

Operation and Service Manual

Bessel & Butterworth Filter

SIM965



Stanford Research Systems

Revision 1.7 • August 3, 2011

Certification

Stanford Research Systems certifies that this product met its published specifications at the time of shipment.

Warranty

This Stanford Research Systems product is warranted against defects in materials and workmanship for a period of one (1) year from the date of shipment.

Service

For warranty service or repair, this product must be returned to a Stanford Research Systems authorized service facility. Contact Stanford Research Systems or an authorized representative before returning this product for repair.

Information in this document is subject to change without notice.

Copyright © Stanford Research Systems, Inc., 2005 – 2011. All rights reserved.

Stanford Research Systems, Inc.
1290–D Reamwood Avenue
Sunnyvale, CA 94089 USA
Phone: (408) 744-9040 • Fax: (408) 744-9049
www.thinkSRS.com • e-mail: info@thinkSRS.com

Printed in U.S.A.

Document number 9-01597-903



SIM965 Bessel & Butterworth Filter

Contents

General Information	iii
Safety and Preparation for Use	iii
Symbols	iv
Notation	v
Specifications	vi
1 Operation	1 – 1
1.1 Overview	1–2
1.2 Nominal transfer functions	1–3
1.3 Front-Panel Operation	1–8
1.4 Clock Stopping	1–9
1.5 SIM Interface	1–10
2 Remote Operation	2 – 1
2.1 Index of Common Commands	2–2
2.2 Alphabetic List of Commands	2–4
2.3 Introduction	2–6
2.4 Commands	2–6
2.5 Status Model	2–16
3 Circuitry	3 – 1
3.1 Circuit Descriptions	3–2
3.2 Parts Lists	3–4
3.3 Schematic Diagrams	3–6

General Information

The SIM965 Analog Filter, part of Stanford Research Systems' Small Instrumentation Modules family, is a continuous-time, programmable filter capable of high-pass and low-pass operation as a Butterworth or Bessel filter.

Safety and Preparation for Use

The front-panel input, front-panel output, and the rear-panel output coaxial (BNC) connectors in the SIM965 are referenced to the Earth, and their outer casings are grounded. No dangerous voltages are generated by the module.

 **WARNING**

Do not exceed ± 15 volts to the Earth at the center terminal of any BNC connector. Do not install substitute parts or perform unauthorized modifications to this instrument.

The SIM965 is a single-wide module designed to be used inside the SIM900 Mainframe. Do not turn on the power until the module is completely inserted into the mainframe and locked in place.

Symbols you may Find on SRS Products

Symbol	Description
	Alternating current
	Caution - risk of electric shock
	Frame or chassis terminal
	Caution - refer to accompanying documents
	Earth (ground) terminal
	Battery
	Fuse
	On (supply)
	Off (supply)

Notation



WARNING

The following notation will be used throughout this manual.

A warning means that injury or death is possible if the instructions are not obeyed.



CAUTION

A caution means that damage to the instrument or other equipment is possible.

Typesetting conventions used in this manual are:

- Front-panel buttons are set as [Button];
[Adjust ▲▼] is shorthand for “[Adjust ▲] & [Adjust ▼]”.
- Front-panel indicators are set as *Overload*.
- Remote command names are set as *IDN?.
- Literal text other than command names is set as OFF.

Remote command examples will all be set in monospaced font. In these examples, data sent by the host computer to the SIM965 are set as *straight teletype font*, while responses received by the host computer from the SIM965 are set as *slanted teletype font*.

Specifications

Performance Characteristics

		Min	Typ	Max	Units
Input	Impedance		1		MΩ
	Coupling	AC or DC			
	Gain	1×			
	AC-coupling time const.		1		s
	Range, 48 dB/oct Butterworth 36 dB/oct Butterworth all others	-5 -7 -10		+5 +7 +10	V
Filter	Band	low-pass or high-pass			
	Cutoff frequency	1.00		5×10^5	Hz
	Resolution		3		digits
	Accuracy	-1		+1	%
	Type	Butterworth, Bessel			
	Rolloff	12, 24, 36, 48			dB/octave
Output	Noise	< 200μV _{rms} (1 MHz bandwidth)			
	THD	0.01 % (-80 dB) at 1 kHz			
Operating	Temperature	0		40	°C, non-condensing
	Power	+5, ±15			V DC
	Supply current, +5 V ±15 V		100		mA
			300		mA

General Characteristics

Interface	Serial (RS-232) through SIM interface
Connectors	BNC (2 front, 1 rear)
	DB-15 (male) SIM interface
Weight	3 lbs
Dimensions	1.5'' W × 3.6'' H × 7.0'' D

1 Operation

This chapter gives you the necessary information to get started quickly with the SIM965 Analog Filter.

In This Chapter

1.1	Overview	1-2
1.2	Nominal transfer functions	1-3
1.2.1	Butterworth filters	1-3
1.2.2	Bessel filters	1-3
1.2.3	High-pass filters	1-6
1.3	Front-Panel Operation	1-8
1.3.1	Frequency	1-8
1.3.2	Type	1-8
1.3.3	Filter	1-8
1.3.4	Slope	1-8
1.3.5	Input	1-8
1.3.6	Output	1-9
1.4	Clock Stopping	1-9
1.5	SIM Interface	1-10
1.5.1	SIM interface connector	1-10
1.5.2	Direct interfacing	1-10

1.1 Overview

The SIM965 Analog Filter is a continuous-time, digitally-programmable filter with fully analog signal paths. By using a modified state-variable circuit design, the SIM965 provides a variety of programmed filter configurations. From the front panel, the user can select a Butterworth filter, for maximum passband gain flatness, or a Bessel filter, for minimum pulse overshoot and constant time delay in the passband. The user can also select either a low-pass or high-pass filter, and change the order of the filter between 2nd, 4th, 6th, or 8th order, corresponding to 12, 24, 36 or 48 dB/octave roll-off in the stop band.

For any filter configuration, a single continuous parameter, the “cut-off frequency,” f_c can be set with 3-digit resolution (and $\pm 1\%$ accuracy) in the range of 1 Hz to 500 kHz. For Butterworth filters, f_c corresponds to the frequency at which the filter response is -3 dB. For Bessel filters, f_c is determined so that the far-stop-band attenuation slope asymptotically approaches that of the Butterworth filter.



Figure 1.1: The SIM965 front and rear panels.

1.2 Nominal transfer functions

The SIM965 circuitry is configured to provide a near-ideal Butterworth or Bessel filter transfer function for the user.

1.2.1 Butterworth filters

For a Butterworth filter, the nominal transfer function for an n th-order low pass filter is given by:

$$G_n(f) = \sqrt{\frac{1}{1 + \eta^{2n}}} \quad (1.1)$$

where, for low-pass filters, $\eta = (f/f_0)$, and $f_0 = f_c$, the -3 dB frequency of the response function. Figures 1.2 and 1.3 show the frequency and step response for Butterworth low-pass filters.

1.2.2 Bessel filters

For Bessel filters, the nominal transfer function for an n th-order low pass filter is given by:

$$G_n(f) = \sqrt{\frac{1}{\left(\frac{B_N}{b_{0N}}\right)^2 + \left(\frac{P_N}{b_{0N}}\right)^2}} \quad (1.2)$$

where, for low-pass filters, $\eta = (f/f_0)$, and B_N , P_N , and b_{0N} are determined iteratively, based on

$$B_N = (2N - 1)B_{(N-1)} - \eta^2 B_{(N-2)}$$

with $B_0 = 1$, $B_1 = 1$,

$$P_N = (2N - 1)P_{(N-1)} - \eta^2 P_{(N-2)}$$

with $P_0 = 0$, $P_1 = \eta$, and

$$b_{0N} = (2N - 1) * b_{0(N-1)}$$

with $b_{00} = 1$. Figures 1.4 and 1.5 show the frequency and step response for Bessel low-pass filters.

The SIM965 uses a frequency normalization for Bessel filters such that the far-stop-band response asymptotically approaches that of the same-order Butterworth filter. Table 1.1 gives the scaling factors to obtain the formal f_0 (needed for the Bessel formulae) and the actual -3 dB frequency, in terms of the SIM965 setting f_c . For example, for a 6-pole low-pass Bessel filter with $f_c = 100$ Hz can be calculated using $f_0 = 0.21409 \times f_c$, or 21.409 Hz.

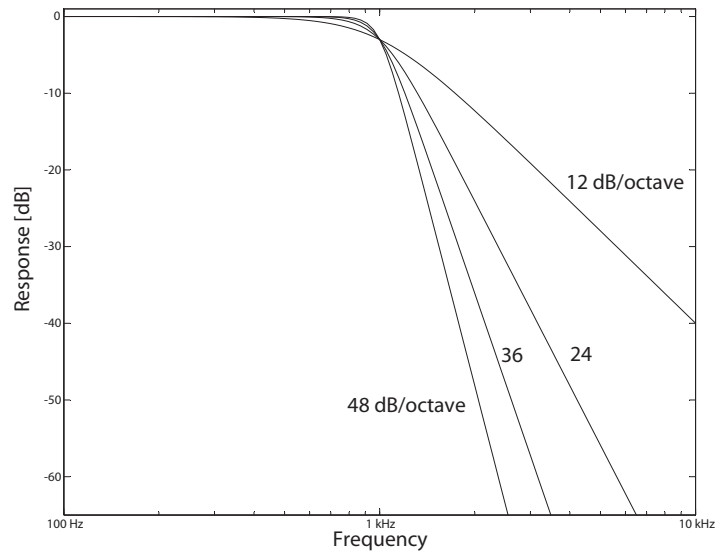


Figure 1.2: The nominal frequency response for Butterworth low-pass filters of various orders. All filters are tuned to $f_c = 1$ kHz.

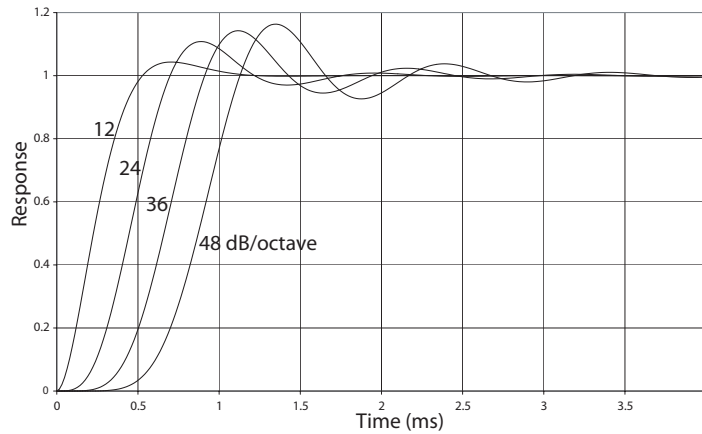


Figure 1.3: The nominal step response for Butterworth low-pass filters of various orders. All filters are tuned to $f_c = 1$ kHz.

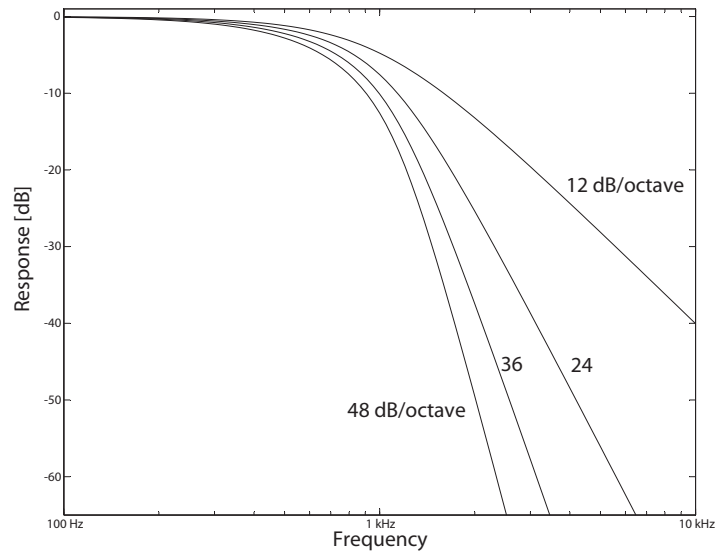


Figure 1.4: The nominal frequency response for Bessel low-pass filters of various orders. All filters are tuned to $f_c = 1$ kHz.

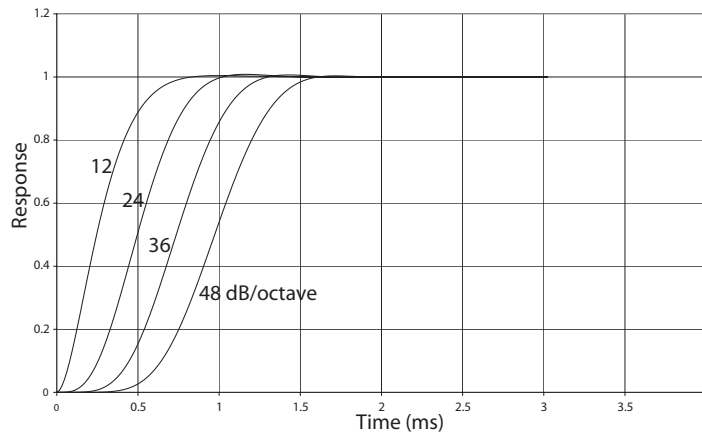


Figure 1.5: The nominal step response for Bessel low-pass filters of various orders. All filters are tuned to $f_c = 1$ kHz.

Order	f_0	$f_{-3\text{dB}}$
2	$0.57739 \times f_c$	$0.7862 \times f_c$
4	$0.31243 \times f_c$	$0.6604 \times f_c$
6	$0.21409 \times f_c$	$0.5787 \times f_c$
8	$0.16283 \times f_c$	$0.5177 \times f_c$

Table 1.1: Bessel filter normalization factors

1.2.3 High-pass filters

To obtain the formulae for a high-pass Butterworth filter, simply substitute $\eta = (f_0/f)$ into Equation 1.1.

For a high-pass Bessel filter, one similarly substitutes $\eta = (f/f_0)$ into Equation 1.2. However, the scaling factors from Table 1.1 must be inverted. For example, a 6-pole high-pass Bessel filter with $f_c = 100$ Hz can be calculated using $f_0 = f_c \div 0.21409$, or 467.09 Hz. Figures 1.6 and 1.7 show the frequency response for Butterworth and Bessel high-pass filters.

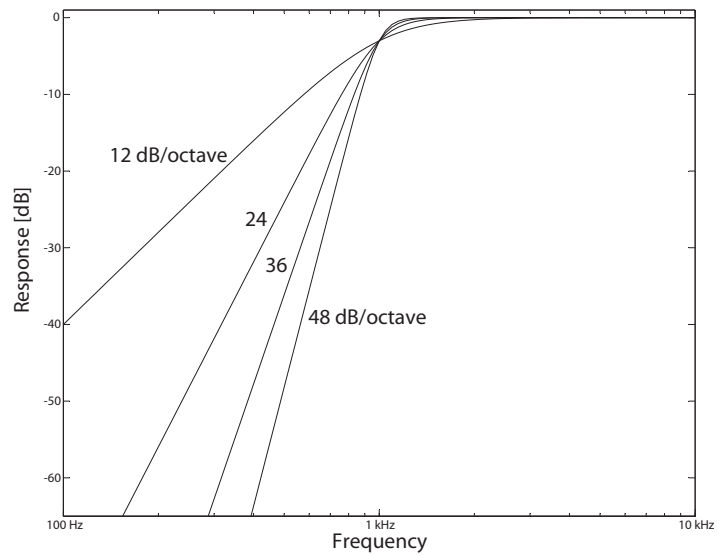


Figure 1.6: The nominal frequency response for Butterworth high-pass filters of various orders. All filters are tuned to $f_c = 1$ kHz.

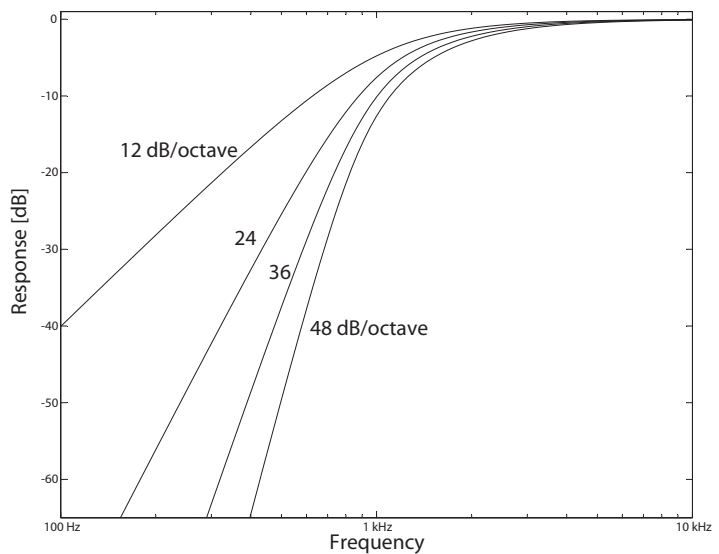


Figure 1.7: The nominal frequency response for Bessel high-pass filters of various orders. All filters are tuned to $f_c = 1$ kHz.

1.3 Front-Panel Operation

All settings of the SIM965 can be set from the front panel (see Figure 1.1).

1.3.1 Frequency

The cutoff frequency can be incremented or decremented using the [Freq. ▲▼] buttons. Pressing either [Freq. ▲] or [Freq. ▼] once will cause the least significant digit in the display to increment (or decrement) by one. If the button is held down, the display will begin to change at a steadily-increasing rate, accelerating to allow large f_c changes to be made easily. Note that the circuitry is not reprogrammed until the button is released.

1.3.2 Type

The [Type] button allows the user to toggle between Butterworth or Bessel filter type.

1.3.3 Filter

The [Filter] button allows the user to toggle between high pass or low pass filter pass band.

1.3.4 Slope

The [Slope] button allows the user to cycle through the four available stop band roll-off rates: 12, 24, 36 and 48 dB/octave.

1.3.5 Input

Input signals to the SIM965 at the front-panel BNC connector in the "Input" block. If the input signal exceeds the specified ± 10 V range, the *Ovld* indicator will light and remain on as long as the signal exceeds the specified input range.

1.3.5.1 Couple

The [Coupling] button allows the user to toggle the input coupling of the SIM965 between AC and DC coupling. When AC-coupled, the input is high-pass filtered by a single-pole RC filter with a 1 second time constant.

1.3.6 Output

The filtered signal is available from the SIM965 at the front-panel BNC connector in the “Output” block. A second output connector is available on the rear panel as well. Each output is (separately) connected the filter circuitry through an internal 50 Ω resistor.

1.4 Clock Stopping

The microprocessor clock of the SIM965 stops if the module is idle, “freezing” the digital circuitry. The following actions “wake up” the clock:

1. A power-on.
2. A press of a front-panel button.
3. Activity (send or receive) at the remote interface.
4. An overload.

The clock runs for as long as is necessary to complete a filter setting adjustment, or to communicate the output of a query through the remote interface. However, the clock will remain active for as long as the overload condition exists.

This default behavior can be modified with the remote command *AWAK*. Setting *AWAK ON* will prevent the clock from stopping. The module returns to *AWAK OFF* upon power-on.

1.5 SIM Interface

The primary connection to the SIM965 Analog Filter is the rear-panel DB-15 SIM interface connector. Typically, the SIM965 is mated to a SIM900 Mainframe via this connection, either through one of the internal Mainframe slots, or the remote cable interface.

It is also possible to operate the SIM965 directly, without using the SIM900 Mainframe. This section provides details on the interface.



CAUTION

The SIM965 has no internal protection against reverse polarity, missing supply, or overvoltage on the power supply pins. Misapplication of power may cause circuit damage. SRS recommends using the SIM965 together with the SIM900 Mainframe for most applications.

1.5.1 SIM interface connector

The DB-15 SIM interface connector carries all the power and communications lines to the instrument. The connector signals are specified in Table 1.2

Pin	Signal	Direction Src ⇒ Dest	Description
1	SIGNAL_GND	MF ⇒ SIM	Ground reference for signal
2	-STATUS	SIM ⇒ MF	Status/service request (GND = asserted, +5 V = idle)
3	RTS	MF ⇒ SIM	HW handshake (unused in SIM965)
4	CTS	SIM ⇒ MF	HW handshake (unused in SIM965)
5	-REF_10MHZ	MF ⇒ SIM	10 MHz reference (no connection in SIM965)
6	-5 V	MF ⇒ SIM	Power supply (no connection in SIM965)
7	-15 V	MF ⇒ SIM	Power supply
8	PS_RTN	MF ⇒ SIM	Power supply return
9	CHASSIS_GND		Chassis ground
10	TXD	MF ⇒ SIM	Async data (start bit = "0" = +5 V; "1" = GND)
11	RXD	SIM ⇒ MF	Async data (start bit = "0" = +5 V; "1" = GND)
12	+REF_10MHz	MF ⇒ SIM	10 MHz reference (no connection in SIM965)
13	+5 V	MF ⇒ SIM	Power supply
14	+15 V	MF ⇒ SIM	Power supply
15	+24 V	MF ⇒ SIM	Power supply (no connection in SIM965)

Table 1.2: SIM Interface Connector Pin Assignments, DB-15

1.5.2 Direct interfacing

The SIM965 is intended for operation in the SIM900 Mainframe, but users may wish to directly interface the module to their own systems without the use of additional hardware.

The mating connector needed is a standard DB-15 receptacle, such as Tyco part # 747909-2 (or equivalent). Clean, well-regulated supply voltages of +5, ± 15 V DC must be provided, following the pin-out specified in Table 1.2. Ground must be provided on pins 1 and 8, with chassis ground on pin 9. Note that internally the SIM965 ties all three of these terminals, pins 1, 8, and 9, together to form the internal ground. The -STATUS signal may be monitored on pin 2 for a low-going TTL-compatible output indicating a status message.

1.5.2.1 Direct interface cabling

If the user intends to directly wire the SIM965 independent of the SIM900 Mainframe, communication is usually possible by directly connecting the appropriate interface lines from the SIM965 DB-15 plug to the RS-232 serial port of a personal computer.¹ Connect RXD from the SIM965 directly to RD on the PC, TXD directly to TD. In other words, a null-modem style cable is *not* needed.

To interface directly to the DB-9 male (DTE) RS-232 port typically found on contemporary personal computers, a cable must be made with a female DB-15 socket to mate with the SIM965, and a female DB-9 socket to mate with the PC's serial port. Separate leads from the DB-15 need to go to the power supply, making what is sometimes know as a "hydra" cable. The pin-connections are given in Table 1.3.

DB-15/F to SIM965	Name
	DB-9/F
10 \longleftrightarrow 3	TxD
11 \longleftrightarrow 2	RxD
5	Computer Ground
	to P/S
7 \longleftrightarrow -15 VDC	
13 \longleftrightarrow +5 VDC	
14 \longleftrightarrow +15 VDC	
1,8,9 \longleftrightarrow Ground (chassis, & P/S return)	

Table 1.3: SIM965 Direct Interface Cable Pin Assignments

¹ Although the serial interface lines on the DB-15 do not satisfy the minimum voltage levels of the RS-232 standard, they are typically compatible with desktop personal computers

1.5.2.2 Serial settings

The initial serial port settings at power-on are: 9600 Baud, 8–bits, no parity, 1 stop bit, and no flow control. The parity can be changed with the **PARI** remote command.

2 Remote Operation

This chapter describes operating the SIM965 over the serial interface.

In This Chapter

2.1	Index of Common Commands	2-2
2.2	Alphabetic List of Commands	2-4
2.3	Introduction	2-6
2.3.1	Power-on configuration	2-6
2.3.2	Buffers	2-6
2.3.3	Device Clear	2-6
2.4	Commands	2-6
2.4.1	Command Syntax	2-7
2.4.2	Notation	2-8
2.4.3	Examples	2-8
2.4.4	Filter Commands	2-9
2.4.5	Serial Communication Commands	2-10
2.4.6	Status Commands	2-10
2.4.7	Interface Commands	2-12
2.5	Status Model	2-16
2.5.1	Status Byte (SB)	2-17
2.5.2	Service Request Enable (SRE)	2-18
2.5.3	Standard Event Status (ESR)	2-18
2.5.4	Standard Event Status Enable (ESE)	2-18
2.5.5	Communication Error Status (CESR)	2-19
2.5.6	Communication Error Status Enable (CESE)	2-19

2.1 Index of Common Commands

symbol	definition
i, j	Integers
f	Floating-point values
z	Literal token
(?)	Required for queries; illegal for set commands
var	parameter always required
{ var }	required parameter for set commands; illegal for queries
[var]	optional parameter for both set and query forms

Filter

FREQ(?) { f }	2 – 9	Filter Frequency
TYPE(?) { z }	2 – 9	Filter type
PASS(?) { z }	2 – 9	Filter pass band
SLPE(?) { i }	2 – 9	Filter slope
COUP(?) { z }	2 – 9	Input coupling

Serial Communications

PARI(?) { z }	2 – 10	Parity
-----------------	--------	--------

Status

*STB? [i]	2 – 10	Status Byte
*SRE(?) [i, j]	2 – 10	Service Request Enable
*CLS	2 – 10	Clear Status
*ESR? [i]	2 – 10	Standard Event Status
*ESE(?) [i, j]	2 – 11	Standard Event Status Enable
CESR? [i]	2 – 11	Comm Error Status
CESE(?) [i, j]	2 – 11	Comm Error Status Enable
OVLDT?	2 – 11	Overload
PSTA(?) { z }	2 – 11	Pulse –STATUS Mode

Interface

*RST	2 – 12	Reset
*IDN?	2 – 12	Identify
*OPC(?)	2 – 12	Operation Complete
CONS(?) { z }	2 – 13	Console Mode
AWAK(?) { z }	2 – 13	Awake mode
LEXE?	2 – 13	Execution Error
LCME?	2 – 14	Command Error
LBTN?	2 – 14	Button
TOKN(?) { z }	2 – 15	Token Mode
TERM(?) { z }	2 – 15	Response Termination

2.2 Alphabetic List of Commands

★

*CLS	2 – 10	Clear Status
*ESE(?) [i,] {j}	2 – 11	Standard Event Status Enable
*ESR? [i]	2 – 10	Standard Event Status
*IDN?	2 – 12	Identify
*OPC(?)	2 – 12	Operation Complete
*RST	2 – 12	Reset
*SRE(?) [i,] {j}	2 – 10	Service Request Enable
*STB? [i]	2 – 10	Status Byte

A

AWAK(?) {z}	2 – 13	Awake mode
-------------	--------	------------

C

CESE(?) [i,]{j}	2 – 11	Comm Error Status Enable
CESR? [i]	2 – 11	Comm Error Status
CONS(?) {z}	2 – 13	Console Mode
COUP(?) {z}	2 – 9	Input coupling

F

FREQ(?) {f}	2 – 9	Filter Frequency
-------------	-------	------------------

L

LBTN?	2 – 14	Button
LCME?	2 – 14	Command Error
LEXE?	2 – 13	Execution Error

O

OVLD?	2 – 11	Overload
-------	--------	----------

P

PARI(?) {z}	2 – 10	Parity
PASS(?) {z}	2 – 9	Filter pass band
PSTA(?) {z}	2 – 11	Pulse –STATUS Mode

S

SLPE(?) {i}	2 – 9	Filter slope
-------------	-------	--------------

T

TERM(?) {z}	2 – 15	Response Termination
TOKN(?) {z}	2 – 15	Token Mode

TYPE(?) {z} 2-9 Filter type

2.3 Introduction

Remote operation of the SIM965 is through a simple command language documented in this chapter. Both set and query forms of most commands are supported, allowing the user complete control of the filter from a remote computer, either through the SIM900 Mainframe or directly via RS-232 (see Section 1.5.2.1).

See Table 1.2 for specification of the DB-15 SIM interface connector.

2.3.1 Power-on configuration

The settings for the remote interface are 9600 baud with no parity and no flow control, and local echo disabled (CONS OFF).

Most of the SIM965 instrument settings are stored in non-volatile memory, and at power-on the instrument returns to the state it was last in when power was removed. Exceptions are noted in the command descriptions.

Reset values of parameters are shown in **boldface**.

2.3.2 Buffers

Incoming data from the host interface is stored in a 32-byte input buffer. Characters accumulate in the input buffer until a command terminator (either <CR> or <LF>) is received, at which point the message is parsed and executed. Query responses from the SIM965 are buffered in a 32-byte output queue.

If the input buffer overflows, then all data in *both* the input buffer and the output queue are discarded, and an error is recorded in the CESR and ESR status registers.

2.3.3 Device Clear

The SIM965 host interface can be asynchronously reset to its power-on configuration by sending an RS-232-style <break> signal. From the SIM900 Mainframe, this is accomplished with the SIM900 SRST command; if directly interfacing via RS-232, then use a serial break signal. After receiving the Device Clear, the interface is reset and CONS mode is turned OFF. Note that this *only* resets the communication interface; the basic function of the SIM965 is left unchanged; to reset the instrument, see *RST.

2.4 Commands

This section provides syntax and operational descriptions for remote commands.

2.4.1 Command Syntax

The four letter mnemonic (shown in **CAPS**) in each command sequence specifies the command. The rest of the sequence consists of parameters.

Commands may take either *set* or *query* form, depending on whether the “?” character follows the mnemonic. *Set only* commands are listed without the “?”, *query only* commands show the “?” after the mnemonic, and *optionally query* commands are marked with a “(?)”.

Parameters shown in { } and [] are not always required. Parameters in { } are required to set a value, and are omitted for queries. Parameters in [] are optional in both set and query commands. Parameters listed without any surrounding characters are always required.

Do *not* send () or { } or [] as part of the command.

Multiple parameters are separated by commas. Multiple commands may be sent on one command line by separating them with semicolons (;) so long as the input buffer does not overflow. Commands are terminated by either <CR> or <LF> characters. Null commands and whitespace are ignored. Execution of command(s) does not begin until the command terminator is received.

tokens Token parameters (generically shown as *z* in the command descriptions) can be specified either as a keyword or integer value. Command descriptions list the valid keyword options, with each keyword followed by its corresponding integer value. For example, to set the response termination sequence to <CR>+<LF>, the following two commands are equivalent:

TERM CRLF —or— TERM 3

For queries that return token values, the return format (keyword or integer) is specified with the TOKN command.

2.4.2 Notation

The following table summarizes the notation used in the command descriptions:

symbol	definition
<i>i,j</i>	Integers
<i>f</i>	Floating-point values
<i>z</i>	Literal token
(?)	Required for queries; illegal for set commands
<i>var</i>	parameter always required
{ <i>var</i> }	required parameter for set commands; illegal for queries
[<i>var</i>]	optional parameter for both set and query forms

2.4.3 Examples

Each command is provided with a simple example illustrating its usage. In these examples, all data sent by the host computer to the SIM965 are set as *straight teletype font*, while responses received the host computer from the SIM965 are set as *slanted teletype font*.

The usage examples vary with respect to set/query, optional parameters, and token formats. These examples are not exhaustive, but are intended to provide a convenient starting point for user programming.

2.4.4 Filter Commands

FREQ(?) {f}	<p>Filter Frequency</p> <p>Set (query) the filter cutoff frequency in Hz.</p> <p>The valid range for <i>f</i> is 1.00 to 5.00e+5. If <i>f</i> is out of range, it will be ignored and the frequency will remain unchanged.</p> <p><i>f</i> can be given in either decimal form (1270 or 3.14) or exponential notation (1.27E+3 or 3.14E+0), but in all cases the value will be truncated to 3 digits. Digit truncation occurs <i>after</i> range checking, so the value 5.001e+5 is out-of-range and would be rejected.</p> <p><i>Example:</i> FREQ 12345 FREQ? 1.23E+04</p>
<hr/>	
TYPE(?) {z}	<p>Filter type</p> <p>Set (query) the filter type {to z = ((BUTTER 0, BESSEL 1))}.</p> <p><i>Example:</i> TYPE BESSEL TYPE? 1</p>
<hr/>	
PASS(?) {z}	<p>Filter pass band</p> <p>Set (query) the filter pass band {to z = ((LOWPASS 0, HIGHPASS 1))}.</p> <p><i>Example:</i> PASS? LOWPASS</p>
<hr/>	
SLPE(?) {i}	<p>Filter slope</p> <p>Set (query) the filter stop band rolloff rate {to i = (12, 24, 36, 48 dB/octave)}.</p> <p><i>Example:</i> SLPE 24 SLPE? 24</p>
<hr/>	
COUP(?) {z}	<p>Input coupling</p> <p>Set (query) the SIM965 input coupling {to z = (DC 0, AC 1)}.</p> <p><i>Example:</i> COUP 1 COUP? AC</p>

2.4.5 Serial Communication Commands

PARI(?) {z}	Parity Set (query) parity {to z = (NONE 0, ODD 1, EVEN 2, MARK 3, SPACE 4)}. After power-on, modules default to PARI NONE. <i>Example:</i> PARI EVEN
-------------	---

2.4.6 Status Commands

The Status commands query and configure registers associated with status reporting of the SIM965.

*STB? [i]	Status Byte Reads the Status Byte register [bit <i>i</i>]. Execution of the *STB? query (without the optional bit <i>i</i>) always causes the –STATUS signal to be deasserted. Note that *STB? <i>i</i> will <i>not</i> clear –STATUS, even if bit <i>i</i> is the only bit presently causing the –STATUS signal. See also the PSTA command. <i>Example:</i> *STB? 16
*SRE(?) [i,] [j]	Service Request Enable Set (query) the Service Request Enable register [bit <i>i</i>] [to <i>j</i>]. <i>Example:</i> *SRE 0,1
*CLS	Clear Status *CLS immediately clears the ESR and CESR registers, and the OVLD bit in the Status Byte. <i>Example:</i> *CLS
*ESR? [i]	Standard Event Status Reads the Standard Event Status Register [bit <i>i</i>]. Upon executing *ESR?, the returned bit(s) of the ESR register are cleared. <i>Example:</i> *ESR? 64

*ESE(?) [i,] {j}	<p>Standard Event Status Enable</p> <p>Set (query) the Standard Event Status Enable Register [bit <i>i</i>] {to <i>j</i>}.</p> <p><i>Example:</i> *ESE 6,1 ESE? 64</p>
<hr/>	
CESR? [i]	<p>Comm Error Status</p> <p>Query Comm Error Status Register [for bit <i>i</i>].</p> <p>Upon executing a CESR? query, the returned bit(s) of the CESR register are cleared.</p> <p><i>Example:</i> CESR? 0</p>
<hr/>	
CESE(?) [i,]{j}	<p>Comm Error Status Enable</p> <p>Set (query) Comm Error Status Enable Register [for bit <i>i</i>] {to <i>j</i>}</p> <p><i>Example:</i> CESE? 0</p>
<hr/>	
OVLD?	<p>Overload</p> <p>Query the current overload condition. The SIM965 responds with 1 during a signal overload, and 0 at all other times.</p> <p>OVLD? always returns the <i>realtime</i> value of overload, independent of the value of the OVLD bit in the Status Byte.</p> <p><i>Example:</i> OVLD? 0</p>
<hr/>	
PSTA(?) {z}	<p>Pulse –STATUS Mode</p> <p>Set (query) the Pulse –STATUS Mode {to z=(OFF 0, ON 1)}.</p> <p>When PSTA ON is set, any new service request will only <i>pulse</i> the –STATUS signal low (for a minimum of 1 μs). The default behavior is to latch –STATUS low until a *STB? query is received.</p> <p>At power-on, PSTA is set to OFF.</p> <p><i>Example:</i> PSTA? OFF</p>

2.4.7 Interface Commands

Interface commands provide generic control over the interface between the SIM965 and the host computer.

*RST

Reset

Reset the SIM965 to default configuration.

The following commands are internally executed upon *RST:

- `FREQ 1.00E+3`
- `TYPE BUTTER`
- `PASS LOWPASS`
- `SLPE 12`
- `COUP DC`
- `AWAK OFF`
- `TOKN OFF`

Example: *RST

*IDN?

Identify

Read the device identification string.

The identification string is formatted as:

`Stanford_Research_Systems,SIM965,s/n*****,ver#.#`

where `*****` is the 6-digit serial number, and `#.#` is the firmware revision level.

Example: *IDN?

`Stanford_Research_Systems,SIM965,s/n003075,ver3.0`

*OPC(?)

Operation Complete

Operation Complete. Sets the OPC flag in the ESR register.

The query form *OPC? writes a 1 in the output queue when complete, but does not affect the ESR register.

Example: *OPC

CONS(?) {z} Console Mode

Set (query) the Console mode {to z=(**OFF 0**, **ON 1**)}.

CONS causes each character received at the Input Buffer to be copied to the Output Queue.

At power-on and Device-Clear, CONS is set to OFF.

Example: CONS?
0

AWAK(?) {z} Awake mode

Set (query) the SIM965 keep-awake mode {to z = (**OFF 0**, **ON 1**)}.

Ordinarily, the clock oscillator for the SIM965 microcontroller is held in a stopped state, and only enabled during processing of events (Section 1.4). Setting **AWAK ON** forces the clock to stay running, and is useful only for diagnostic purposes.

Example: AWAK ON

LEXE? Execution Error

Query the last execution error code. A query of LEXE? always clears the error code, so a subsequent LEXE? will return 0. Valid codes are:

Value	Definition
0	No execution error since last LEXE?
1	Illegal value
2	Wrong token
3	Invalid bit
16	Invalid parameter
17	Missing parameter
18	No change

Example: *STB? 12; LEXE?; LEXE?
3
0 The error (3, "Invalid bit,") is because *STB? only allows bit-specific queries of 0-7. The second read of LEXE? returns 0.

LCME?

Command Error

Query the last command error code. A query of LCME? always clears the error code, so a subsequent LCME? will return 0. Valid codes are:

Value	Definition
0	No execution error since last LCME?
1	Illegal command
2	Undefined command
3	Illegal query
4	Illegal set
5	Missing parameter(s)
6	Extra parameter(s)
7	Null parameter(s)
8	Parameter buffer overflow
9	Bad floating-point
10	Bad integer
11	Bad integer token
12	Bad token value
13	Bad hex block
14	Unknown token

Example: *IDN
LCME?
4 The error (4, "Illegal set") is due to the missing "?".

LBTN?

Button

Query the last button-press code. A query of LBTN? always clears the button code, so a subsequent LBTN? will return 0. Valid codes are:

Value	Definition
0	no button pressed since last LBTN?
1	[Freq. ▲]
2	[Type]
3	[Freq. ▼]
4	[Slope]
5	[Filter]
6	[Coupling]

Example: LBTN?
1

TOKN(?) {z} Token Mode

Set (query) the Token Query mode {to z=(**OFF 0**, **ON 1**)}.

If TOKN ON is set, then queries to the SIM module that return tokens will return the text keyword; otherwise they return the decimal integer value.

Thus, the only possible responses to the TOKN? query are ON and 0.

On reset, TOKN is set to OFF.

Example: TOKN OFF

TERM(?) {z} Response Termination

Set (query) the <term> sequence {to z=(**NONE 0**, **CR 1**, **LF 2**, **CRLF 3**, **LFCR 4**)}. The <term> sequence is appended to all query responses sent by the module, and is constructed of ASCII character(s) 13 (carriage return) and 10 (line feed). The token mnemonic gives the sequence of characters.

At power-on, TERM is set to CRLF.

Example: TERM?
3

2.5 Status Model

The SIM965 status registers follow the hierarchical IEEE-488.2 format. A block diagram of the status register array is given in Figure 2.1.

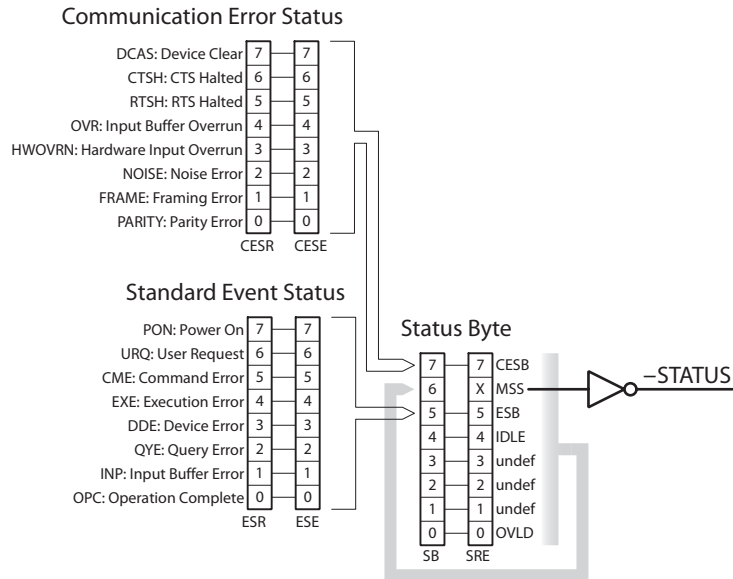


Figure 2.1: Status Register Model for the SIM965.

There are two categories of registers in the SIM965 status model:

Event Registers : These read-only registers record the occurrence of defined events. When the event occurs, the corresponding bit is set to 1. Upon querying an event register, any set bits within it are cleared. These are sometimes known as “sticky bits,” since once set, a bit can only be cleared by reading its value. Event register names end with SR.

Enable Registers : These read/write registers define a bitwise mask for their corresponding event register. If any bit position is set in an event register while the same bit position is also set in the enable register, then the corresponding summary bit message is set. Enable register names end with SE.

2.5.1 Status Byte (SB)

The Status Byte is the top-level summary of the SIM965 status model. When masked by the Service Request Enable register, a bit set in the Status Byte causes the -STATUS signal to be asserted on the rear-panel SIM interface connector.

Typically, -STATUS remains asserted (low) until a *STB? query is received, at which time -STATUS is deasserted (raised)¹. After clearing the -STATUS signal, it will only be re-asserted in response to a *new* status-generating condition.

Weight	Bit	Flag
1	0	OVL D
2	1	undef (0)
4	2	undef (0)
8	3	undef (0)
16	4	IDLE
32	5	ESB
64	6	MSS
128	7	CESB

OVL D : Overload Status. Indicates that an overload has occurred.

IDLE : Indicates that the Input Buffer is empty and the command parser is idle. Can be used to help synchronize SIM965 query responses.

ESB : Event Status Bit. Indicates whether one or more of the enabled events in the Standard Event Status Register is true.

MSS : Master Summary Status. Indicates whether one or more of the enabled status messages in the Status Byte register is true. Note that while -STATUS is released by the *STB? query, MSS is only cleared when the underlying enabled bit message(s) are cleared.

CESB : Communication Error Summary Bit. Indicates whether one or more of the enabled flags in the Communication Error Status Register has become true.

The OVL D bit is a “true” event status bit, and after being set by an overload, the “1” value persists until read by the *STB? query. After a *STB? query, the OVL D bit is cleared to “0”, and can only be set back to 1 by a *new* overload event.

The remaining bits in the Status Byte are *not* cleared by the *STB? query. These bits are only cleared by reading the underlying event registers, or by clearing the corresponding enable registers.

¹ but see the PSTA command

2.5.2 Service Request Enable (SRE)

Each bit in the SRE corresponds one-to-one with a bit in the SB register, and acts as a bitwise AND of the SB flags to generate the MSS bit in the SB and the -STATUS signal. Bit 6 of the SRE is undefined—setting it has no effect, and reading it always returns 0. This register is set and queried with the `*SRE(?)` command.

This register is cleared at power-on.

2.5.3 Standard Event Status (ESR)

The Standard Event Status register consists of 8 event flags. These event flags are all “sticky bits” that are set by the corresponding event, and cleared only by reading or with the `*CLS` command. Reading a single bit (with the `*ESR? i` query) clears only bit i .

Weight	Bit	Flag
1	0	OPC
2	1	INP
4	2	QYE
8	3	DDE
16	4	EXE
32	5	CME
64	6	URQ
128	7	PON

OPC : Operation Complete. Set by the `*OPC` command.

INP : Input Buffer Error. Indicates data has been discarded from the Input Buffer.

QYE : Query Error. Indicates data in the Output Queue has been lost.

DDE : Device Dependent Error. Not implemented in the SIM965.

EXE : Execution Error. Indicates an error in a command that was successfully parsed. Out-of-range parameters are an example. The error code can be queried with `LEXE?`.

CME : Command Error. Indicates a parser-detected error. The error code can be queried with `LCME?`.

URQ : User Request. Indicates a front-panel button was pressed.

PON : Power On. Indicates that an off-to-on transition has occurred

2.5.4 Standard Event Status Enable (ESE)

The ESE acts as a bitwise AND with the ESR register to produce the single bit ESB message in the Status Byte Register (SB). It can be set and queried with the `*ESE(?)` command.

This register is cleared at power-on.

2.5.5 Communication Error Status (CESR)

The Communication Error Status register consists of 8 event flags; each of which is set by the corresponding event, and cleared only by reading or with the *CLS command. Reading a single bit (with the CESR? *i* query) clears only bit *i*.

Weight	Bit	Flag
1	0	PARITY
2	1	FRAME
4	2	NOISE
8	3	HWOVRN
16	4	OVR
32	5	RTSH
64	6	CTSH
128	7	DCAS

PARITY : Parity Error. Set by serial parity mismatch on incoming data byte.

FRAME : Framing Error. Set when an incoming serial data byte is missing the STOP bit.

NOISE : Noise Error. Set when an incoming serial data byte does not present a steady logic level during each asynchronous bit-period window.

HWOVRN : Hardware Overrun. Set when an incoming serial data byte is lost due to internal processor latency. Causes the Input Buffer to be flushed, and resets the command parser.

OVR : Input Buffer Overrun. Set when the Input Buffer is overrun by incoming data. Causes the Input Buffer to be flushed, and resets the command parser.

RTSH : RTS Holdoff Event. Not implemented in the SIM965.

CTSH : CTS Holdoff Event. Not implemented in the SIM965.

DCAS : Device Clear. Indicates the SIM965 received the Device Clear signal (an RS-232 <break>). Clears the Input Buffer and Output Queue, and resets the command parser.

2.5.6 Communication Error Status Enable (CESE)

The CESE acts as a bitwise AND with the CESR register to produce the single bit CESB message in the Status Byte Register (SB). It can be set and queried with the CESE(?) command.

This register is cleared at power-on.

3 Parts Lists and Schematics

This chapter presents a brief description of the SIM965 circuit design. A complete parts list and circuit schematics are included.

In This Chapter

3.1	Circuit Descriptions	3-2
3.1.1	Grounding	3-2
3.1.2	Microcontroller	3-2
3.1.3	Front Panel Display	3-3
3.1.4	Input Coupling and Protection	3-3
3.1.5	Input Overload Detection	3-3
3.1.6	Filter Design	3-3
3.1.7	Output Circuitry	3-4
3.2	Parts Lists	3-4
3.2.1	Digital Board & Front Panel	3-5
3.2.2	Analog Board	3-6
3.3	Schematic Diagrams	3-6

3.1 Circuit Descriptions

The SIM965 consists of three separate printed circuit boards: the digital board, the front-panel board, and the analog board. The digital board is directly beneath the left-hand cover (as viewed from the front of the module).

Pages 1–3 of the schematics correspond to the digital and front-panel boards. Pages 4–9 are the analog board.

3.1.1 Grounding

Internally in the SIM965, JP102-pin 1 (SIGNAL_GND) and JP102-pin 8 (PS_RTN), are tied together on the PCB ground plane. JP102-pin 9 (CHASSIS_GND) is tied to the module chassis through the mounting screws of JP102. CHASSIS_GND is connected to the PCB ground plane through a short wire permanently installed in J101 at the factory.

3.1.2 Microcontroller

The SIM965 is controlled by microcontroller U104. The controller is clocked at 5 MHz.

A critical aspect of the design is the clock-stop circuitry implemented by U101 and U102. A simple RC-oscillator is enabled or disabled at pin 1 of U102, which is driven by synchronizing flip-flop U101B to ensure that no “runt” clock pulses are produced that would violate U104’s minimum clock periods. Four separate clock-starting signals are combined by U106:

- Power-on reset
- Overload (any)
- Incoming serial data
- Front-panel button press

The fast start-time of the RC-oscillator ensures that incoming serial data will be correctly decoded by the microcontroller’s UART, even when the clock is started by the serial start bit of the incoming data. When the microcontroller has completed all pending activity, it drives the STOP signal high (pin 8 of U104), effectively halting its own processor clock. In this way, the SIM965 guarantees no digital clock artifacts can be generated during quiescent operation.

3.1.3 Front Panel Display

The seven segment displays and LED lamps on the front panel are powered by U204–U208, a daisy-chain of 5 74HC595 serial input shift registers. The currents to the lamps are limited by resistor networks in series with the displays and LEDs. The displays and LEDs are all statically driven (i.e., *not* multiplexed).

3.1.4 Input Coupling and Protection

A relay switch, K301, in parallel with a capacitor C301, allows for DC or AC input coupling. The input signal is buffered by U306. Following that, the signal passes through a soft-limiting clamp circuit to ensure that the input to the filter circuitry does not exceed the ± 10 V input range. The limiter consists of resistor R306 and a diode/transistor arrangement to actively sink the resistors output current when the input signal goes out of the desired input range. This clamps the resistors output voltage. This is followed by an additional follower U307 to buffer the signal input to the filter circuitry.

3.1.5 Input Overload Detection

Input overload is detected via dual comparator U302 with a wire-or'd output arrangement that is level shifted using Q304 and Q305. This signal is input to the controller.

3.1.6 Filter Design

The filter design consists of four cascaded biquad state-variable filters, using a combination of analog switches to control the analog signal path. The state-variable design allows for independent control of filter gain, cutoff frequency, and Q-factor. These are shown on pages 5–8 of the schematics (one biquad per page). Component references below are for the first biquad, on page 5 (reference numbers in the 200's). Equivalent corresponding components for the subsequent stages are numbered in the 300's, 400's, and 500's, respectively.

Each biquad filter section consists of four multiplying DACs in series, with multiple feedback paths. The first DAC in the series, U207A controls the filter section gain. The second DAC, U207B controls filter Q-factor. The remaining two DACs, U209A & U209B, are configured as integrators using feedback capacitor banks. Analog multiplexers U203, U204, U205, and U208 select the appropriate capacitor combinations to achieve a desired cutoff frequency range. The integrator DACs allow for trimming of the cutoff frequency.

An attractive feature of the state-variable filter architecture is that it provides output “taps” for low pass, high pass, and band pass filters.

These taps are fed to three of the inputs to a 4-pole analog switch. The remaining pole of the switch is connected to the filter input. In this way, the output of the overall filter section circuit can be selected to be either of the three filter types, or the fourth option, to bypass the filter. When the filter is bypassed, a second switch is used to ground the filter section input. This switching arrangement allows each filter section to be switched in or out of the complete cascaded filter to achieve the desired stop band rolloff slope and passband type.

3.1.7 Output Circuitry

The output of the final filter stage is routed back to the digital board, which contains the output circuitry (see page 3). U303 and U304 form a composite amplifier for the output driver. This arrangement provides the driving capability of the BUF634 without suffering its large input offset voltage, since the output of U303 is servoed to the noninverting input of U304 via the feedback resistor R315.

U305 is a photo-MOS switch that remains off during power-up until the $\pm 15\text{ V}$ rails reach about $\pm 13\text{ V}$. By then, analog circuitry have settled, permitting the output signal to be connected to the output BNCs without large transients. Until switch U305 closes, the SIM965 output is referenced to ground via R320 (100 k).

3.2 Parts Lists

The parts list for the analog board is separate from the digital & front-panel boards.

3.2.1 Digital Board & Front Panel

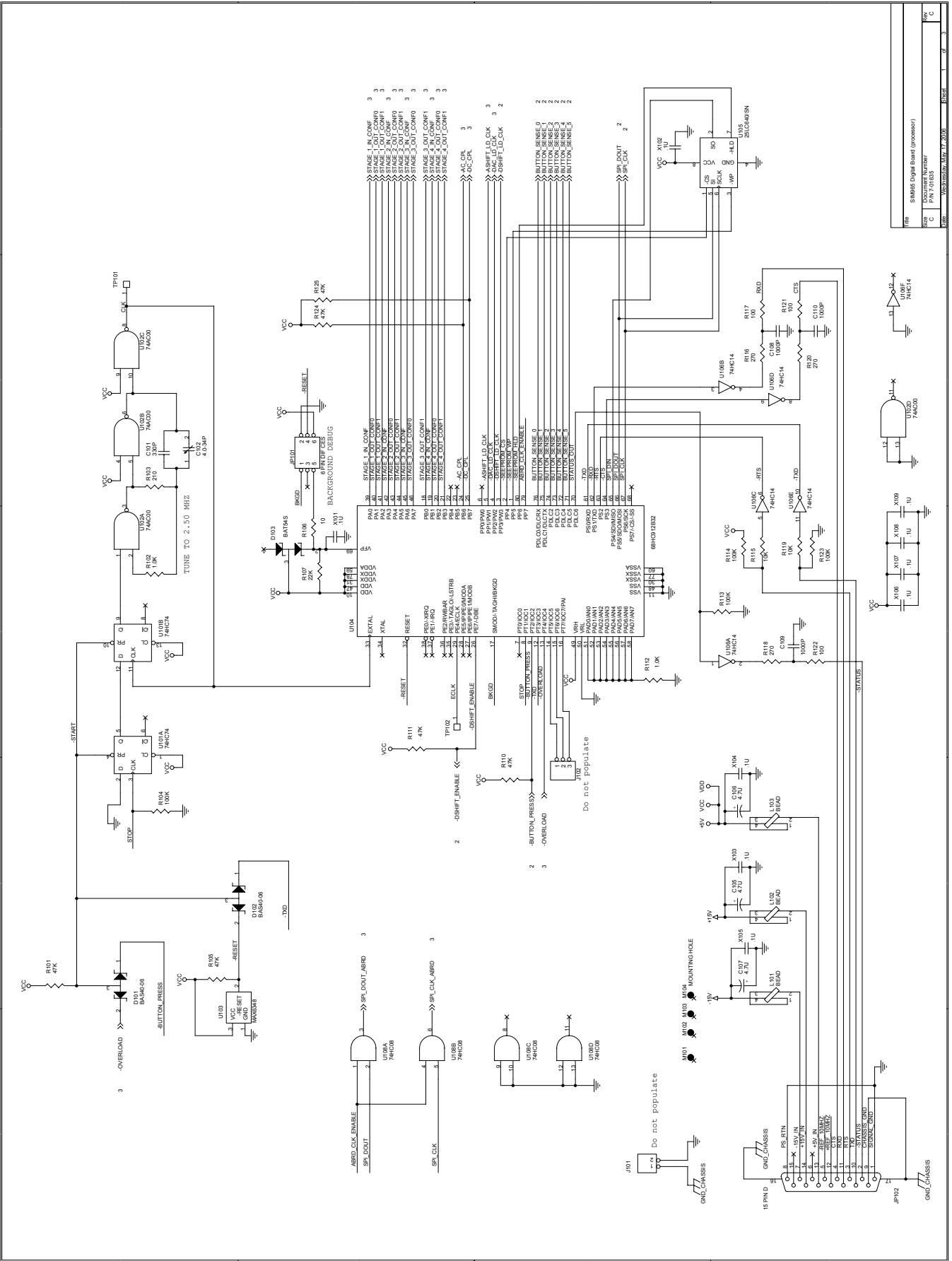
Reference	SRS P/N	Value	Reference	SRS P/N	Value
C101	5-00381	330P	R107	4-01511	22K
C102	5-00345	4.0-34P	R115,R119	4-01503	10K
C105,C106,C107,C302,C305	5-00102	4.7U	R116,R118,R120	4-01465	270
C108,C109,C110	5-00387	1000P	R117,R121,R122	4-01455	100
C201,C202,C203,C204,C205	5-00299	.1U	R301,R302,R311,R313,R316	4-01117	1.00K
C301	5-00542	1.0U	R303,R312	4-01059	249
C306	5-00367	22P	R304,R310	4-01242	20.0K
C307,C308	5-00318	2.2U/T35	R305	4-01067	301
D101,D102	3-00901	BAS40-06	R306	4-01184	4.99K
D103	3-00945	BAT54S	R307,R309	4-01280	49.9K
D201	3-00425	RED	R308	4-01405	1.00M
D202,D203,D204,D205,D206, D207,D208,D209,D210,D211, D212,D213	3-00424	GREEN	R314	4-01021	100
D301,D308	3-01357	MMBZ5230	R315	4-01104	732
D302	3-00783	MMBZ5226	R317	4-01406	0
D303	3-01430	BAS40-05	R320	4-01309	100K
D304	3-00901	BAS40-06	R325,R326	4-00913	49.9 FP
D305,D306	3-00230	1N5240A	RN201,RN202,RN203, RN204,RN205,RN206	4-00407	2.7K 1206 MINI
D307	3-00783	MMBZ5226	RN207,RN208,RN209,RN210	4-00442	1.2K 1206 MINI
D309,D310	3-01487	MMBZ5242	S201,S202,S203,S204,S205, S206	2-00053	B3F-1052
J211	1-00636	FLE-123-01-GDVA	U101	3-00742	74HC74
J212	1-00637	FTSH-123-04-LMT	U102	3-01405	MC74AC00D
J302	1-00618	SSW-113-01-G-D	U103	3-00903	MAX6348UR44
JP101	1-00302	6 PIN DIF CES	U104	3-01379	MC68HC912B32CFU
JP102	1-00367	15 PIN D	U105	3-01390	25LC640/SN
K301	3-00617	DS1E-ML2-DC5V	U106	3-00662	74HC14
L101,L102,L103	6-00174	6611 TYPE 43	U108	3-00663	74HC08
PCB	7-01635	SIM965, DIGITAL	U201,U202,U203	3-00290	HDSP-A101
Q301,Q302	3-00927	MMBT2907ALT1	U204,U205,U206,U207,U208	3-00787	74HC595
Q303	3-00601	MMBT3904LT1	U302	3-00728	LM393
Q304	3-00580	MMBT3906LT1	U303	3-01221	BUF634P
Q305	3-00601	MMBT3904LT1	U304	3-01289	LT1363CS8
Q306	3-00580	MMBT3906LT1	U305	3-01488	AQY221R2S
R101,R105,R110,R111,R124, R125	4-01519	47K	U306,U307	3-01218	AD825AR
R102,R112	4-01479	1.0K	X101,X102,X103,X104,X105, X106,X107,X108,X109,X301, X302,X303,X304,X305,X306, X307,X308,X309,X310	5-00299	.1U
R103	4-01052	210			
R104,R113,R114,R123	4-01527	100K			
R106	4-01431	10			

3.2.2 Analog Board

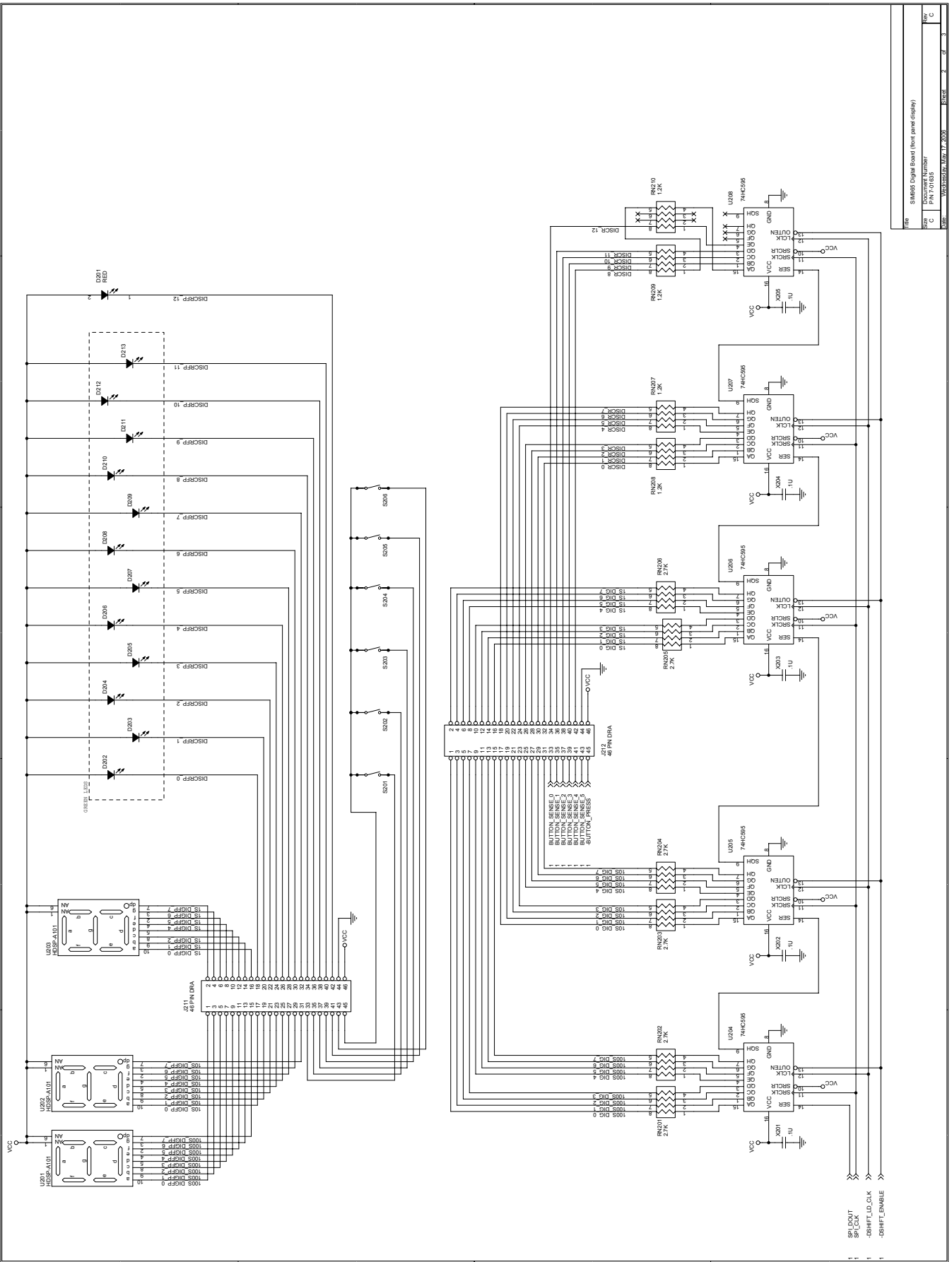
Reference	SRS P/N	Value	Reference	SRS P/N	Value
C203,C204,C303,C304,C403, C404,C503,C504	5-00371	47P	C238,C239,C338,C339,C438, C439,C538,C539	5-00368	27P
C205,C208,C305,C308,C405, C408,C505,C508	5-00616	220P	C240,C340,C440,C540	5-00313	1P
C209,C210,C309,C310,C409, C410,C509,C510	5-00442	.001U	C241,C341,C441,C541	5-00625	1.8-6.0P RED
C211,C212,C311,C312,C411, C412,C511,C512	5-00450	.0047U	C242,C243,C244,C245,C342, C343,C344,C345,C442,C443, C444,C445,C542,C543,C544, C545	5-00318	2.2U/T35
C213,C214,C313,C314,C413, C414,C513,C514	5-00458	.022U	J113	1-00638	TSW-113-15-G-D
C215,C216,C315,C316,C415, C416,C515,C516	5-00570	.1U - PPS	PC1	7-01636	SIM965, ANALOG
C217,C218,C317,C318,C417, C418,C517,C518	5-00617	.47U / 250V	R202R302	4-01208	8.87K
C219,C220,C319,C320,C419, C420,C519,C520	5-00618	2.2U / 250V	R401R405	4-01242	20.0K
C221,C222,C321,C322,C421, C422,C521,C522	5-00574	100P - PPS	R402	4-01204	8.06K
C223,C224,C323,C324,C423, C424,C523,C524	5-00573	470P	R407	4-01280	49.9K
C225,C226,C325,C326,C425, C426,C525,C526	5-00446	.0022U	R502	4-01230	15.0K
C227,C228,C327,C328,C427, C428,C527,C528	5-00454	.01U	R507	4-01251	24.9K
C229,C230,C329,C330,C429, C430,C529,C530	5-00462	.047U	RU507	4-01364	374K
C231,C232,C331,C332,C431, C432,C531,C532	5-00619	.22U / 100V	U201,U202,U301,U302,U401, U402,U501,U502	3-01328	LT1361CS8
C235,C236,C335,C336,C435, C436,C535,C536	5-00615	4.7U / 63V 5%	U203,U204,U206,U208,U303, U304,U306,U308,U403,U404, U406,U408,U503,U504,U506, U508	3-01386	DG408DY
			U205,U305,U405,U505	3-00787	74HC595
			U207,U209,U307,U309,U407, U409,U507,U509	3-01171	AD5415YRU
			U210,U310,U410,U510	3-01367	DG419DY
			U211,U311,U411,U511	3-01369	DG409DY
			X201-X220,X301-X320, X401-X420,X501-X520	5-00299	.1U

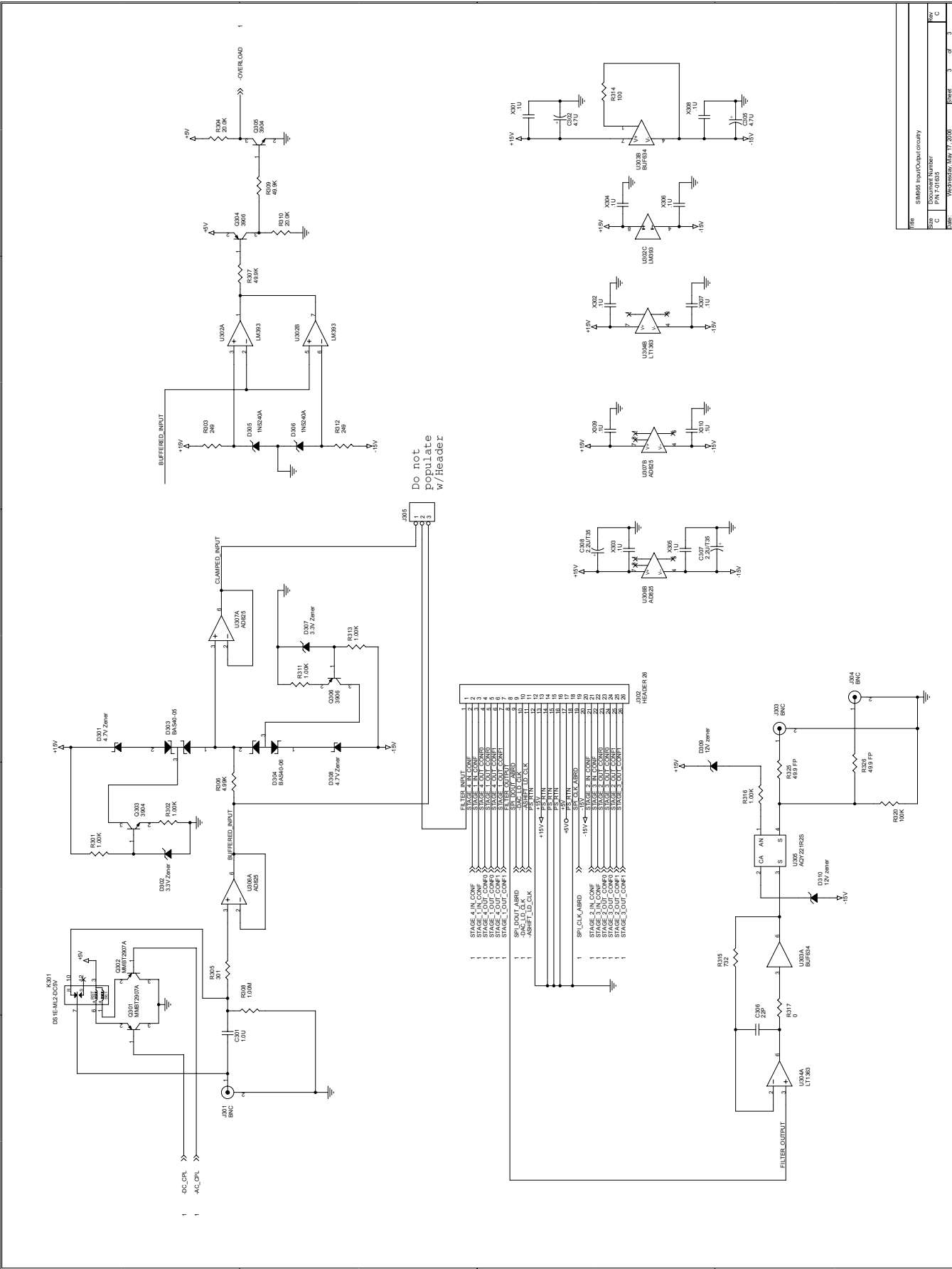
3.3 Schematic Diagrams

Schematic diagrams follow this page.



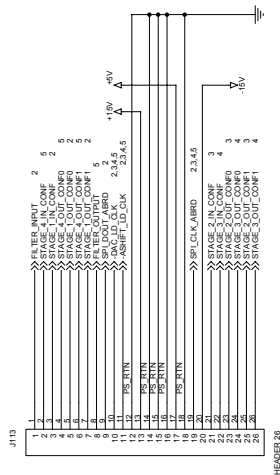
REV	SMB05 Digital Board (processor)
DOC NUMBER	U108F
SIZE	A1 (11x17)
DATE	10/20/2009, Rev. 1.0



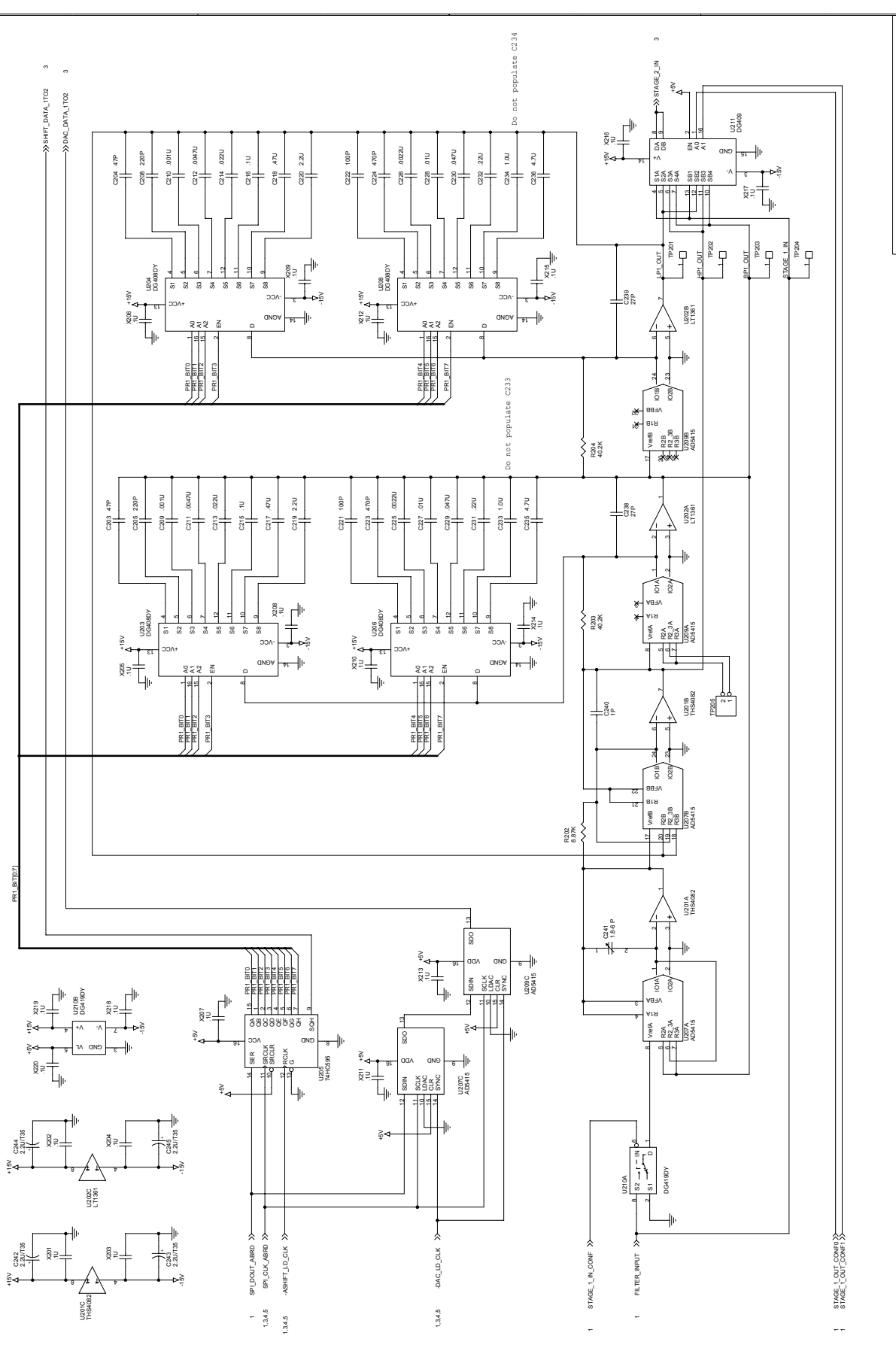


REV	S1665 Input/Output Circuitry
DOC NUMBER	1665
DATE	11/11/04
DESIGNED BY	WJW
CHECKED BY	WJW
DATE	11/11/04
REV	0
DATE	11/11/04

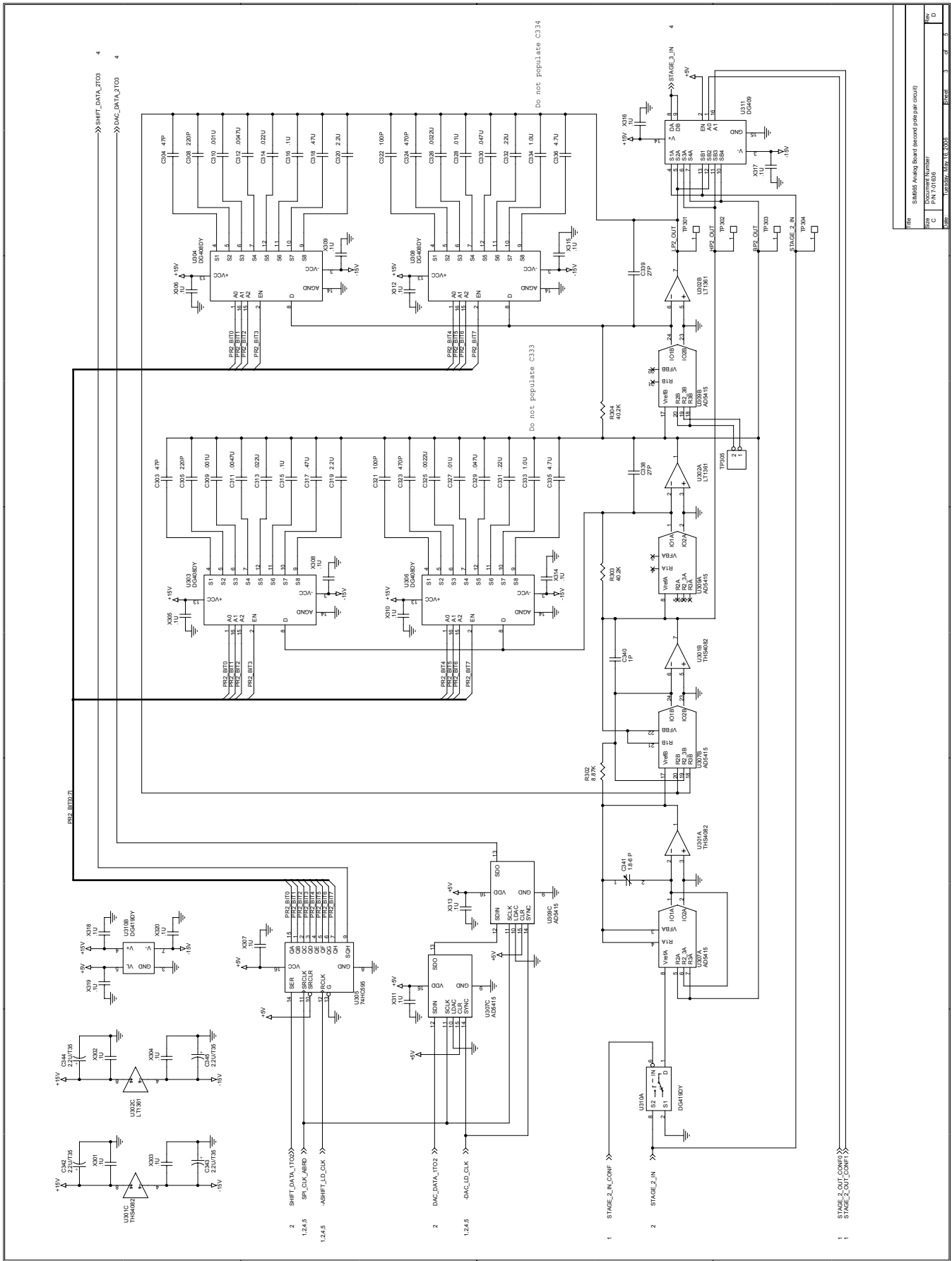
MOUNTING HOLES

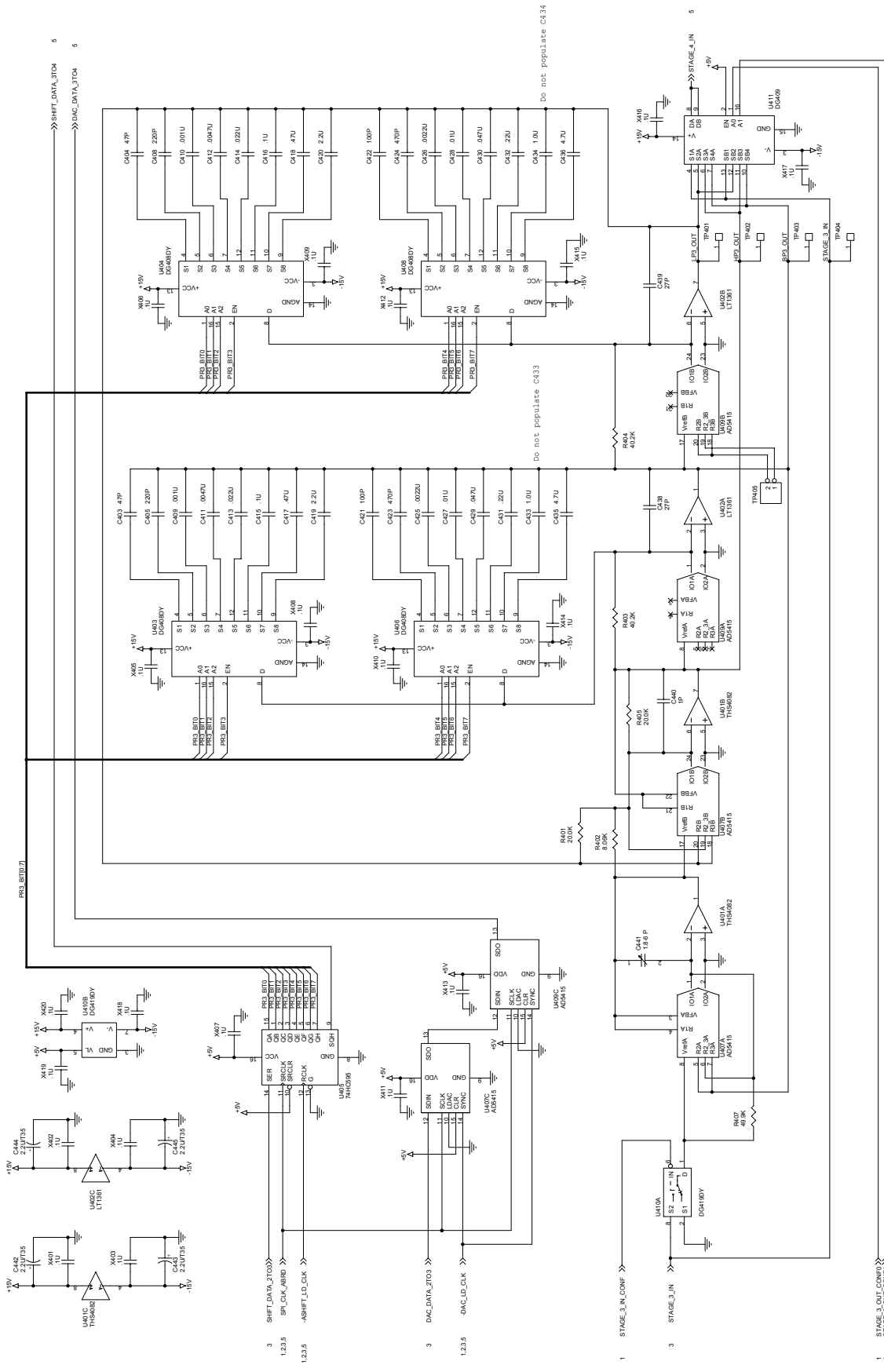



FILE	S1665 Analog Board Connector
DocuName	Pin-1-Board
DocuNo	16
DATE	08/25/08 16:20:08
SHEET	8
TOTAL	8



REV	SWISS Analog Board (8-pairs per circuit)
1	DocuNumber
2	PIN-1/640
3	LIBRARY: 487, 16, 2025





REV	SW665 Analog Board (final power circuit)
DATE	2008.12.16
DOCUMEN NUMBER	
FILE	SW665_ANALOG.PCB

