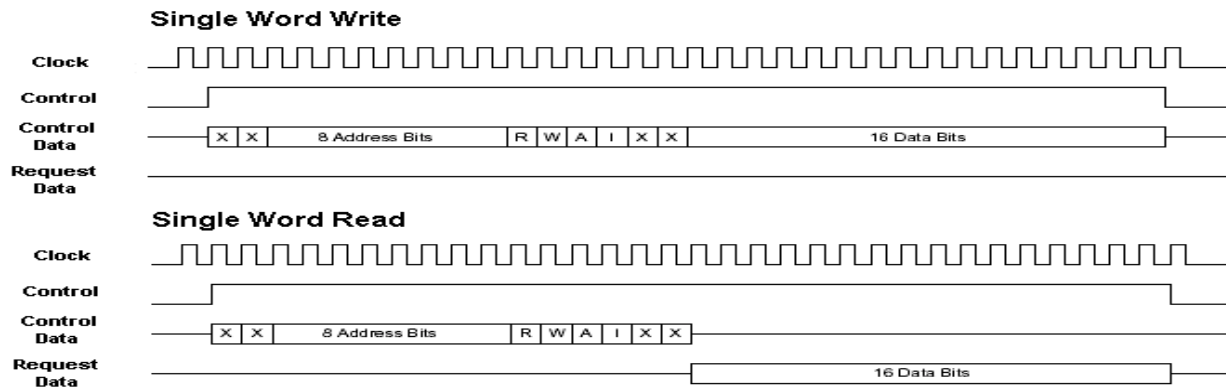


Figure 5: The DSP to integrated analog control interface consists of a clock, control, control data, and request data lines. The control data line has a specific format that is configured differently for reading and writing [4].

The control data line consists of an address field, a RWAI field, and a data field (Write Request only). The address field can access up to 256 registers in the IA. The RWAI field stands for reading, write, address read back, and indexing. If a response message is required, the address read back is set; if consecutive address locations are required to be accessed, the index field is set. The DSP is the actual parameter manipulation mechanism. When it receives instruction from the micro controller it manipulates parameter registers in the IA. Actual binary values in the registers cause the PAC to function in different ways. The registers in the IA are directly related to specific function of the PAC block, which will be described in Section 2.7. Since the DSP is the parameter manipulation mechanism for the PAC, the values can be changed during

normal operation in the digital PAC in comparison to the analog PAC that has set parameter values.

2.4 Integrated Analog

The integrated analog (IA) device is a highly integrated mixed-signal device that controls the timing and interfacing between the different subsystems in the device. The IA contains a RAM block and the DSP control interface. Serial data is received from the DSP. The data is converted into parallel data to be written to the IA registers. Parallel data is received by the DSP control interface from the IA and converted to serial for the DSP.

The RAM block in the IA can be categorized into six different groups: intra-frame sequencer, ramp, burst, tune, control, and monitor. The intra-frame sequencer store (IFS) is the heart of the IA that controls the system timing and RF control signals. The ramp store provides the control of the RF transmit signal ramping profile. The burst store contains the transmit data. The tune store provides the data used by the frequency synthesizer device in the RF subsystem. The control store provides the different configuration settings for all the IA functionality, including the PAC. The monitor store is read only and provides calibration and monitor information.

2.5 External Power Amplifier Controller

The external PAC, basically the analog PAC, takes information from the intra-frame sequencer store, ramp store, and control store. The external PAC is the current functioning controller used in the system. The power output ramp profile is fed into the external controller and the entire closed loop process uses discrete parts. The power output ramp profile is inputted by the DSP and the output power should follow this profile according to loop parameters. The IA allows for backward compatibility so the end customer can use the internal or external PAC. Only the ramp profile can be modified during normal operation, as all other parameter values in terms of resistors, capacitors, and actual performance characteristic of discrete components are set and non changeable.

2.6 General Purpose Input / Output Bus

The General Purpose Input/Output bus is a set of 10 lines that can be configured as inputs or outputs. The GPIO has the ability to route internal test signals from the IA to the test pins. The GPIO also has the ability to accept digital input signals that can be used for internal test purposes. Each pin is bi-directional and has the ability of inversion.

2.7 Internal Power Amplifier Controller

The PAC takes information from the intra-frame sequencer store, ramp store, control store, and sends back information to the monitor store. The internal PAC, basically the digital PAC, is noted in Figure 3 as the PAC and is located inside the IA block. The PAC provides an input signal to the PA that is located outside the IA. The input signal is a voltage that controls the output power level in the power amplifier. The output of the power amplifier is fed into a coupler that feeds the RF subsystem. A detector converts the power back to a DC voltage level. Figure 6 shows a detailed breakdown of the digital closed-loop PAC.

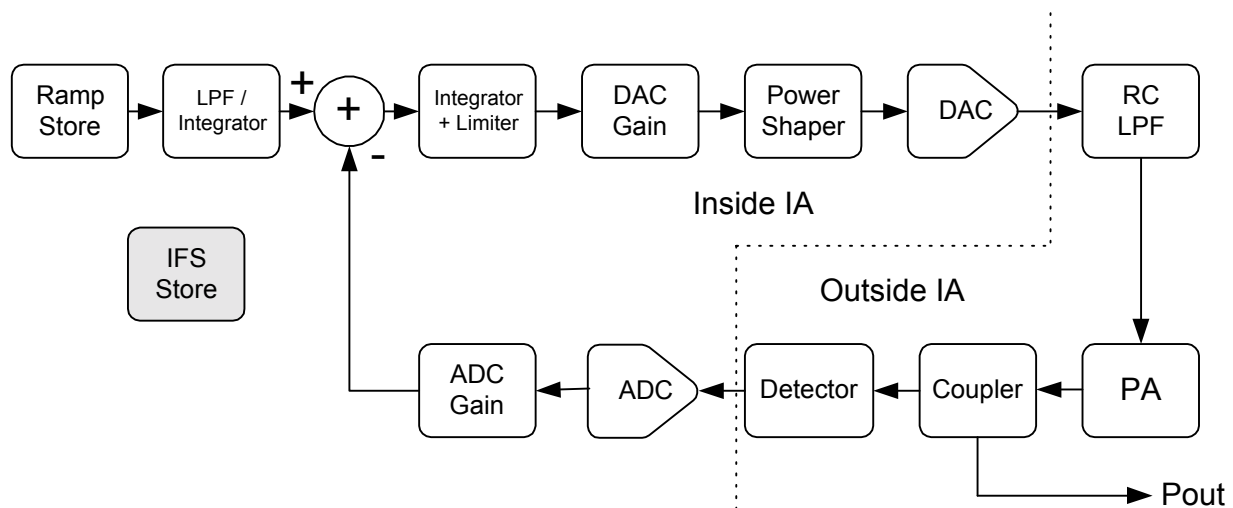


Figure 6: The digital closed-loop PAC is slightly more complex than the generalized version shown in Figure 2. The IFS is not connected to any of the blocks because it controls the base timing of the system. The IFS provides no other purpose than timing of system. The PA, coupler, and detector in this figure are the same as in Figure 3. The dotted line shows the division between the elements that are found inside and outside the IA device.

The following subsections include detailed descriptions of all the sub-blocks in Figure 6. Each sub-section describes the functionality of the block. Following up in Section 4, is a description

of how each block is tested for functionality and performance. Each block will be noted in italics.

2.7.1 Intra-Frame Sequencer

The *Intra-Frame Sequencer* (IFS) is a programmable state machine that provides the timing and control signals for the RF subsystem and internal IA functions. The *IFS* is a RAM block of 64 x 16-bit words [5]. Each state is controlled by two words, which allows for a total of 32 states. One word controls the duration of the state and the other word controls the logic level of each control signal in the state. The “duration word” for each state dictates the length of the state. The clock that drives the IFS is 2.166 MHz (0.4625μs). The actual duration of a state is calculated as:

$$(DurationWord[14..0] + 1) \times 0.4625\mu s . \quad (1)$$

Bit 15 of the duration word is used as a reset bit. If bit 15 is set high, the state machine resets the next state to the initial state after the current state duration has expired. The “assertions word” controls the logic level of 16 control signals. Some control signals are used internally and some are routed out of the IA. The PAC subsystem uses three of the control signals provided by the *IFS*. One control signal is dedicated to power on the PAC and power amplifier. The other control signal invokes the ramp sequence from the ramp store. Table 1 demonstrates an example of a control sequence that may be used. For the purposes of the example, generic names are used.

Table 1: An example IFS control sequence. The example shows the control of four of a total 16 possible signals and a total of five of the available 32 states. The duration of State 1 is the longest and the duration of State 4 is the shortest.

Control Signal	State 1	State 2	State 3	State 4	State 5
Control 1	1	1	1	1	1
Control 2	1	1	1	0	0
Control 3	1	0	1	1	1
Control 4	0	0	0	0	1
Duration	25	10	15	5	20

Table 2 shows the values that are written to the corresponding registers to create this type of control sequence.

Table 2: An example set of IFS control store programmed values. The example shows the actual programmable values that are required to implement the example sequence in Table 1. Note the duration of State 5 includes the high reset bit to restart the sequence.

Address	Comment	Hex	Binary			
0x00	State 1 controls	0007	0000	0000	0000	0111
0x01	State 1 duration	0018	0000	0000	0001	1000
0x02	State 2 controls	0003	0000	0000	0000	0011
0x03	State 2 duration	0009	0000	0000	0000	1001
0x04	State 3 controls	0007	0000	0000	0000	0111
0x05	State 3 duration	000F	0000	0000	0000	1111
0x06	State 4 controls	0005	0000	0000	0000	0101
0x07	State 4 duration	0004	0000	0000	0000	0100
0x08	State 5 controls	000D	0000	0000	0000	1101
0x09	State 5 duration (Reset)	8014	1000	0000	0001	0100

Figure 7 shows the timing diagram of the corresponding output of the control signals. Note that at the end of State 5, the entire sequence is repeated.

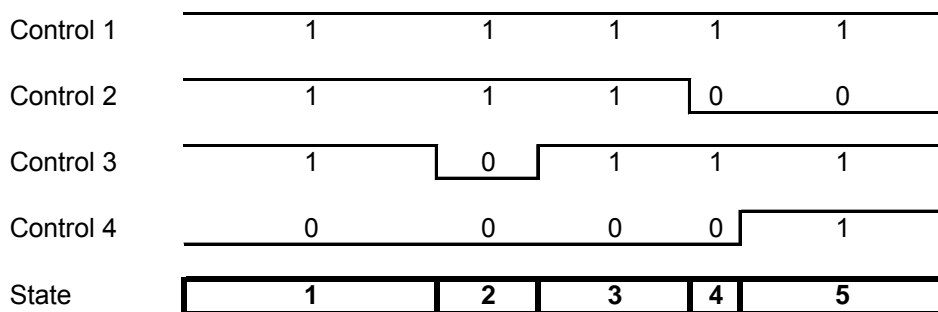


Figure 7: An example IFS control timing output. The actual output of the control signals that can be used to drive internal or external circuitry. An output of zero corresponds to a zero voltage level and an output of one corresponds to a positive rail voltage.

The *IFS* provides excellent control for a system because it provides flexibility in number and duration of states.

2.7.2 Ramp Store

The *Ramp Store* contains the ramping profile for the transmit slot. This is the fundamental input to the control loop. Changing this profile, along with loop parameters, can help to optimize battery life and performance. The *Ramp Store* is a RAM block of 32 x 16-bit words [6]. Each portion of the ramping profile is controlled by one word that allows for a total of 32 different output levels. The word is broken down into a duration and magnitude of the ramp signal. Bits

bits 15 through 6 specify the voltage level magnitude and bits 5 through 0 specify the duration. The clock that drives the ramp store is 2.166 MHz (0.4625µs). The actual duration of each portion of the ramp profile is calculated as

$$(RampStoreWord[5..0] + 1) \times 0.4625\mu s. \quad (2)$$

The *Ramp Store* can feed the ramp sequence to the internal or the external PAC. Retaining the ability to generate the ramp profile for the external PAC makes the IA device backward compatible. Table 3 demonstrates an example of a possible ramp sequence.

Table 3: An example Ramp Store sequence. The example shows use of seven ramp signal states with the largest magnitude and longest duration in State 5.

State	Magnitude	Duration
1	5	5
2	10	5
3	15	10
4	20	30
5	15	10
6	10	5
7	5	5

Table 4 shows the values that are written to the corresponding registers to create this particular ramp store profile sequence. Note the least possible duration length is one equivalent to binary zero, for example that a duration value of five is equivalent to a binary representation of four.

Table 4: An example set of Ramp Store programmed values. The example shows the actual programmed values that are required to implement the example in Table 3.

Address	Comment	Hex	Magnitude Binary [15 ... 6]	Duration Binary [5 ... 0]
0x40	State 1 Magnitude and Duration	0144	00 0000 0101	00 0100
0x41	State 2 Magnitude and Duration	0284	00 0000 1010	00 0100
0x42	State 3 Magnitude and Duration	03C9	00 0000 1111	00 1001
0x43	State 4 Magnitude and Duration	051D	00 0001 0100	01 1101
0x44	State 5 Magnitude and Duration	03C9	00 0000 1111	00 1001
0x45	State 6 Magnitude and Duration	0284	00 0000 1010	00 0100
0x46	State 7 Magnitude and Duration	0144	00 0000 0101	00 0100

Figure 8 shows the timing diagram of the resulting programmed sequence. Since the *Ramp Store* can only provide a sharp edged square signal output, filtering is required to shape the signal. The

Low Pass Filter (LPF)/Integrator block described in Section 2.7.3 provides the required shaping for the *Ramp Store* signal.

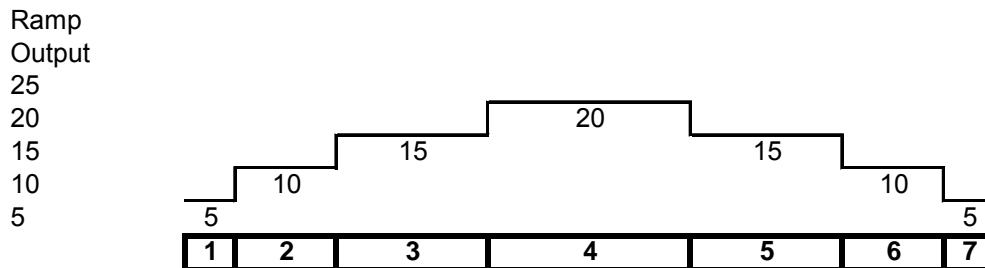


Figure 8: An example Ramp Store timing output. The actual output of the Ramp Store used to drive the LPF/Integrator portion of the control loop.

2.7.3 LPF / Integrator

The *LPF/Integrator* block provides additional shaping to the crude *Ramp Store* voltage output. On the rising edge of the transmit pulse, the block acts as a low pass filter and smoothes the edge of the voltage signal. On the falling edge of the transmit pulse, the block acts as an integrator. The block subtracts the current sample of the voltage signal from the previous sample. The integrator allows for much better control of the falling edge with fewer states by creating a piecewise linear falling edge. The transfer function of the low pass filter, where A is the block coefficient, is

$$H(z) = \frac{Az^{-1}}{1 - (1 - A)z^{-1}} \quad (3)$$

The transfer function of the integrator, where A is the block coefficient, is

$$H(z) = -\frac{Az^{-1}}{1 - z^{-1}} \quad (4)$$

The block can be configured with four different settings. Table 5 shows the values that can be selected.

Table 5: LPF/Integrator block has four possible configuration values for the A coefficient. Each coefficient setting has a corresponding time constant for the LPF.

Configuration	Time Constant	Coefficient A
1	2.67	0.0781
2	3.07	0.0625
3	3.69	0.0547
4	4.31	0.0469

2.7.4 Integrator with Limiter

The *Integrator with Limiter* block provides the cumulative addition of the loop error. The addition of the limiter can help to combat PA saturation issues by allowing for more control over block functionality. The filtered output of the *Ramp Store* and feedback path are subtracted and fed into the input of this block. The integrator contains a limiter that controls the maximum output. The behavior of the *Integrator with Limiter block* is described by

$$\begin{aligned}
 H(z) &= \frac{1}{1-z^{-1}} \\
 \text{If Output} < 0, \quad H(z) &= 0 \\
 \text{If Output} > \text{Limiter}, \quad H(z) &= \text{Limiter} .
 \end{aligned} \tag{5}$$

The only configuration value for this block is the value of the limiter, which is a 14-bit value.

2.7.5 Digital to Analog Converter Gain

The *Digital to Analog Converter (DAC) Gain* provides control for forward path gain and can be adjusted to maintain stability in the closed loop. The transfer function of the *DAC Gain* block, where A is the gain value, is

$$H(z) = A. \tag{6}$$

The block can be configured with eight different settings. Table 6 shows the values that can be selected.

Table 6: DAC Gain block has eight possible configuration values.

Configuration	Gain Value (A)
1	1/64
2	3/128
3	1/32
4	3/64
5	1/16
6	3/32
7	1/8
8	3/16

2.7.6 Power Shaper

The power shaper adds additional gain for compensation of the slow rising edge near the power amplifier's saturation point. The power shaper re-shapes the integrated signal to compensate for the non-linearity of the control gain curve of the power amplifier. This digital control block adds a great amount of controllability over loop characteristics. Variable gain over a range helps to add a greater level of control to compensate for the PA non-linearity. The power shaper contains three programmable thresholds that allow a change in gain at the thresholds. The behavior of the power shaper, where x is the input, g is the gain value and t is the threshold, is

$$\begin{aligned}
 Y(x) = & (x-t_3) \cdot g + \frac{(t_3-t_2) \cdot g}{2} + \frac{(t_2-t_1) \cdot g}{4} + t_1 & \text{when } x > t_3 \\
 & \frac{(x-t_2) \cdot g}{2} + \frac{(t_2-t_1) \cdot g}{4} + t_1 & \text{when } x > t_2 \\
 & \frac{(x-t_1) \cdot g}{4} + t_1 & \text{when } x > t_1 \\
 & \text{else } x &
 \end{aligned} \tag{7}$$

Figure 9 shows a graph of input to the power shaper versus the output of the power shaper.

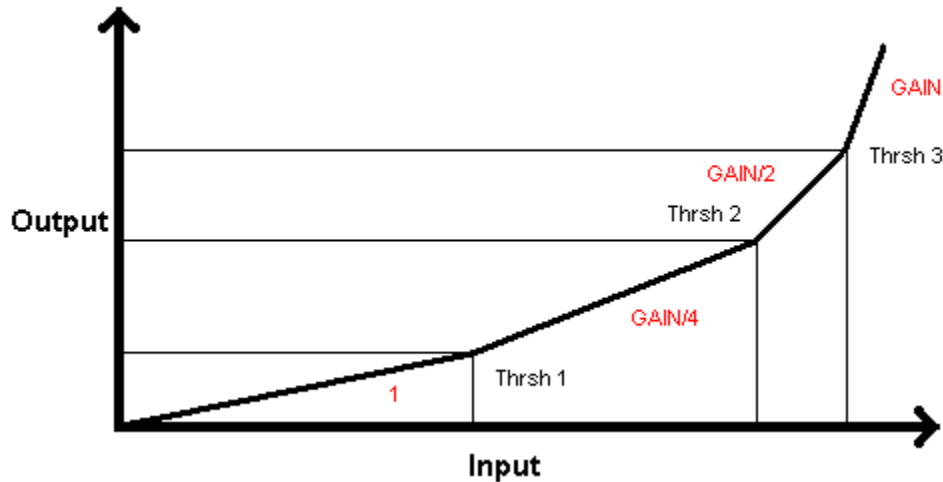


Figure 9: The Power Shaper transfer function is piecewise as shown. The forward path signal is shaped according to the transfer function to allow for high gains at higher input levels and no additional gain at lower input levels.

The block can be configured with four different settings for the gain, and 32 different settings for each threshold. Table 7 shows the values that can be selected.

Table 7: Power Shaper block has seven configuration values. The shape of the transfer curve can be adjusted by four gain settings and three threshold values.

Settings	Configuration	Gain Value (g)	Threshold Value
Gain	1		-
Gain	2	12	-
Gain	3	16	-
Gain	4	20	-
Threshold 1	1 – 32	-	0 – 992
Threshold 2	1 – 32	-	0 – 992
Threshold 3	1 – 32	-	0 – 992

2.7.7 Digital to Analog Converter

The *DAC* is a 10-bit Digital to Analog Converter that converts the samples it receives to a single-ended analog signal. The *DAC* outputs samples at 4.875 MHz. The maximum *DAC* output is 2.4V and the minimum is 0.3V, giving a range of 2.1V. Each bit value is the equivalent of approximately 2 mV at the output of the *DAC*.

2.7.8 RC Low Pass Filter

The *Resistor Capacitor (RC) LPF* is present to reduce *DAC* clock energy feed through. The block is outside of the IA. The transfer function of the *RC LPF*, where R is the resistor value and C is the capacitor value of a basic filter, is

$$H(s) = \frac{1}{1 + RCs} . \quad (8)$$

The value of the resistor is 20 ohms and capacitor is 100 pF, combining to form a 2 ns time constant. Since the components are discrete their values cannot be changed during runtime. However, these parameters can be optimized and should not need to be changed during normal operation.

2.7.9 Power Amplifier

The power amplifier accepts a control voltage that adjusts output power accordingly. The block is outside of the IA. Figure 10 shows the power amplifier characteristic measured from one sample in the laboratory [7].

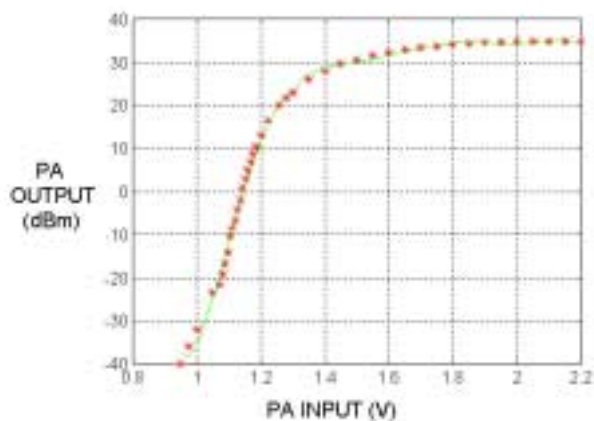


Figure 10: The Power Amplifier transfer function is described by the figure. The input to the PA is a control voltage ranging from approximately 0.9 volts to 2.2 volts. The PA saturates at the voltage of 2.2 volts. The output of the PA is power measured in the figure by dBm.

2.7.10 Coupler and Detector

The *Coupler* is assumed to be linear device. The *Coupler* has -19 ± 1 dB coupling for GSM and -14 ± 1 dB for the Digital Communications System (DCS) band. The detector converts the RF power to a corresponding voltage level.

2.7.11 Analog-To-Digital Converter

The *Analog to Digital Converter* (ADC) is an 8-bit Analog to Digital Converter that converts the input analog signal to digital samples. The ADC samples at 2 Mbps and generates a one-bit data stream. The maximum input to the ADC is 1.9V and the minimum is 0V, with a total range of 1.9V. Each bit value is equivalent to 7.4 mV on the output of the ADC.

2.7.12 ADC Gain

The *ADC Gain* provides control for the feedback path gain. The transfer function of the *ADC Gain* block, where A is the gain value, is

$$H(z) = A. \quad (9)$$

The block can be configured with eight different settings. Table 8 shows the values that can be selected.

Table 8: ADC Gain block has eight configuration values for the ADC Gain block.

Configuration	Gain Value (A)
1	3/4
2	7/8
3	1
4	1 1/8
5	1 1/4
6	1 1/2
7	1 3/4
8	2

3 Test Platform Development

In this section we describe the test method ideologies used to create the PAC Test Platform. In each sub-section the test method is broken down and described in detail. The benefits associated with this approach are discussed. The method is applied and summarized for the PAC Test Platform developed. The test platform is laid out for the purposes of describing the systematic methodology of producing a realizable and effective testing mechanism. The digital PAC has many blocks with a variety of parameter configurations and only a structured test procedure will allow for effective verification.

3.1 Modular Verification Environment

The motivation for creating a test platform for the digital PAC is to provide reproducible test results with respect to functionality and performance. The methodological approach to for this platform encompasses the ability to create verification configurations from reusable verification models. The key is to reduce the effort required for testing, increase test reproduction consistency, and the increase the ability to test more functionality in less time. The use of a reusable model will simplify the effort required to create new test cases.

To address these fundamental issues and achieve verification reuse, a modular environment was created. Figure 11 shows the modular verification environment.

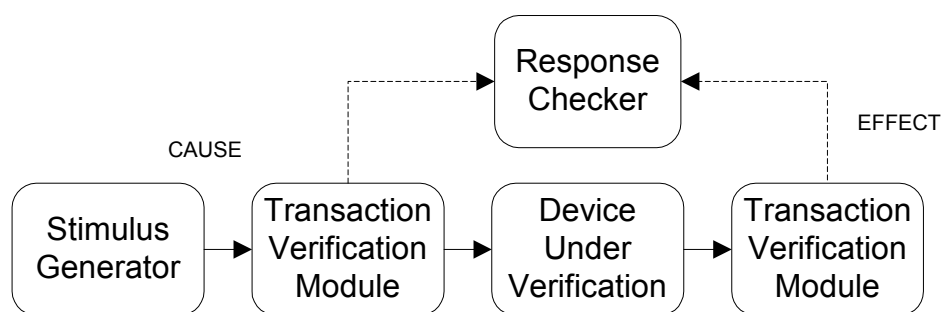


Figure 11: The Modular Verification Environment consists of the creation of a cause and observing the effect.

The Stimulus Generator block represents the high-level approach of providing configuration information for a test case. The stimulus generator provides the stimulus, which is different for each case, for the actual test. The Transaction Verification Module (TVM) block takes the high-

level commands and converts them to formatted information for the Device Under Verification (DUV). The TVM module is also responsible for converting information from the DUV to usable information back for the Response Checker. The TVM provides the interface protocol to communicate with other blocks. The DUV receives configuration information from the TVM and provides feedback. The Response Checker interprets the transaction sequences and computes expected results from these values by knowing the expected behavior of the DUV. The expected results may be compared with the actual results if feedback is available from the DUV. The Result Checker must be able to compute expected responses, store actual responses, compare responses, and report errors [8].

3.1.1 Stimulus generator

Test cases to test specific blocks or set of blocks need to be designed so that there can be a high level of confidence that the system is fully operational. It is virtually impossible to simulate all possibilities and scenarios when dealing with a complex system. Neither the time, nor the resources are available recreate every possible system configuration. Therefore, a selection of well-rounded test cases must be formed. Adequate test cases created for block testing encompasses input values ranging from typical, default, maximum, and minimum. Other factors must be considered when creating test cases for the system, which include temperature and voltage supply levels that must be varied to verify the system works at the design limits. For each test, pass or fail criteria must be established so that a test's success can be measured.

3.1.2 Transaction Verification Module

The transaction verification module collects test data. Information needs to be collected in order to verify the level of success. Information collection can consist of reading a monitor register or making actual measurements with an oscilloscope. The test case should specify the data that is required for verification for a particular case.

3.1.3 Response Checker

The response checker takes collected data, performs calculations, and outputs a "pass" or "fail" result. The amount of operator involvement when using the response checker should be minimum and can be achieved by using a spreadsheet program that can perform multiple

calculations on data and create graphs for visualization. The test case should specify the required calculations and expected results for verification for a particular case. The test case should also specify what information should be reported when a failure occurs.

3.1.4 Modularity Benefits

The methodology of modular verification reuse provides many benefits. The benefits of using a stimulus generator are it simplifies the creation and reuse of complex configurations, increases the ease of implementing because only information on creating the test cases are needed, and eases the production of random stimuli. Designs with similar interfaces can also take advantage of the stimulus generator. The benefits of using a response checker are it enables faster error detection, analysis, and debugging. The response checker helps to determine design and interface errors quickly.

Based on the benefits accompanying this abstract methodology for implementing a test platform, this method was chosen. The PAC Test Platform, which is described in detail in Section 3.2, was developed based on the same principles outlined in Section 3.1.

3.2 PAC Test Platform

Figure 12 shows the PAC test platform that was created using the modular verification environment system.

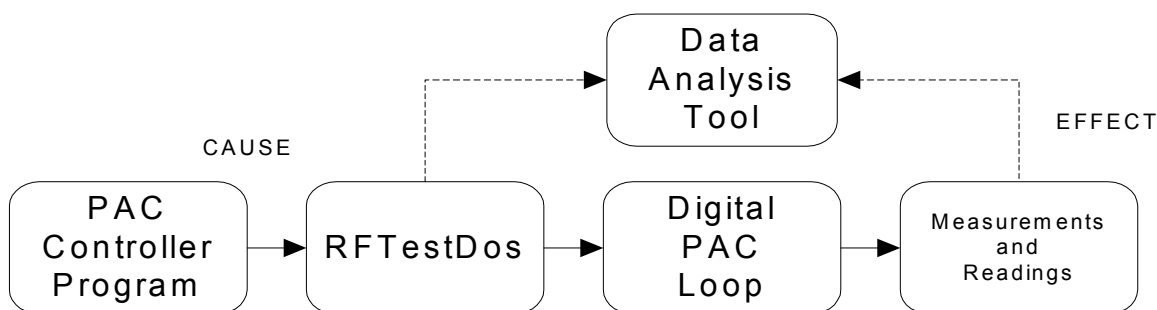


Figure 12: The PAC Test Platform based on the methodology described in Section 3.1.

The PAC Controller Program, discussed in Section 5, is the high-level stimulus generator. The Digital PAC Loop is the device under verification. RFTestDOS and Measurements and Readings are the Transaction Verification Module blocks. RFTestDOS is the backend engine of

the PAC Controller. The technician manually performs the measurements and readings. The Data Analysis Tool is the Response Checker Block that consists of Microsoft Excel spreadsheets [9].

4 Testing and Verification

The following sections address the actual stimulus generation and response checker for each functional block in the PAC loop, shown in Figure 6. Each sub-section describes in detail the actual stimulus used for the block verification and the response check method. Additionally, the sections address the performance testing procedures and stability measurements. This section makes references to window tools in the PAC Controller Program, which will be described in detail in Section 5.

4.1 Functional Specifications

4.1.1 Intra-Frame Sequencer

The device under verification has a special ability to redirect certain control signals out to the General Purpose Input/Output pins, as shown in Figure 3. The GPIO can be configured to relay most of the control signals from the *IFS* out to pins on the board. The ability to view the control signals makes it possible to verify the functionality of the *IFS*. The PAC uses three of the control signals in the *IFS*: Tx Enable, PAC Enable, and Ramp Enable.

Stimulus

Figure 13 shows the how the *IFS* should be programmed for verification purposes. The value of one period of the 2.166 MHz clock is referred to as a “tic” (462 ns).

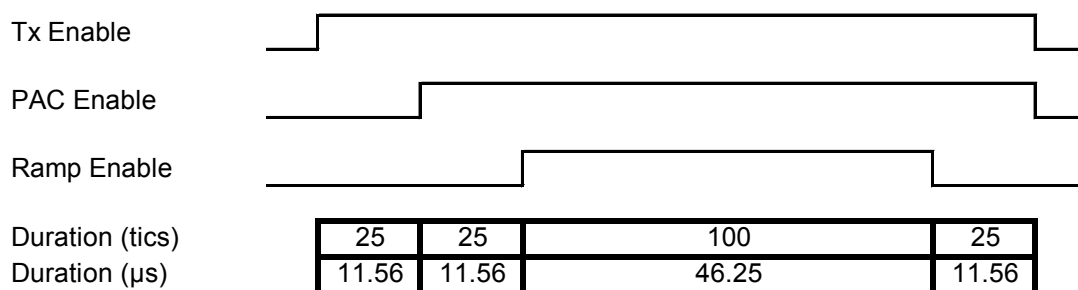


Figure 13: IFS stimulus sequence containing the three PAC control signals Tx Enable, PAC Enable, and Ramp Enable.

The *IFS* sequence can be easily programmed using the *IFS* window tool in the test program. The values of the duration and logic level for each state are entered as described in Section 2.7.1. The GPIO must be configured to route these signals out the appropriate test pins on the board, illustrated in Figure 3.

Response Checker

Each control signal was measured for correct functionality on an oscilloscope. The output voltage values were measured and verified that they were in the voltage range required for a high logic level value. The duration of each pulse was measured and verified that they were correct with respect to the programmed values.

4.1.2 Digital-To-Analog Converter

The 10-bit *DAC* has a voltage range of 2.1V, with a 0.3V minimum and 2.4V maximum, with each bit corresponding to ~2 mV. The *DAC* output is only available during the time period when the PAC Enable signal is high. The PAC Enable signal supplies power to the *DAC*, which brings it to the pedestal value of 0.3V. The *DAC* is powered off when the PAC Enable goes low.

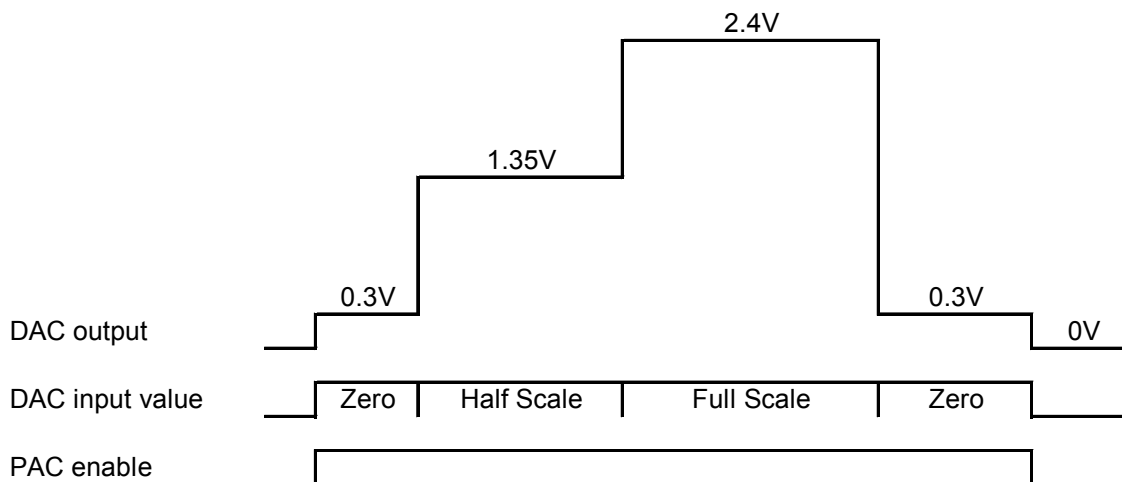


Figure 14: DAC output with PAC enable signal. The DAC output is zero volts when the PAC enable signal is low. The DAC output is 0.3 volts when the PAC enable signal is high and the digital input is zero. The DAC output can vary from 0.3 volts to 2.4 volts depending on the digital input value.

Stimulus

The stimulus that feeds the *DAC* is the 10-line GPIO, configured as inputs. A test bit controls the routing of the 10 lines to the input of the *DAC*. Since the PAC Enable signal is high for a 125 tics, the *DAC* output was observed for the entire 125 tics. Ten different constant profiles were simulated using an external power supply. Each test case consisted of setting one of the GPIO lines high. These test cases provide a wide range of voltages for the *DAC* to interpret.

Response Checker

The output of the *DAC* was measured for correct functionality via an oscilloscope. The control signals from the IFS test case were also viewed to verify that the *DAC* powers on at the correct time. The *DAC* power-on should be coincident with the PAC Enable signal as illustrated in Figure 14. The voltage of the *DAC* should read the appropriate value when the PAC Enable signal goes high. The *DAC* should power-off and clamp off to 0V when PAC Enable goes low. Table 9 shows the corresponding correct voltage values that should be available at the output of the *DAC* for each test case. The magnitude of the digital value input to the *DAC* is referred to as a “nit”. The output voltage level is the minimum voltage plus the value of nits.

Table 9: DAC verification values with the corresponding test case to actual voltage level.

Test Case	Voltage Level	Test Case	Voltage Level
Line 1 High (1 nits)	302 mV	Line 6 High (32 nits)	365.6 mV
Line 2 High (2 nits)	304.1 mV	Line 7 High (64 nits)	431.3 mV
Line 3 High (4 nits)	308.2 mV	Line 8 High (128 nits)	562.5 mV
Line 4 High (8 nits)	316.4 mV	Line 9 High (256 nits)	825.0 mV
Line 5 High (16 nits)	332.8 mV	Line 10 High (511 nits)	1.35 V

4.1.3 Ramp Store

The Ramp Store can provide a ramp profile with 32 different levels, each with a maximum of 1023 nits and 64 tics. A value of zero nits programmed in the ramp store provides the lowest possible output on the *DAC*, which is equal to 0.3 V. The value of 1023 provides the largest possible output on the *DAC*, which is equal to 2.4 V.

Stimulus

A step-up and step-down sequence was used to test the *Ramp Store* to verify the maximum and minimum magnitudes. All the durations are kept constant to have the ability to view a visible symmetric waveform. The second stimulus tests the maximum and minimum durations. The Ramp tool window can be easily configured to provide these settings. The ramp mode must be set to external mode, which routes the *Ramp Store* directly to the *DAC*. Figure 15 illustrate the test stimuli for the *Ramp Store*.

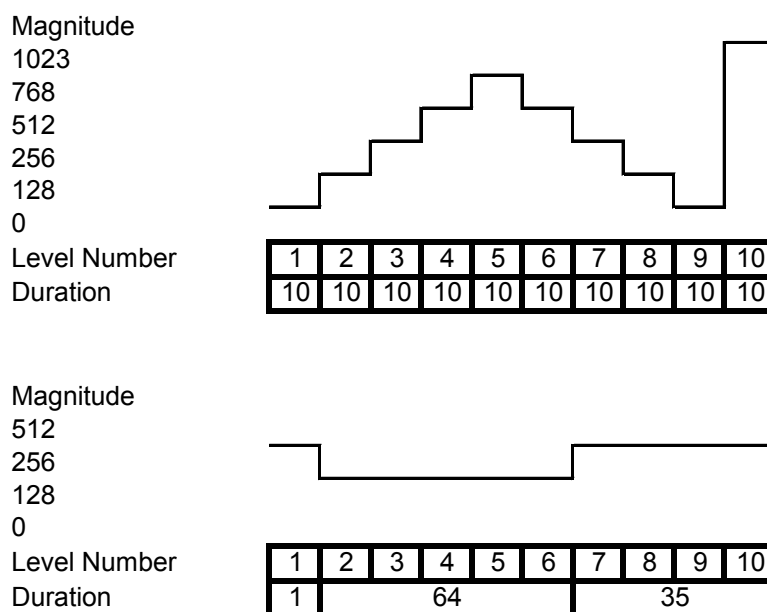


Figure 15: Ramp Stimulus sequences for the two test cases. The first sequence tests the magnitude levels and the second sequence test durations.

Response Checker

The output of the *DAC* was measured for correct functionality via an oscilloscope. Each pulse in the first sequence should be approximately 260 mV apart from each other, and each level should be equally time spaced by 10 tics (4.62 μ s). The final two levels should represent the maximum and minimum values of the *DAC*. In the second sequence, the duration of the first pulse should be one tic (0.462 μ s) and the duration of the second segment should be 64 tics (29.5 μ s).

4.1.4 Analog-To-Digital Converter

The 8-bit *ADC* has an input voltage range of 1.9 V, with a 0V minimum and 1.9 V maximum, with each bit corresponding to ~7.4 mV. The *ADC* continuously samples during PAC Enable. However, the sample value is written to a monitor register only once per cycle. The sample value is written when a control signal called Calibration Enable goes high. The single write per cycle is sufficient for testing because only DC voltage values are injected into the *ADC*. The *ADC* can work in unipolar and bipolar mode.

Stimulus

The stimulus fed to the *ADC* was a DC voltage to the input of IA, as shown in Figure 6. Both positive and negative voltages were injected into the *ADC*. With the Calibration Enable bit set high during the time PAC enable is high, the *ADC* saves the last read value into the monitor register. Since the range of the *ADC* is 1.9V, the range of the *ADC* in bipolar mode is -0.95V to +0.95V.

Response Checker

The *ADC* read tool window has the ability to read the last sample value of the *ADC*. The *ADC* read tool takes the register value and performs a conversion with respect to the mode that the *ADC* is running in. If the mode is bipolar, the value is treated as a two's complement value. If the mode is unipolar the value is left as is. With the different values injected into the *ADC*, it is a fairly straightforward procedure to determine if the *ADC* is reading the DC voltage values correctly. Table 10 shows the DC test voltages that should be used to test the *ADC* for both modes.

Table 10: ADC verification values with the corresponding test case to actual output reading.

Test Case (Unipolar)	ADC Reading	Test Case (Bipolar)	ADC Reading
0	0x00	-950 mV	0x80
500 mV	0x43	-500 mV	0xBC
1 V	0x86	0 V	0x00
1.5 V	0x65	500 mV	0x43
1.9 V	0xFF	950 mV	0x7F

4.1.5 LPF/Integrator

The *LPF/Integrator* function is to smooth the crude PA curve. On the first rising edge of the Ramp Enable signal, the block acts as a LPF. On the falling edge, the block acts as an integrator with a negative coefficient.

Stimulus

Figure 16 shows the verification of the *LPF/Integrator* block, also called the digital loop back setup.

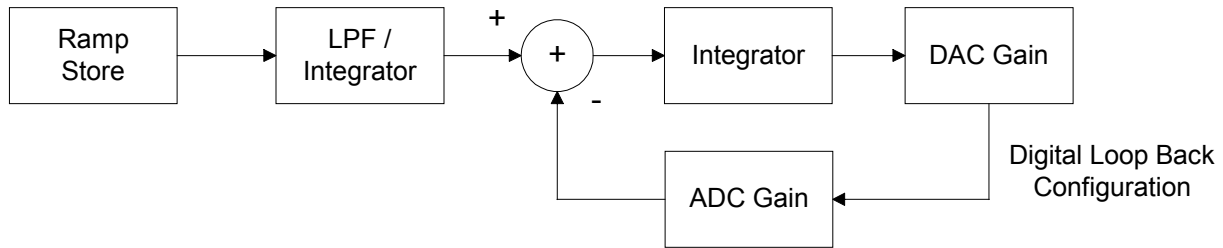


Figure 16: LPF/Integrator verification setup uses digital loop to send actual digital values to the feedback path.

The digital loop back configuration was used, where the output of the *DAC Gain* block is fed back to the *ADC Gain* block. The transfer function for the digital loop back setup when operating as a LPF, where A is the *LPF/Integrator* coefficient, K_{DAC} is the gain coefficient of the *DAC Gain* block, K_{ADC} is the gain coefficient of the *ADC Gain* block,

$$H(z) = \left(\frac{Az^{-1}}{1 - (1 - A)z^{-1}} \right) \left(\frac{K_{DAC}}{(1 + K_{DAC}K_{ADC}) - z^{-1}} \right). \quad (10)$$

The transfer function for the digital loop back setup when operating as an integrator, where A is the *LPF/Integrator* coefficient, K_{DAC} is the gain coefficient of the *DAC Gain* block, K_{ADC} is the gain coefficient of the *ADC Gain* block,

$$H(z) = \left(-\frac{Az^{-1}}{1 - z^{-1}} \right) \left(\frac{K_{DAC}}{(1 + K_{DAC}K_{ADC}) - z^{-1}} \right). \quad (11)$$

With known values for the *DAC Gain*, *ADC Gain*, and A coefficient, it is straightforward to generate the transfer curve. The values for *DAC Gain* and *ADC Gain* are 1/16 and 1,

respectively. The ramp store inputs are a step and an impulse stimulus of 250 nits. The *LPF/Integrator* block has four different coefficient configurations to verify for.

The expected responses for the LPF case shown in Figure 17 and Figure 18 were plotted using

$$y_m = x_{m-1} \left(\frac{K_{DAC} \times A}{1 + K_{DAC} K_{ADC}} \right) + y_{m-1} \left(\frac{K_{DAC} K_{ADC} + 2 - A - K_{DAC} K_{ADC} A}{1 + K_{DAC} K_{ADC}} \right) + y_{m-2} \left(\frac{A - 1}{1 + K_{DAC} K_{ADC}} \right). \quad (12)$$

The expected responses for the integrator case shown in Figure 19 and Figure 20 were plotted using

$$y_m = x_{m-1} \left(\frac{-K_{DAC} A}{1 + K_{DAC} K_{ADC}} \right) + y_{m-1} \left(\frac{2 + K_{DAC} K_{ADC}}{1 + K_{DAC} K_{ADC}} \right) + y_{m-2} \left(\frac{-1}{1 + K_{DAC} K_{ADC}} \right). \quad (13)$$

Figure 17 shows the expected impulse response for the LPF test case.

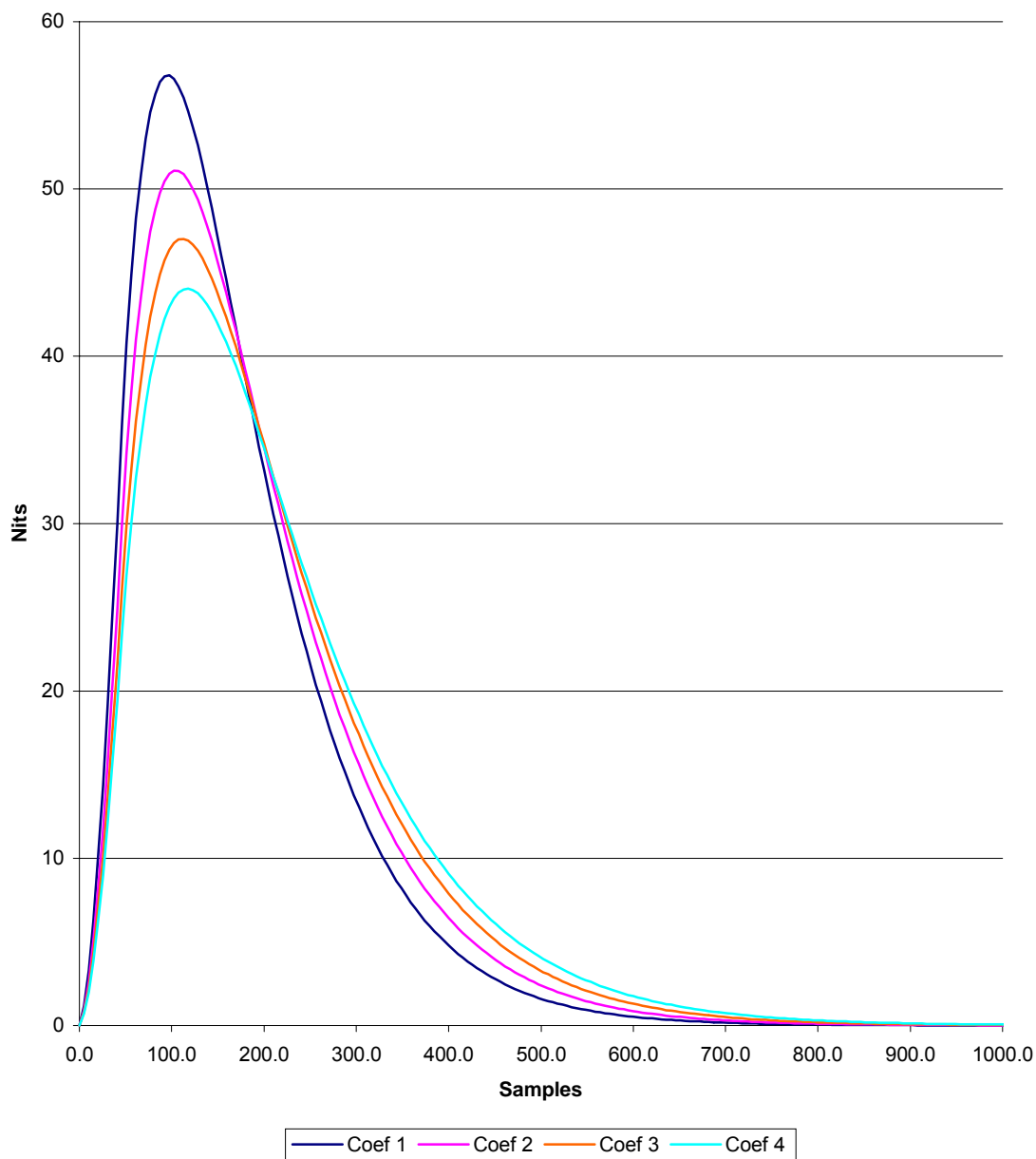


Figure 17: Impulse response for the LPF test case using a 250 nit impulse input with varying coefficient A of the LPF/Integrator block. The y-axis is the value in output value in nits and the x-axis the number of samples. Four curves for the four corresponding coefficient values are described in the legend.

Figure 18 shows the expected step response for the LPF test case.

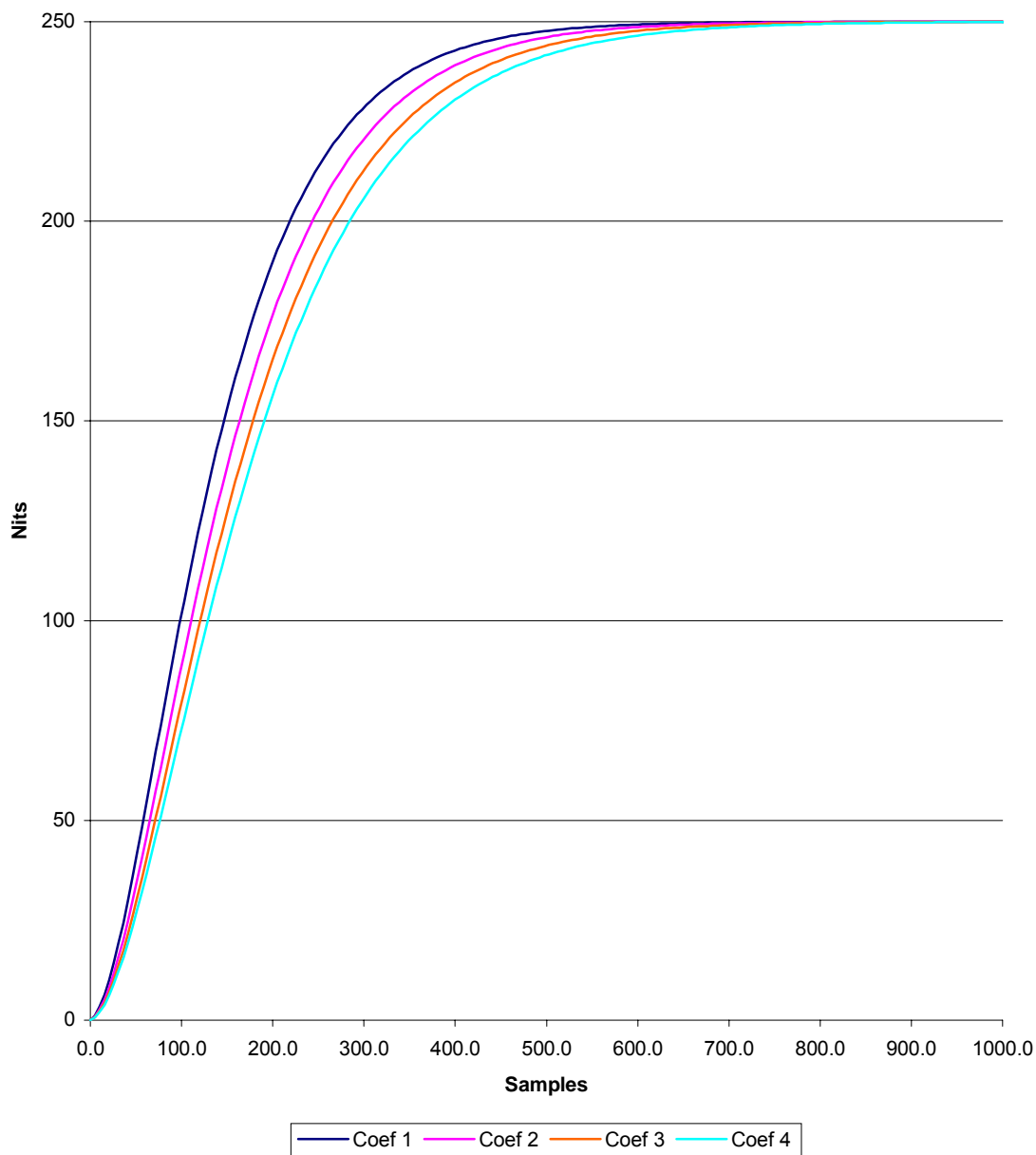


Figure 18: Step response for the LPF test case using a 250 nit step input with varying coefficient A of the LPF/Integrator block. The y-axis is the value in output value in nits and the x-axis the number of samples. Four curves for the four corresponding coefficient values are described in the legend.

Figure 19 shows the expected impulse response for the integrator test case.

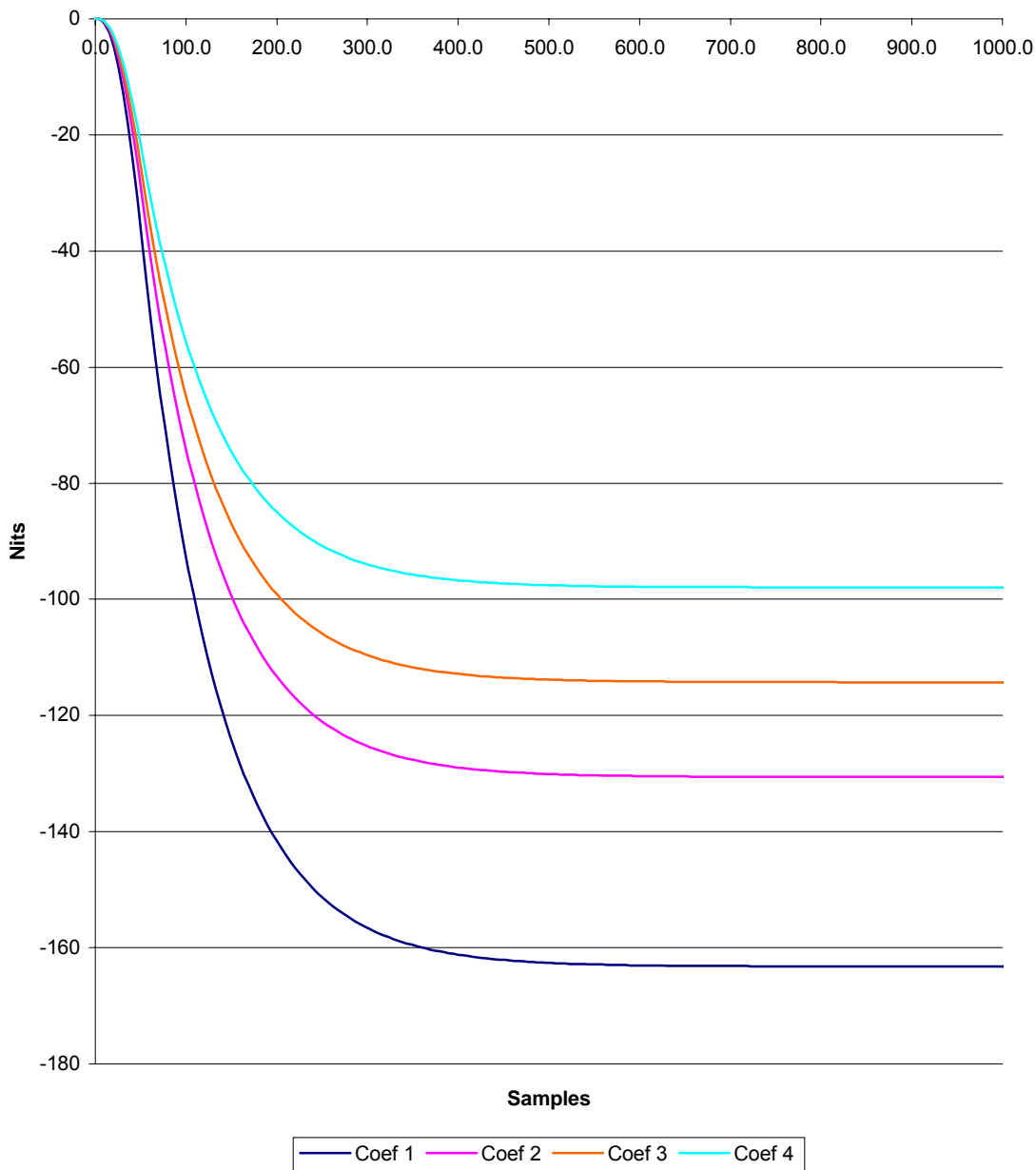


Figure 19: Impulse response for the integrator test case using a 250 nit impulse input with varying coefficient A of the LPF/Integrator block. The y-axis is the value in output value in nits and the x-axis the number of samples. Four curves for the four corresponding coefficient values are described in the legend.

Figure 20 shows the expected step response for the integrator test case.

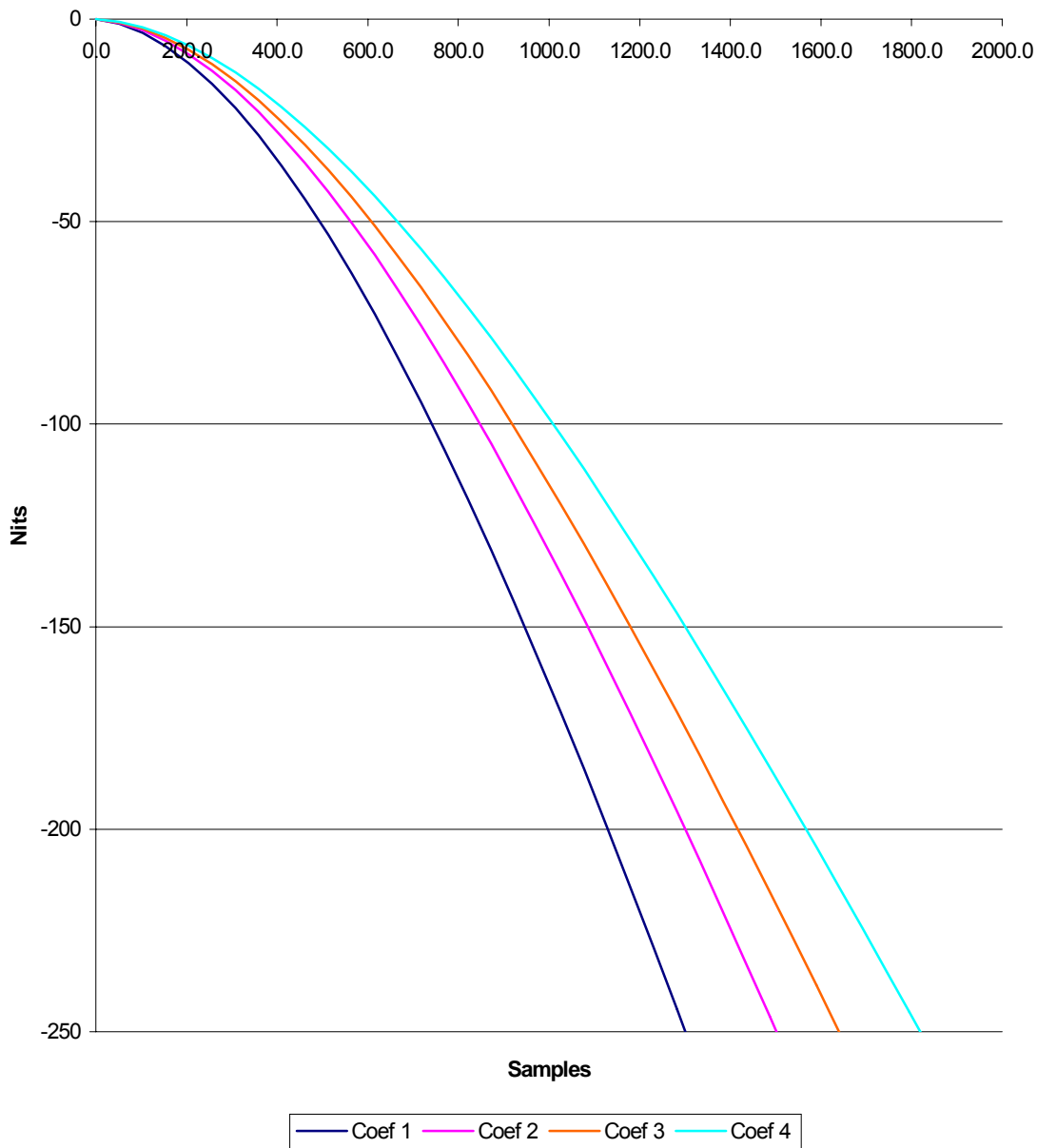


Figure 20: Step response for the integrator test case using a 250 nit step input with varying coefficient A of the LPF/Integrator block. The y-axis is the value in output value in nits and the x-axis the number of samples. Four curves for the four corresponding coefficient values are described in the legend.

Response Checker

The step and impulse responses of the digital loop back configuration are measured using the oscilloscope and plotted. The expected and actual transfer curve can be compared to verify the block is functioning properly. Each response and configuration for the LPF and integrator should be verified.

4.1.6 DAC Gain

The *DAC Gain* provides the loop with the forward path gain control. The test setup for the *DAC Gain* assumes that there is full control of the *Ramp Store* and *ADC* to provide constant signals, and that the *LPF/Integrator* block reaches its steady state value in a reasonable time.

Stimulus

Figure 21 illustrates the *DAC Gain* verification setup.

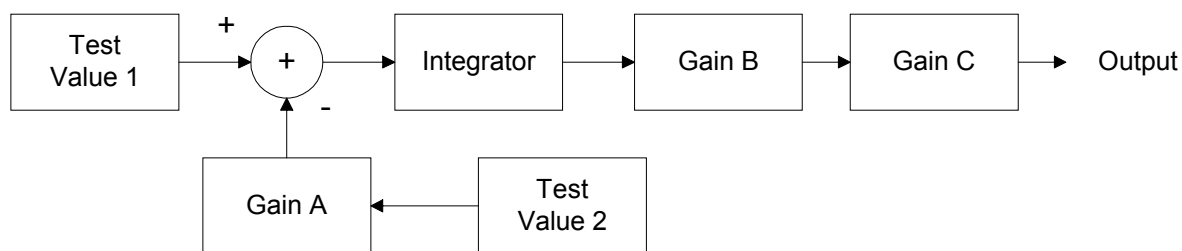


Figure 21: DAC Gain verification setup assumes the Ramp Store, ADC and LPF/Integrator block are functional.

Test Value 1 represents the *Ramp Store* value after the filter has settled, Test Value 2 represents the constant *ADC* value, Gain A represents the *ADC Gain*, Gain B represents the *DAC Gain*, and Gain C represents the *Power Shaper* gain. All the values and gains remain constant throughout the test and only the DAC gain is varied. The integrator block provides the addition of the error signal and an output for monitoring. The slope of the rising edge line is equivalent to the error signal multiplied by the forward path gain. The typical values for the static blocks are listed in Table 11.

Table 11: DAC Gain verification setup values.

Block	Value
Test Value 1	100 nits
Test Value 2	0x32, 50 nits, 371 mV ADC input
Gain A	1
Gain C	1

The *Power Shaper* gain is assumed to be one because all the thresholds are set to the maximum value. The *DAC Gain* block has eight different gain configurations.

Response Checker

The slope of the expected output rising edge is

$$Slope = (\text{Test Value 1} - \text{Gain A} \times \text{Test Value 2}) \times \text{Gain B} \times \text{Gain C} . \quad (14)$$

The linear portion of the output is measured when determining the actual resulting slope. The output should have an increasing slope at the beginning during the period the LPF is reaching a steady state value. The non-linear portion must be ignored to provide accurate results. All the digital logic is running at 4.875MHz. The following equation is used to compute the actual slope.

$$Slope = \frac{\Delta Voltage \times \frac{1024}{2.1V}}{\Delta Time \times 4.875MHz} . \quad (15)$$

Table 12 shows the expected slope values with each of the eight *DAC Gain* configurations.

Table 12: DAC Gain verification values with the corresponding gain value to expected slope value.

DAC Gain Value	Expected Slope
1/64	0.781
3/128	1.719
1/32	1.523
3/64	2.344
1/16	3.125
3/32	4.688
1/8	6.250
3/16	9.375

4.1.7 Power Shaper

The *Power Shaper* gain provides the loop with additional forward path gain adjustment. The test setup for the *Power Shaper* gain uses the same setup as the *DAC Gain* test (Section 4.1.6). Since the block's gain values are dependent on the three programmable thresholds of the block, four distinct sections of the output are visible on the output with different slopes.

Stimulus

All the values and gains remain constant throughout the test and only the *Power Shaper* gain is varied. The *Power Shaper* gain block has four different gain configurations. The typical values to choose for the static blocks are listed in Table 13.

Table 13: Power Shaper gain verification setup values.

Block	Value
Test Value 1	100 nits
Test Value 2	0x32, 50 nits, 371 mV ADC input
Gain A	1
Gain B	1/64

The *Power Shaper* can be programmed with three different threshold values. Any combination of thresholds can be used, as long as Threshold 1 is less than Threshold 2 and Threshold 2 is less than Threshold 3. The three basic test cases have all the thresholds at small values, all thresholds at large values, and all the thresholds equally spaced out.

Response Checker

The linear part of the actual slope must be measured and compared with the expected slope value. The equations in Section 4.1.6 are used to calculate the expected and actual slopes for verification. Table 14 shows the expected slope values with each of the four power shaper configurations. Note there are four expected slopes that correspond to the four section piece-wise output that should occur.

Table 14: Power Shaper gain verification values with the corresponding gain value to expected slope values.

Power Shaper Gain Value	Expected Slope
8	0.781, 1.5625, 3.125, 6.250
12	0.781, 2.344, 4.688, 9.375
16	0.781, 3.124, 6.248, 12.496
20	0.781, 3.906, 7.813, 15.625

The transitions between the different gains must be measured. The measured voltage values must be converted back to nit values and compared to the threshold values. The test verifies that the *Power Shaper* gain values changed at the appropriate values. The actual measured threshold can be compared to the expected threshold by

$$Threshold = (Voltage - 0.3V) \times \frac{1024}{2.1V}. \quad (16)$$

4.1.8 ADC Gain

The *ADC Gain* provides the loop with the feedback path gain adjustment. The test setup for the *ADC Gain* uses the same information as the *DAC Gain* test (Section 4.1.6).

Stimulus

All values and gains remain constant throughout the test and only the *ADC Gain* is varied. The *ADC Gain* block has eight different gain configurations. The typical values for the static blocks are listed in Table 15.

Table 15: ADC Gain verification setup values.

Block	Value
Test Value 1	150 nits
Test Value 2	0x32, 50 nits, 371 mV ADC input
Gain B	1/16
Gain C	1

Response Checker

The linear part of the actual slope must be measured and compared with the expected slope value. The equations in Section 4.1.6 are used to calculate the expected and actual slopes for

verification. Table 16 shows the expected slope values with each of the eight *ADC Gain* configurations.

Table 16: ADC Gain verification values with the corresponding gain value to expected slope values.

ADC Gain Value	Expected Slope
$\frac{3}{4}$	7.031
$\frac{7}{8}$	6.641
1	6.250
$1 \frac{1}{8}$	5.859
$1 \frac{1}{4}$	5.469
$1 \frac{1}{2}$	4.688
$1 \frac{3}{4}$	3.906
2	3.125

4.1.9 Integrator with Limiter

The *Integrator with Limiter* provides the summation of the error signal samples with the ability to limit its output sample value. The limiter value allows the control of the maximum output level of the integrator.

Stimulus

All the values and gains remain constant throughout the test and only the limiter value is varied. The range for the limit value is 0 to 16383 decimal. The maximum and minimum values for the limit are tested, along with the typical, low and high values. Typical values for the static blocks are listed in Table 17.

Table 17: Integrator with Limiter verification setup values.

Block	Value
Test Value 1	100 nits
Test Value 2	0x00, 0 nits, 0 mV ADC input
Gain A	G=8, Th1 = 0, Th2 = 0, Th3 = 0
Gain B	$\frac{1}{8}$
Gain C	1

The *Power Shaper* gain is assumed to be eight because when all the thresholds are set to the minimum value this causes the block gain to operate at a maximum. The total forward gain is one.

Response Checker

The output should ramp up to the limit value corresponding to the programmed value. The integrator ramps up and limits at the maximum level implies the integrator is functioning. The output level value with the given limit is calculated by

$$Output = \left(\frac{Limit}{16383} \times 2.1V \right) + 0.3V . \quad (17)$$

Table 18 shows the corresponding voltage value that should be available at the output of the DAC for each test case.

Table 18: Integrator with Limiter verification values with the corresponding test case to expected voltage level values.

Test Case	Voltage Level
Limiter at 0	300 mV
Limiter at 1	301 mV
Limiter at 2500	620.4 mV
Limiter at 5000	940.8 mV
Limiter at 7500	1.26 V

Test Case	Voltage Level
Limiter at 10000	1.58 V
Limiter at 12500	1.90 V
Limiter at 15000	2.22 V
Limiter at 16383	2.40 V
Limiter at 8192	1.35 V

4.2 Performance Specifications

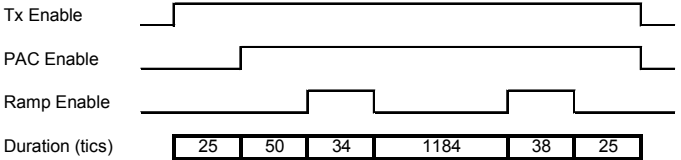
4.2.1 Output Power

The output power test verifies that the radio can maintain an acceptable output power level for a phone call. The output power is defined as the average power supplied to the antenna during the useful part of the transmit burst.

Stimulus

Table 19 shows the typical configuration values for the PAC.

Table 19: Typical configuration values for performance testing. The values for each block are the typical values for regular operation and assumed to provide stable operation. The latched value in the Ramp Profile will last for 1184 tics, which coincides to the first falling edge of the Ramp Enable signal.

Settings	Value						
LPF / Integrator	T=3.28, A=1/16						
Integrator + Limiter	12800 (decimal)						
DAC Gain	1/16						
Power Shaper	G=8, T1=400, T2=450, T3=500						
ADC Gain	1						
Ramp Profile	[364,26] [380,7] [370,1]* [72,9] [114,10] [228,9] [576,10] (*latched value)						
IFS Profile	 <p>The diagram shows the timing of three enable signals: Tx Enable, PAC Enable, and Ramp Enable. Below the signals is a duration table in tics:</p> <table border="1"> <tr> <td>25</td> <td>50</td> <td>34</td> <td>1184</td> <td>38</td> <td>25</td> </tr> </table>	25	50	34	1184	38	25
25	50	34	1184	38	25		

The output power test is performed on three channels, one in each absolute radio frequency change number (ARFCN) range and on each power level. The power also is verified under temperature extremes, voltage supply extremes and in both GSM (900 MHz) and DCS (1800 MHz) bands.

Response Checker

Table 20 describes the expected power levels with the corresponding tolerances.

Table 20: GSM and DCS Power Levels with corresponding output power and level. The normal tolerance is the limit for normal voltage and temperature operating levels. The extreme tolerance is the limit for extreme voltage and temperature operating levels [10].

GSM Power Levels				DCS Power Levels			
Power Level	Output Power (dBm)	Tolerance (dB)		Power Level	Output Power (dBm)	Tolerance (dB)	
		Normal	Extreme			Normal	Extreme
5	33	±2	±2.5	0	30	±2,5	±2,5
6	31	±3	±4	1	28	±4	±4
7	29	±3	±4	2	26	±4	±4
8	27	±3	±4	3	24	±4	±4
9	25	±3	±4	4	22	±4	±4
10	23	±3	±4	5	20	±4	±4
11	21	±3	±4	6	18	±4	±4
12	19	±3	±4	7	16	±4	±4
13	17	±3	±4	8	14	±4	±4
14	15	±3	±4	9	12	±4	±4
15	13	±3	±4	10	10	±4	±4
16	11	±5	±6	11	8	±4	±4
17	9	±5	±6	12	6	±5	±4
18	7	±5	±6	13	4	±5	±4
19	5	±5	±6	14	2	±5	±6
				15	0	±5	±6

Figure 22 shows how the test equipment is setup for the power measurements.

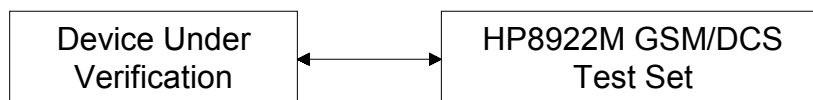


Figure 22: Output Power verification setup requires the use of HP8922M GSM/DCS Test Set. The connection from the Device to the HP8922M is a standard RF cable.

The connection between the test set and device is a radio frequency (RF) cable. The test set must be calibrated properly, including compensation for cable loss. The HP8922M provides the output power level of the current call on the output screen. The measured output power should correspond to the values in the table. The power levels on each of three channels and two bands must comply with normal tolerance limits. The output power must comply with the extreme tolerance limits during tests of the temperature and voltage supply extremes.

4.2.2 Output Burst Timing

The output burst timing test verifies that the radio transmitter does not disrupt any users that are using adjacent time slots on the same frequency. The transmit burst timing is the envelope of the transmitted RF power with respect to time. Figure 23 illustrates an example of how transmit slot timing can affect users in adjacent time slots. The GSM standard includes 8 time slots of $577\ \mu\text{s}$ of time each.

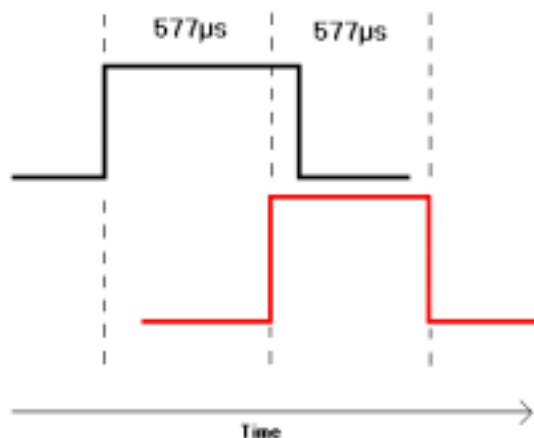


Figure 23: Output burst timing example. The timing process of the transmit signal may cause interference into other user time slots if slot timing is not accurate.

Stimulus

The typical values for the configuration of the PAC can be found in Section 4.2.1. The output burst timing test is performed on three channels, one in each ARFCN range and on each power level. The burst timing also is verified under temperature extremes, voltage supply extremes and in both GSM and DCS bands.

Response Checker

Figure 24 illustrates the time mask that the burst must follow.

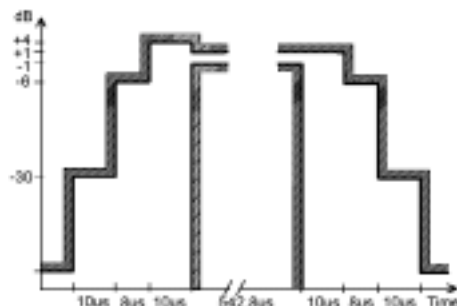


Figure 24: Transmit burst time mask that output power must fit to pass performance testing [11].

Figure 22 shows the test setup for the burst time. The HP8922 provides an on-screen time mask of the current call. The power burst profile must fit within the time mask template in all cases. The output power must also fit the time mask for the temperature and voltage supply extremes.

4.2.3 Output RF Spectrum Due To Modulation

The output RF spectrum due to modulation test verifies that the transmit bursts are within the bandwidth allowed so that the burst do not disrupt other users who may be using one of the nine adjacent channels. The output RF spectrum due to modulation is the relation between the frequency offset from the carrier and the power, produced by the device due to the effects of modulation, measured in a specific bandwidth and time. Figure 25 illustrates an example of how spectrum due to modulation can affect users in adjacent channels. The GSM standard includes 124 channels of 200 kHz of bandwidth each.

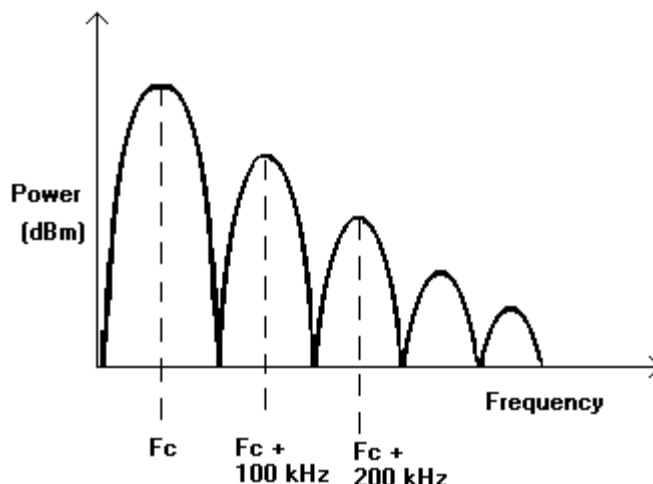


Figure 25: Output RF spectrum due to modulation example. The modulation process may cause interference into other user channels and only specific levels of power are allowed according to the GSM standard.

Stimulus

The typical values for the configuration of the PAC can be found Section 4.2.1. The output RF spectrum due to modulation test is performed on three channels, one in each ARFCN range and on the minimum and maximum power level. The spectrum also is verified under temperature extremes, voltage supply extremes and in both GSM and DCS bands.

Response Checker

Figure 26 shows how the test equipment is setup for the spectrum measurements.

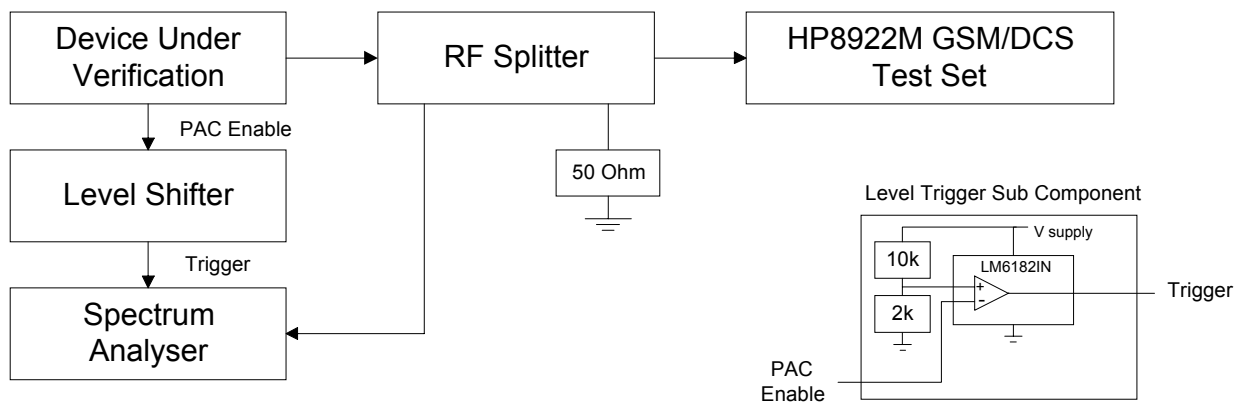


Figure 26: Spectrum measurements test setup with the device and HP8922M Test Set. The Level Shifter provides the trigger for the Spectrum Analyser. The RF Splitter provides the input signal to both the HP8922M and the Spectrum Analyser, and is also terminated with a 50 Ohm connector.

The analyzer is setup so that it is gated during the usable part of the transmit burst. The time does not include the burst ramping and midamble. The resolution bandwidth (RB) and video bandwidth (VB) are each set to 30 kHz and 50 bursts are captured. With the analyzer span set to zero, measurements of the power at 30 kHz increments on either side of the carrier are performed. The tests are performed with the following settings: RB/VB are 100 kHz with 100 kHz incremented measurements, and RB/VB are 200 kHz with 200 kHz measurements. Table 21 shows the modulation spectrum limits for the frequency offsets from the carrier.

Table 21: Modulation spectrum power limits with the corresponding frequency range [12].

Frequency Offset (kHz)	Relative Power (dBc)
± 100	+ 0.5
± 200	-30
± 250	-33
± 400	-60
± 600 to < ± 1800	-60

The spectrum must fit within the spectral mask template in all cases. The power must also fit the spectral mask for the temperature and voltage supply extremes.

4.2.4 Output RF Spectrum Due To Switching Transients

The output RF spectrum due to switching transients test verifies that the transmit bursts are within the bandwidth allowed so that the bursts do not disrupt other users who may be using one of the nine adjacent channels. The output RF spectrum due to switching transients is the relation between the frequency offset from the carrier and power produced by the device due to the effects of power ramping, measured in a specific bandwidth and time. The effect is known as spectral splatter. Figure 27 illustrates an example of how spectrum due to switching transients can affect users in adjacent channels

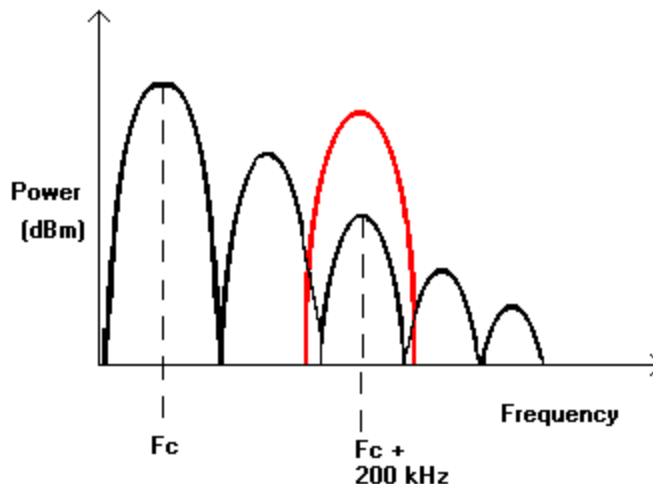


Figure 27: Output RF spectrum due to switching transients example. The switching process of the transmit signal may cause interference into other user channels and only specific levels of power are allowed according to the GSM standard.

Stimulus

The typical values for the configuration of the PAC are shown in Section 4.2.1. The output RF spectrum due to switching transients test is performed on three channels, one in each ARFCN range and on all power levels. The spectrum also is verified under temperature extremes, voltage supply extremes and in both GSM and DCS bands.

Response Checker

Figure 26 shows the test equipment setup for the spectrum measurements. The analyzer is setup so that it is gated during the usable part of the transmit burst. The resolution bandwidth (RB) and video bandwidth (VB) are set to 100 kHz, with at least 10 bursts captured. With the analyzer span set to 4MHz and the trace set to max hold, the measurements of the peak power at ± 400 kHz, ± 600 kHz, ± 1200 kHz, and ± 1800 kHz are taken. Table 22 shows the modulation spectrum limits for the frequency offsets from the carrier.

Table 22: Switching transients spectrum power limits with corresponding frequency range [13].

GSM Levels					DCS Levels				
Power (dBm)	Maximum Allowed (dBm)				Power (dBm)	Maximum Allowed (dBm)			
	±400kHz	±600kHz	±1200kHz	±1800kHz		±400kHz	±600kHz	±1200kHz	±1800kHz
33	-19	-21	-21	-24	20	-22	-24	-24	-27
31	-21	-23	-23	-26	28	-23	-25	-26	-29
29	-23	-25	-25	-28	26	-23	-26	-28	-31
27	-23	-26	-27	-30	24	-23	-26	-30	-33
25	-23	-26	-29	-32	22	-23	-26	-31	-35
23	-23	-26	-31	-34	≤20	-23	-26	-32	-36
≤21	-23	-26	-32	-36					

The spectrum must fit within the spectral mask template in all cases. The power must also fit the spectral mask for the temperature and voltage supply extremes.

4.3 Loop Stability

Loop stability is a key element in the verification of the closed-loop system. The system must operate in closed loop mode without oscillating. The two elements that specify the level of loop stability are the gain margin and phase margin. The gain margin is the difference in the gain of the actual response and unity gain (0 dB), where the phase is 180 degrees. Phase margin is the difference in the phase of response when the gain response is unity and 180 degrees. The loop bandwidth is the maximum rate that a control loop can respond to a change in a control parameter. It is measured at the frequency that the gain drops to unity.

Stimulus

Figure 28 illustrates the block diagram and the circuit used to verify the loop stability.

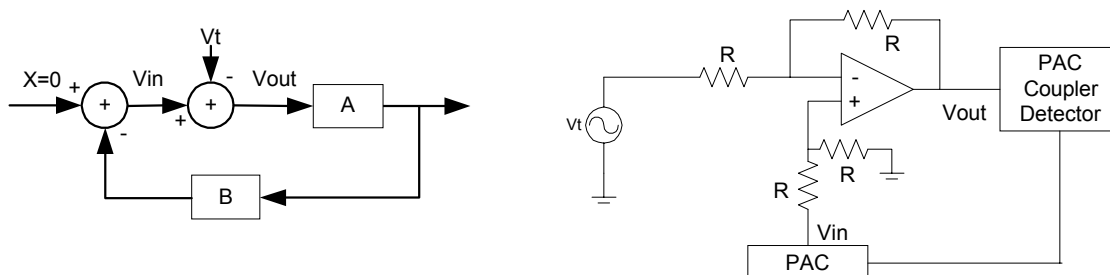


Figure 28: Loop stability verification setup. The diagram on the left illustrates the block diagram of the signals and gain blocks. The diagram on the right illustrates the actualization of the block diagram using an operational amplifier and resistors. The value of the resistors (R) is 10K Ohms. The value of V_t sweeps for 0 mV to 50 mV. V_{out} and V_{in} will be the points of measurement.

The test concept is to inject the sweep voltage V_t , into the loop and measure the effects, while the loop is operating in normal mode, as Jason Millard stated during a private meeting. The small signal should not change the loop's operating point. The summing node in Figure 28 allows the signal to be injected. The transfer function of the summing node is

$$\frac{V_{out}}{V_{in}} = -AB. \quad (18)$$

The value is equivalent to the loop gain shifted by 180 degrees.

Response Checker

The measurement of the output voltage with respect to the input voltage can be made using a low frequency network analyzer. The sweep voltage, V_t , is connected to one port of the analyzer and a high impedance probe is connected to the other analyzer input. A measurement is made with the high impedance probe at V_{out} and the network analyzer is then normalized to that data. Once normalized, a measurement is made at V_{in} that results in the actual measurement of V_{in} over V_{out} because the V_t will cancel out. The plots of the response are then interpreted to determine gain and phase margin. To cover temperature and process variation, 60 degrees of phase margin and 10 dB of gain margin are found to be adequate. The phase margin is measured from zero degrees to adjust for the inverting nature of the operational amplifier configuration.

Example analyzer output for the gain margin measurement is shown in Figure 29.

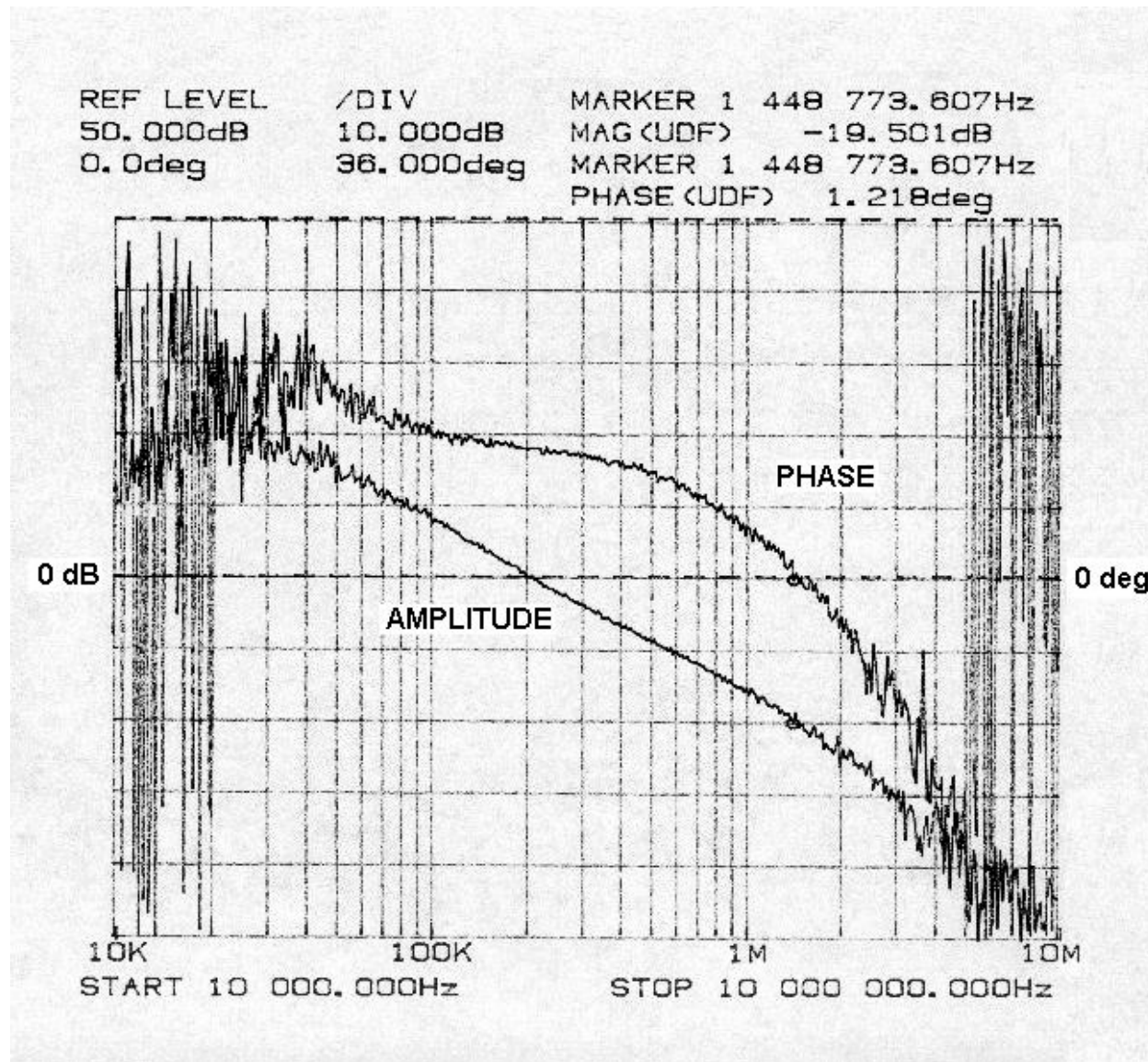


Figure 29: Gain Margin value measurement. The value of the gain margin is determined by measuring the gain in dB when the phase is at zero degrees. The value of gain margin is 19.5dB.

Example analyzer output for the phase margin measurement and loop bandwidth is shown in Figure 30.

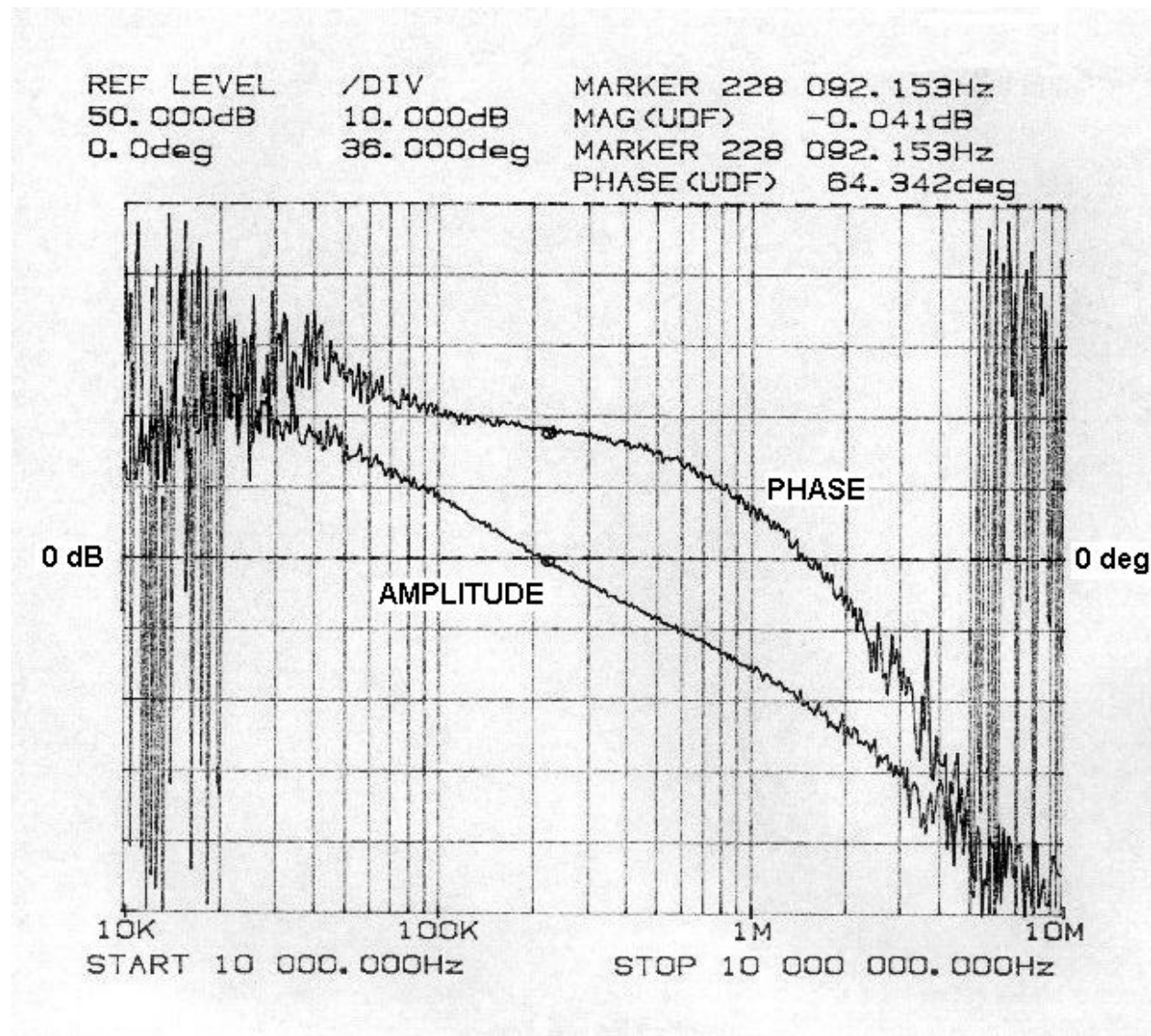


Figure 30: Phase Margin value measurement. The value of the phase margin is determined by measuring the phase in degrees when the gain is unity. The value of phase margin is 92.15 degrees. The value of the loop bandwidth is 228kHz.

5 PAC Controller Program

The following sections describe the motivation and development environment for creating the PAC Controller Program. The benefits and features of using the program are discussed. High level design details and GUI interface descriptions are also given. A high level mechanism to control PAC parameters is essential for testing and verification. The mechanism will allow for accurate and consistent control without the inconvenience of low level command sequences.

5.1 Overview

The PAC Controller Program is a GUI application that can be used to execute a test case using high-level commands. The program was developed using C++ Builder 5.0 Standard [14] along with the ZComm Serial Communication Component [15]. The test program was built using a multi-threaded approach.

The motivation for creating this program was the lack of existing automation in configuring settings. The only method of testing was by writing a RFTTest script that is basically a text file of registers and their contents. Each script was manually compiled and special conversions to determine register contents were required that were not trivial. Another disadvantage to script writing is the fact that only a complete register writes were possible. Ability to perform a masked write on specific bits in a register is not available. Using RFTTest scripts in the testing process is very tedious and not flexible.

The key requirements necessary to obtain an efficient means of creating reproducible and easy to configure test cases are the ability to write to specific bits in a single register, the ability to save setups for future execution, the ability to be compatible with other forms of testing automation at Conexant Systems, and the ability to create new test cases with ease.

The ability to write to specific bits in a single register is achieved by first reading the current contents of the register. With the returned register contents, the new data is masked over the read data while preserving the contents that are not affected by the write. Once the data is masked, a write request is executed and the new data is written. The final step is to read the value once again and verify that the correct value was written.

The ability to save setups for future use is accomplished by saving the current settings to a configuration file. Providing the ability to save configuration files, tests can be easily rerun on demand and there is no ambiguity when running different test cases with different settings.

Compatibility with other forms of testing automation at Conexant was addressed by determining the different types of formats for configuration files. Each department has their own version of configuration scripts that are required. Having the ability to convert my configuration scripts to their format allows me to easily perform tests on their setups to either show or verify that a device is working or not working correctly.

The ability to change settings with ease is one of the keys elements driving the project. With the ability to change settings, any operator of the program without prior knowledge of the system can use the high level controls to adjust settings without dealing with the low level commands.

The automated program has the ability to produce log files, execute manual reads and writes, save register dumps, and obtain monitor information from the device. The program was created in a manner that will allow future developers to create their own GUI programs to control other aspects of the IA device. The control engine that I created allows the developers to concentrate on creating test cases instead of dealing with device protocol issues.

5.2 GUI Interface

The GUI interface was designed in a manner that helps the operator to visualize how the system flows. The layout of the controls resembles the actual control loop. Each block has a button associated with it. If a button for a particular block is selected, a tool windows pops up with the related configuration settings. If a value is selected from the tool window, it shows up on the main window underneath the block. Some blocks have more than one setting, so the corresponding labels are in a column format. The radio group boxes represent special control signals and signal flow. Figure 31 illustrates the screen of the test program.

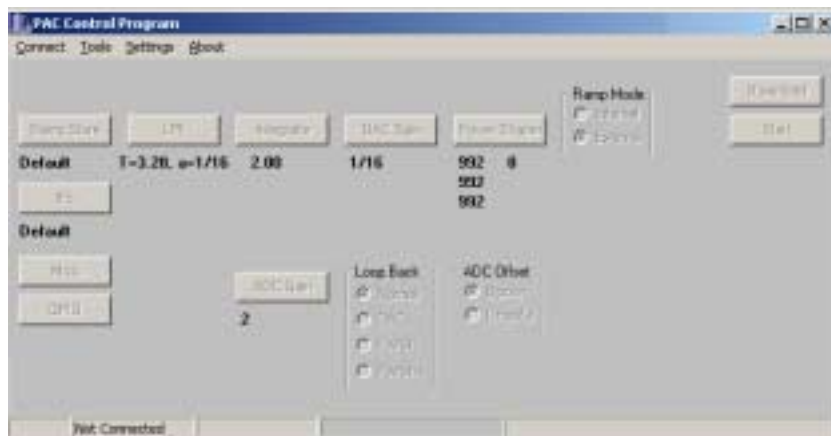


Figure 31: PAC Controller Program GUI interface screen. The interface resembles the actual control loop structure.

Since the Ramp Store and IFS Store are arrays of settings they have special tool windows associated with them. Different ramp profiles and IFS profiles can be saved with user comments. Figure 32 illustrates the Ramp and IFS tool windows.

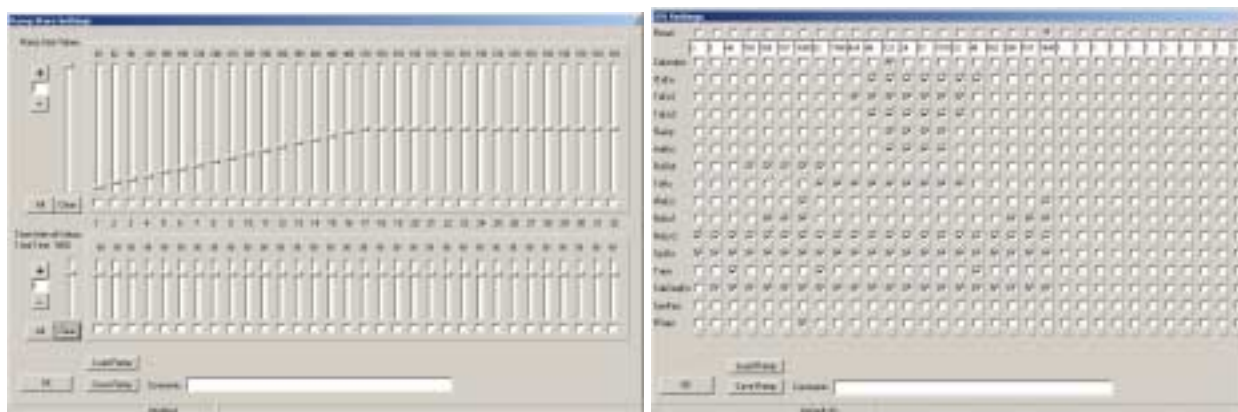


Figure 32: Ramp Store and IFS Store control interface screen. These two interface screens allow direct access to control the Ramp Store and IFS Store functions.

5.3 Command Object

The command object is the low level object that controls request messages. The command object is responsible for the read, write, and masked-write commands. The object encapsulates all the fundamental communication issues that are needed to communicate to the device. A command can be created and a call of the execute function makes all the appropriate requests necessary to perform the command. Figure 33 shows the attributes and methods that the object contains.

Command Object
Address Value Length Start End Mask Type Confirm Message Error State
Read IA Write IA Masked Write IA Get Data Execute

Figure 33: Command Object properties and methods. The command object controls the request messages to the system.

5.3.1 Read Command

The read command is used when a register value needs to be retrieved for function verification or monitor purposes. The user specifies the first address to read and number of consecutive registers to instantiate a read command. Figure 34 shows a flow diagram of how the read command is executed.

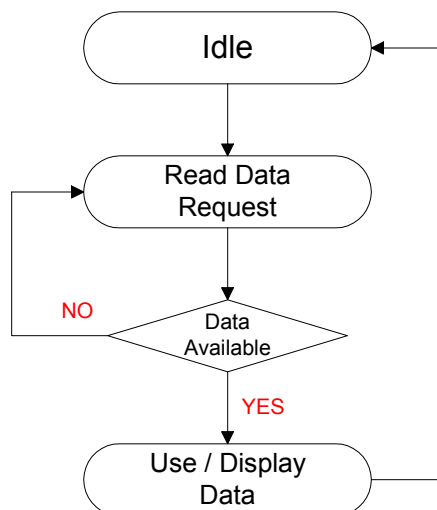


Figure 34: Read command flow diagram. The read command is used when a register value needs to be retrieved for verification or monitor purposes.

The initial state of a read command is set to Idle. The next state, once the command is executed, becomes the Read Data Request and the program generates the appropriate primitive request

commands that are then sent to the device. The next state is set to Use/Display Data. The state machine waits for data to become available from the device. Once the data is received, the data is used or displayed by the program, depending on the original request. The command then returns to the Idle state where it awaits another request.

5.3.2 Write Command

The write command is used when a register value needs to be changed for test case purposes. The user specifies the address to write to and the data associated with the write to instantiate a write command. Figure 35 shows the flow of how the write command is executed.

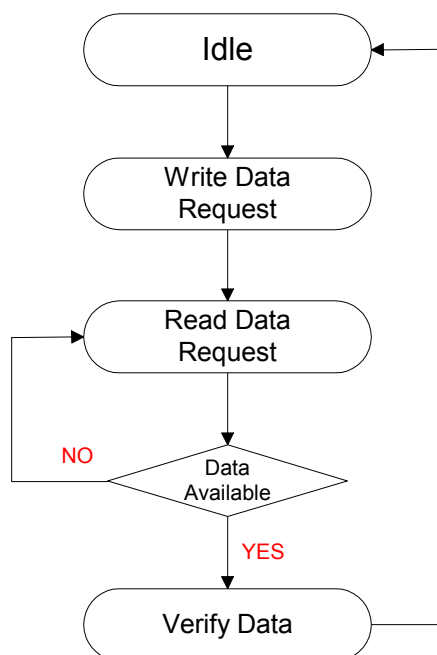


Figure 35: Write command flow diagram. The write command is used when a register value needs to be changed.

The initial state of a write command is set to Idle. The next state, once the command is executed, becomes the Write Data Request and the program generates the appropriate primitive request commands that are sent to the device. Once the write is complete, the next state becomes the Read Data Request that basically reads back the value from the register that was just written to. The state machine waits for data to become available from the device. Once the data is received, the program verifies that the value written was equal to value that is currently in the register. The command then returns to the Idle state where it awaits another request.

5.3.3 Masked Write Command

The masked write command is used when specific bits in a register need to be changed for test case purposes. The user specifies the address to write to, the start- and stop-bit number, and the mask value to instantiate a masked write command. Figure 36 shows the flow of how the masked write command is executed.

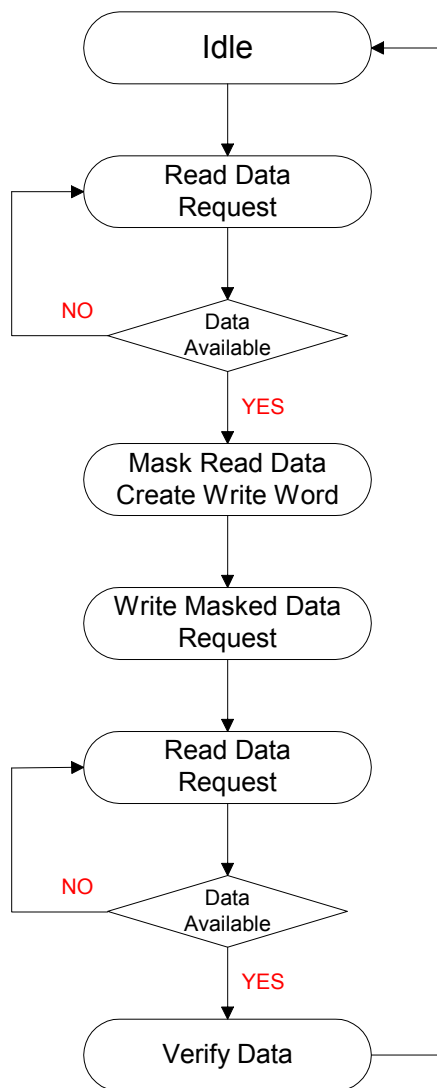


Figure 36: Masked Write command flow diagram. The masked write command allows for specific bits in a register to be changed.

The initial state of a masked write command is set to Idle. The function combines the read and write commands to perform a read-write-read sequence. The next state, once the command is executed, becomes the Read Data Request and the program generates the appropriate primitive

request commands that are sent to the device. The state machine waits for data to become available from the device. Once the data is received, the program combines the new data and the old data and generates the appropriate word to be written to the device. Once complete, the next state becomes the Write Data Request and the program generates the appropriate primitive request commands that are sent to the device. Once the write is complete, the next state becomes the Read Data Request that basically reads back the value from the register that was written to. The state machine waits for data to become available from the device. Once the data is received, the program verifies that the value written is equal to value that is currently in the register. The command then returns to the Idle state where it awaits another request.

5.4 Confirm Message Object

The confirm message object is responsible for deciphering of messages returned from the device. Figure 37 shows the attributes and methods that the object contains.

Confirm Message Object
Message Type Message Length Tag Address Data State
Get Type Get Address Get Data Process

Figure 37: Command Object properties and methods. The confirm message object is responsible for deciphering messages from device.

Figure 38 shows the format of the message to be deciphered.

M e s s a g e T y p e		T a g	M e s s a g e L e n g t h		
X	X	A d d r e s s		R W A I	X X
D a t a [0]					
⋮					
D a t a [M e s s a g e L e n g t h - 2]					

Figure 38: Confirmation Message format. The format of the message includes two header lines and any number of data lines up to the maximum allowed.

The confirm message processor function parses through the confirm message and extracts the information required. Figure 39 illustrates the process state machine.

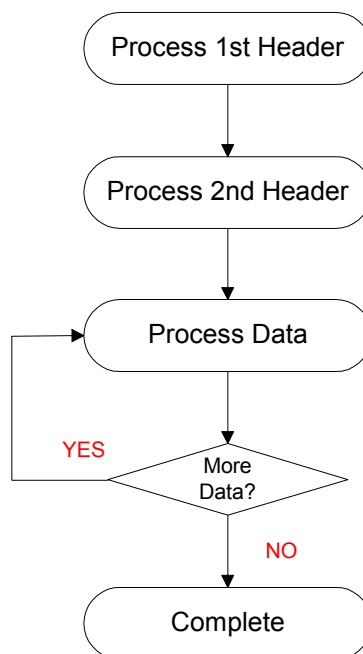


Figure 39: Confirm Message process flow diagram. The confirm message function extracts the message information to be processed and used.

The processor state machine works its way through the received bytes, decipheres the header information, and sets the object properties accordingly. The data portion of message will continue to be processed until no data is left. The data length can vary by size. However, the size is known from the message header.

5.5 Debugging Capability

The debugging capabilities of the program allow the operator to easily determine when errors have occurred. All types of requests to and from the device are captured with relevant information about the request. The operator can monitor the data sent and returned. The files can be saved and filtered for use by the operator.

Additional debugging capabilities are available when the register dump utility is used. The utility sequences through and gathers all the current register contents in the IA. The operator has the ability to have a snap shot of the system to easily determine differences in test setups and for keeping records of what has transpired.

5.6 Test Saving Ability

The test saving feature is one of the key features of the program, allowing the creation of verification configurations with minimum effort and maximum consistency. Each setup can be saved and loaded when required. Configuration of multiple ramp profiles and IFS profiles allows the operator to create multiple test cases with different profiles. The ease of loading a script allows for more functional verification in less time.

The added feature of exporting scripts and configurations to different formats is highly valuable. Extra time and human error in recreating configuration files for different test setups is eliminated. The program supports the feature where an operator can create a settings file for another test system with minimal effort, which results in faster debugging and error correction. With such a complex system with its hundreds of configurations, the export feature will quicken problem resolution when different engineering groups are working together.

6 DAC Power On Transient

In this section, we discuss the problem, investigation and resolution to an unwanted overshoot in the PAC Loop on power up. The problem is introduced and a digital integrated solution is investigated and implemented to resolve the problem. The integrated solution provides a substantial increase in performance.

6.1 Problem

The PAC enable signal from the *IFS* is used to power on the blocks of the PAC. Whenever any type of electrical device is powered on, the ability for a perfect power on sequence is desired but difficult to achieve. The reduction of the transients that occur at power-on is an important factor when dealing with a power amplifier. A transient to the control port on the power amplifier can cause switching transients and spectral splatter. If these types of transients occur, interference with users on adjacent channels can result. Certain specifications need to be met and they are described in Section 4.2.4. Figure 40 shows the current configuration of the interface between the *DAC* output and the control port of the power amplifier.

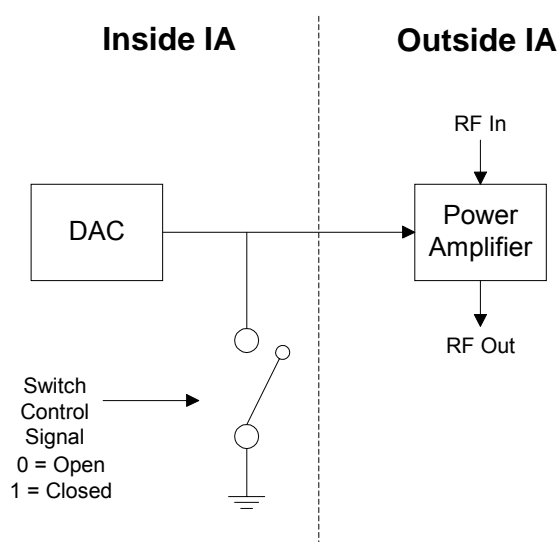


Figure 40: DAC and Power Amplifier interface with control signal to clamp the DAC output to a value of zero volts.

The existing switch control signal is the inverted PAC enable signal. Basically, if the PAC enable is low, the switch control signal will be high, causing the output to be forced to zero,

guaranteeing the power amplifier is off. The moment the PAC enable goes high, the entire loop is enabled and the DAC powers on and its value is fed to the PA. The *DAC* actually overshoots to 400 mV once it powers on to its pedestal value of 300 mV, as described in Section 4.1.2. The overshoot of about 33% that is not acceptable because it is enough for the power amplifier to begin transmitting.

6.2 Investigations

It was determined that a different switch control signal was needed to reduce the voltage transient. The *DAC* needed to be powered on in a manner that the switch would be closed until the transient settled out. The power amplifier does not need to turn on until the rising edge of the Ramp enable signal. The additional control signal was created and Nanded with the PAC enable signal. A secondary control signal required to go low on the rising edge of the Ramp Enable signal must be reset at the end of the transmit burst when PAC enable goes low. Figure 41 illustrates the logic used to create this characteristic.

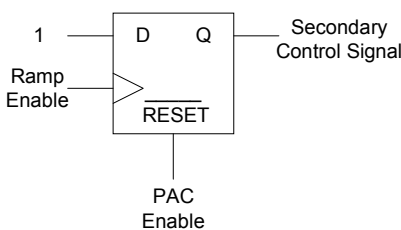


Figure 41: Secondary control signal logic generator. A simple flip-flop is used to generate the control signal.

Figure 42 shows the timing diagram for the output switch control.

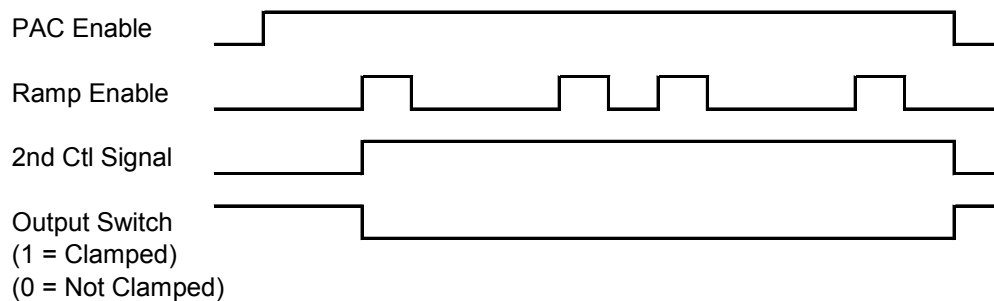


Figure 42: Output switch control logic. If the output switch is clamped the DAC output is held to a zero volt level and when the output switch is not clamped the DAC output is passed to the PA.

6.3 Solution

The new control signal has allowed the DAC power on transient to be reduced down to 30 mV of overshoot. The reduction in overshoot from 33% to 10% resulted in a significant reduction in switching transients. Figure 43 illustrates the reduction of the overshoot of the

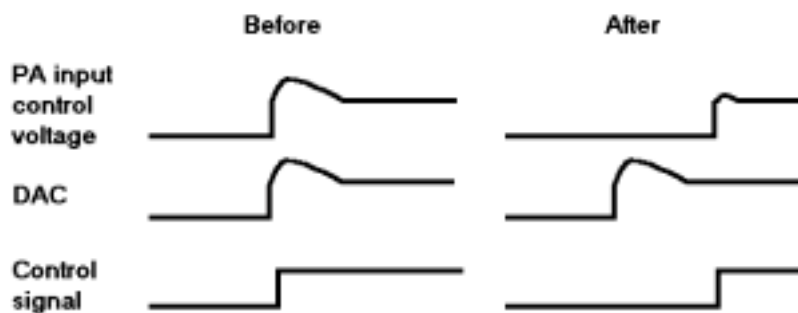


Figure 43: DAC power-on transient resolution results. The input control voltage is shown in the before and after diagram indicating a large reduction in overshoot amount. The resolution is based on delaying the control signal from turning on during the DAC power on sequence.

7 Conclusion And Future Work

The main purpose of this work is to efficiently test and verify the integrated power amplifier controller loop for functionality and performance. This section provides a summary of the results and possible future considerations in the area of the integrated power amplifier controller loop.

7.1 Conclusion

For this thesis we present four main topics related to the testing and verification of a digital power amplifier control loop: The digital PAC was achieved by incorporating all the digital blocks in the mixed signal device. In comparison to analog PAC it has been shown that there is a large variety of loop characteristic control. Modular verification environment methodologies and its benefits, creation of test cases with the associated modular components, the development of a test program that encapsulates the test environment ideologies, and investigation and solution to a problem in the control loop.

The test environment combines a stimulus generator, transaction verification module, device under test, and response checker. The stimulus generator provides the stimuli for each test case. The transaction verification module provides the usable information for the device under test and response checker. The response checker verifies the actual results of the test case to the expected values.

Using the modular verification environment approach for a platform, we have produced a reliable and robust test platform. The fundamental achievement of this platform is the ability to test the system with little programming knowledge of the device. The ease of testing allows the operator to concentrate on the real problems of the system, without worrying about basic testing problems. Eliminating the uncertainty of a manual test system, error resolution is much quicker and a reliable product can be put into production in shorter time.

Each block of the PAC loop was tested for proper functionality. All scenarios ranging from maximum to minimum, and random to nominal inputs were tested. The performance of the entire loop was tested. Output power, burst timing, spectrum due to modulation, and spectrum due to switching transients was verified. The closed loop stability of the loop was verified using

a technique that measured the effects of sweep voltage through the loop when operating a nominal level.

The test cases discussed in this thesis cover the entire design of the PAC. Full coverage of all the functional blocks and their performance has allowed for a high confidence for introducing an integrated digital PAC solution. Developers and testers requiring the ability to transmit data for their tests can be certain of the ability of the power amplifier that it can provide its function according to specification.

The use of a GUI interface simplifies the user interaction with the device. The command object is the low level object that controls all read, write and masked writes requests. The confirm message object is the low level object that controls all returned messages from the device. The messages can contain register or status information. The test program provides debugging capabilities that allow the operator to determine errors quickly. A register dump utility will take a snap shot of the register contents and helps to determine the exact settings of the system. The test case saving feature allows the operator to create and save setups fast and efficiently.

The test program allows other developers to inherit the program engine when developing a test program for other functionalities of the device and for other devices. Since the test program deals with all the low level issues with a high level interface, it can now be used as a building block to a higher level of integration. Future developers can build upon my work to create the perfect test system: “The one button push tests”.

The power on transient of the DAC caused an overshoot in the control voltage to the power amplifier. The overshoot created spectral splatter and caused the PAC loop to fail performance tests. Investigations resulted in determining that a second control signal was needed to latch the output of the DAC low until transient leveled out. A significant reduction in overshoot value increased the performance of the loop.

This thesis discussed the resolution to the issue of a power on transient in the PAC. Using digital control logic yielded more predictable results than an analog solution. Furthermore, PC board modifications, and additional parts are not required when using an integrated logic solution as opposed to a discrete solution.

7.2 Future Work

Two areas of future work for this project are an expansion of the test suite to become more robust and fully automated, and to design an integrated detector.

While there are plans of additional development for the power amplifier controller, the foundation has been developed and tested. The level of reliability and program functionality now available is a good starting point for future developing. We can improve on the modules to communicate with other test equipment, to retrieve data samples, and other monitor information. With full scripting control over test equipment, fully automated test suites can be created. With a fully automated test suite, recurring costs, resources, and time can be saved, resulting in improved production of these high-volume circuits.

Future considerations will be in determining an integrated solution for the discrete detector. Once again by moving towards an integrated design solution, come many benefits. One of the main issues of the detector is creating it with enough dynamic range while balancing the cost to benefit ratio when designing and implementing it.

References

- [1] Conexant Systems, “Multi-Slot GSM/GPRS System Solution,” <http://www.conexant.com/default.sph/SaServletEngine.class/Web/products/prodspecs.jsp?PartId=670&SpecCatId=21#> (current January 2002), pp. 3-4.
- [2] Conexant Systems, “Multi-Slot GSM/GPRS System Solution,” <http://www.conexant.com/default.sph/SaServletEngine.class/Web/products/prodspecs.jsp?PartId=670&SpecCatId=21#> (current January 2002), p. 2.
- [3] ARM Ltd., “ARM7™ Thumb® Family,” http://www.arm.com/armtech.ns4/html/ARM7_Thumb (current January 2002)
- [4] Conexant Systems, “Integrated Analog Device for GSM Applications,” <http://www.conexant.com/default.sph/SaServletEngine.class/Web/products/prodspecs.jsp?PartId=210&SpecCatId=1#> (current January 2002), p. 8.
- [5] Conexant Systems, “Integrated Analog Device for GSM Applications,” <http://www.conexant.com/default.sph/SaServletEngine.class/Web/products/prodspecs.jsp?PartId=210&SpecCatId=1#> (current January 2002), pp. 7-8.
- [6] Conexant Systems, “Integrated Analog Device for GSM Applications,” <http://www.conexant.com/default.sph/SaServletEngine.class/Web/products/prodspecs.jsp?PartId=210&SpecCatId=1#> (current January 2002), p. 11.
- [7] J. Komaili, “D20518 Power Amplifier Control Design and Behavioral Simulation,” August 2001, p. 12.
- [8] Cadence Design Systems Inc., “Verification Reuse Methodology—Overview,” <http://www.cadence.com/whitepapers/VRMethodology.html> (current January 2002).
- [9] Microsoft Corporation, “Microsoft Excel,” <http://www.microsoft.com/office/excel/default.htm> (current January 2002).
- [10] *GM 11.10-1 version 4.20.0, Digital cellular telecommunications system – Mobile Station conformance specification*, European Telecommunications Standards Institute, France, November 1997, pp. 151-152.
- [11] *GM 11.10-1 version 4.20.0, Digital cellular telecommunications system – Mobile Station conformance specification*, European Telecommunications Standards Institute, France, November 1997, pp. 151-152.
- [12] *GM 11.10-1 version 4.20.0, Digital cellular telecommunications system – Mobile Station conformance specification*, European Telecommunications Standards Institute, France, November 1997, pp. 151-152.
- [13] *GM 11.10-1 version 4.20.0, Digital cellular telecommunications system – Mobile Station conformance specification*, European Telecommunications Standards Institute, France, November 1997, pp. 158-159.
- [14] Borland Software Corporation, “C++ Builder,” <http://www.borland.com/bcppbuilder/> (current January 2002).

- [15] ZCOMM, "Serial Communication Component," <http://www.medasis.com/tools/zcomm.htm> (current January 2002).