REVISIONS						
DCN	LTR	DESCRIPTION	DATE	CHANGED	CHECKED	APPROVED
	4	REVISED	090302	S. Pickles		
	-	RELEASED FOR PRODUCTION	090304	P. Jones		
5476	A	REVISED PER DCN	100208	S. Pickles	S.Pickles	D.Wildon
5502	B	REVISED PER DCN	100311	S. Pickles	S.Pickles	S.Pickles

UNCONTROLLED COPY

Contact AMI to verify revision

FILENAME								1	SHT
7665B.DOC								B	REV

UNLESS OTHERWISE SPECIFIED:
ALL DIMENSIONS ARE IN INCHES. ALL CAPACITANCE VALUES ARE IN MICROFARADS. ALL RESISTANCE VALUES ARE IN OHMS.

ANALOG MODULES, INC.

a HEICO company

DRAWN BY	DATE	CHECKED BY	DATE
Pat Jones	090225	Noel Chandler	090917
LEAD APPROVAL	DATE	FINAL APPROVAL	DATE
Steven Pickles	090918	John Harwick	090929

TOLERANCES

DECIMAL
.XX = N/A
.XXX = N/A

TITLE

INTERFACE CONTROL DOCUMENT
MODEL 762 SEED LASER DIODE DRIVER

ANGULAR	SCALE	SIZE	FSCM	SHEET	DRAWING NUMBER	REV
X = N/A	N / A	A	61651	1 OF 35	7665	B

Table of Contents

List of Tables

List of Figures

List of Equations

Scope

Identification

This Interface Control Document (ICD) details the messaging interface between a host system and the 762 Seed Laser Diode Driver (SLDD). The document also describes the device memory map, register descriptions, and digital-to-analog converters.

Software Interface

Bus Data Rate

The bus data rate for asynchronous serial is 9600 bits per second, with 8 data bits, no parity, 1 stop bit, and no flow control. The host defined bus data rate for I^2C synchronous serial should be either 100 kHz or 400 kHz. Review *Figure 1 – I2C Timing Diagram* and *Table 1 – I2C Timing Characteristics* for timing considerations regarding I^2C communication.

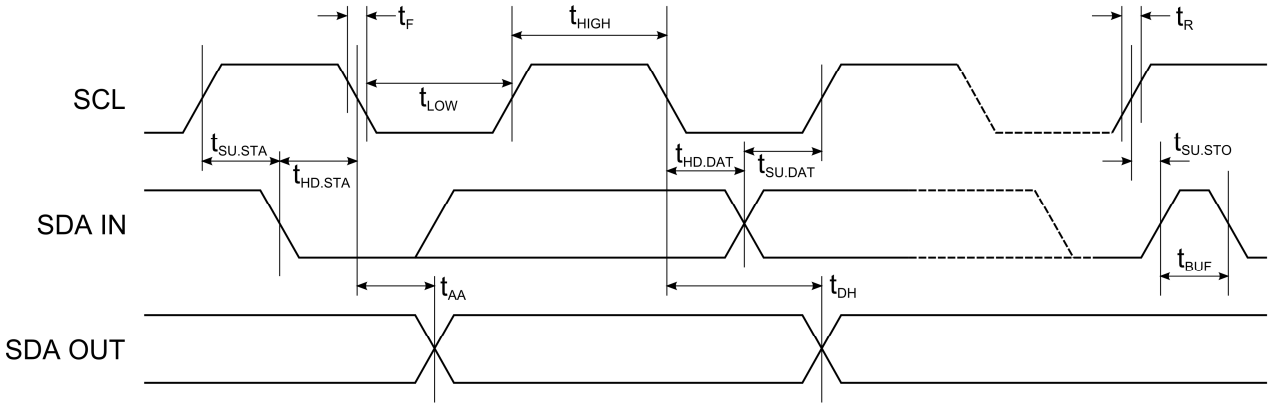

Figure 1 – I^2C Timing Diagram

Parameter	Symbol	Min	Max	Units
Clock Frequency, SCL	f_{SCL}	100	400	kHz
Clock Low to Data Out Valid	t_{AA}	0.1	0.9	µs
Bus Free Time Before New Start	t_{BUF}	1.2		µs
Data Out Hold Time	t_{DH}	50		ns
Input Fall Time	t_F		300	ns
Data in Hold Time	$t_{HD.DAT}$	0		µs
Start Hold Time	$t_{HD.STA}$	0.6		µs
Clock Pulse Width High	t_{HIGH}	0.6		µs
Clock Pulse Width Low	t_{LOW}	1.2		µs
Input Rise Time	t_R		0.3	µs
Data in Setup Time	$t_{SU.DAT}$	100		ns
Start Setup Time	$t_{SU.STA}$	0.6		µs
Stop Setup Time	$t_{SU.STO}$	0.6		µs

Table 1 – I^2C Timing Characteristics

Digital-to-Analog Converters

There are eight digital-to-analog converters (DACs) on the SLDD. Seven of the DACs perform specific functions and the eighth is a test point; see *Table 2 – Description of DACs* for a brief description of what each DAC controls along with the minimum and maximum values. Each of the values listed in the table below are derived from the transfer functions given throughout this section. All of the converters are software selectable and software controllable within the minimum and maximum quantization steps described in the first two rows *Table 5* on page 7. Keep in mind DAC 4 is the only DAC that cannot use all 4096 digital quantization steps.

DAC	Description	Minimum Value	Maximum Value	Units
1	Laser Fire Threshold	0	12.03*	mA
2	TEC Maximum Current	0	3	A
3	TEC Maximum Voltage	0	4.2	V
4	TEC Set Point Temperature	-3.1	82	ºC
5	Laser Diode Current	0	2.5	A
6	Current Limit	0	2.61	A
7	Laser Diode Bias Current	0	681	mA
8	Test Point	0	5	V

Table 2 – Description of DACs

When digitally controlling the parameters listed above and the DAC is enabled, the analog potentiometer settings are overridden. If a DAC is disabled, the analog control of the parameter returns and the potentiometer dictates the set value. To calculate the value in native units of any parameter described above, refer to the equations listed in the following eight sections. The maximum value of DAC 1 – Laser Fire Threshold in the above table contains an asterisk (*) because this value is dependent upon the monitor gain setting.

DAC 1 – Laser Fire Threshold

The laser fire threshold is dependent upon the gain setting. As a result, the monitor gain multiplier value, *G*, in *Equation 1* and *Equation 2* is shown in *Table 3 – Monitor Gain Multiplier* for each gain setting in ascending order from lowest gain to highest gain. The DAC controlling the laser fire threshold has a voltage range between 0 V and +1.25 V. This means each user adjustable quantization step from 0 to 4095 is approximately +0.3052 mV/step.

Monitor Gain	Gain Multiplier, *G*
00	3.13
01	9.83
10	31.55
11	62.44

Table 3 – Monitor Gain Multiplier

$$\text{quantization step number to milliamperes} = \left(\frac{\text{step \#}}{1}\right)\left(\frac{1.25\ \text{V}}{4096\ \text{steps}}\right)\left(\frac{1\ \text{mA}}{0.0332\ \text{V} \cdot G}\right)$$

Equation 1 – Laser Fire Threshold – Quantization Step Number to Milliamperes

$$\text{milliamperes to quantization step number} = \left(\frac{\text{current in mA}}{1}\right)\left(\frac{4096\ \text{steps}}{1.25\ \text{V}}\right)\left(\frac{0.0332\ \text{V} \cdot G}{1\ \text{mA}}\right)$$

Equation 2 – Laser Fire Threshold – Milliamperes to Quantization Step Number

DAC 2 – TEC Maximum Current

The DAC controlling the TEC maximum current has a voltage range between 0 V and +600 mV. This means each user adjustable quantization step from 0 to 4095 is approximately 0.1465 mV/step.

$$\text{quantization step number to amperes} = \left(\frac{\text{step \#}}{1}\right)\left(\frac{600\text{ mV}}{4096\text{ steps}}\right)\left(\frac{\text{A}}{200\text{ mV}}\right)$$

Equation 3 – TEC Maximum Current – Quantization Step Number to Amperes

$$\text{amperes to quantization step number} = \left(\frac{\text{current in A}}{1}\right)\left(\frac{4096\text{ steps}}{600\text{ mV}}\right)\left(\frac{200\text{ mV}}{\text{A}}\right)$$

Equation 4 – TEC Maximum Current – Amperes to Quantization Step Number

DAC 3 – TEC Maximum Voltage

The DAC controlling the TEC maximum voltage has a voltage range between 0 V and +1.05 V. This means each user adjustable quantization step from 0 to 4095 is approximately +0.2563 mV/step.

$$\text{quantization step number to volts} = \left(\frac{\text{step \#}}{1}\right)\left(\frac{1.05\text{ V}}{4096\text{ steps}}\right)\left(\frac{4\text{ V}}{\text{V}}\right)$$

Equation 5 – TEC Maximum Voltage – Quantization Step Number to Volts

$$\text{volts to quantization step number} = \left(\frac{\text{voltage in V}}{1}\right)\left(\frac{4096\text{ steps}}{1.05\text{ V}}\right)\left(\frac{\text{V}}{4\text{ V}}\right)$$

Equation 6 – TEC Maximum Voltage – Volts to Quantization Step Number

DAC 4 – TEC Set Point Temperature

The DAC controlling the TEC set point temperature has a voltage range between 0 V and +1.5 V. This means each user adjustable quantization step from 0 to 4095 is approximately +0.3662 mV/step. Calculating the TEC set point temperature is made slightly more difficult because the thermistor resistance does not vary linearly over the temperature range. While each laser manufacturer may use a different thermistor with a different resistance to temperature table, the data provided in *Table 4 – Thermistor Resistance Values by Temperature* are the thermistor resistance values between -9º C and +90º C for the Bookham LC96A1060 laser diode.

Temp.	Resistance	Temp.	Resistance	Temp.	Resistance	Temp.	Resistance
-9º C	52,380.0 Ω	16º C	15,001.2 Ω	41º C	5,115.6 Ω	66º C	2,010.8 Ω
-8º C	49,633.0 Ω	17º C	14,324.6 Ω	42º C	4,915.5 Ω	67º C	1,942.1 Ω
-7º C	47,047.0 Ω	18º C	13,682.6 Ω	43º C	4,724.3 Ω	68º C	1,876.0 Ω
-6º C	44,610.0 Ω	19º C	13,052.8 Ω	44º C	4,541.6 Ω	69º C	1,812.6 Ω
-5º C	42,315.0 Ω	20º C	12,493.7 Ω	45º C	4,366.9 Ω	70º C	1,751.6 Ω
-4º C	40,150.0 Ω	21º C	11,943.3 Ω	46º C	4,199.9 Ω	71º C	1,693.0 Ω
-3º C	38,109.0 Ω	22º C	11,420.0 Ω	47º C	4,040.1 Ω	72º C	1,636.6 Ω
-2º C	36,183.0 Ω	23º C	10,922.7 Ω	48º C	3,887.2 Ω	73º C	1,582.4 Ω
-1º C	34,366.0 Ω	24º C	10,449.9 Ω	49º C	3,741.1 Ω	74º C	1,530.3 Ω
0º C	32,650.8 Ω	25º C	10,000.0 Ω	50º C	3,601.0 Ω	75º C	1,480.1 Ω
1º C	31,030.4 Ω	26º C	9,572.0 Ω	51º C	3,466.9 Ω	76º C	1,431.9 Ω
2º C	29,500.1 Ω	27º C	9,164.7 Ω	52º C	3,338.6 Ω	77º C	1,385.4 Ω
3º C	28,054.2 Ω	28º C	8,777.0 Ω	53º C	3,215.6 Ω	78º C	1,340.7 Ω
4º C	26,687.6 Ω	29º C	8,407.7 Ω	54º C	3,097.9 Ω	79º C	1,297.6 Ω
5º C	25,395.5 Ω	30º C	8,056.0 Ω	55º C	2,985.1 Ω	80º C	1,256.2 Ω
6º C	24,172.7 Ω	31º C	7,720.9 Ω	56º C	2,876.9 Ω	81º C	1,216.2 Ω
7º C	23,016.0 Ω	32º C	7,401.7 Ω	57º C	2,773.2 Ω	82º C	1,177.8 Ω
8º C	21,921.7 Ω	33º C	7,097.2 Ω	58º C	2,673.9 Ω	83º C	1,140.7 Ω
9º C	20,885.2 Ω	34º C	6,807.0 Ω	59º C	2,578.5 Ω	84º C	1,105.0 Ω
10º C	19,903.5 Ω	35º C	6,530.1 Ω	60º C	2,487.1 Ω	85º C	1,070.6 Ω
11º C	18,973.6 Ω	36º C	6,266.1 Ω	61º C	2,399.4 Ω	86º C	1,037.4 Ω
12º C	18,092.6 Ω	37º C	6,014.2 Ω	62º C	2,315.2 Ω	87º C	1,005.4 Ω
13º C	17,257.4 Ω	38º C	5,773.7 Ω	63º C	2,234.7 Ω	88º C	974.6 Ω
14º C	16,465.1 Ω	39º C	5,544.1 Ω	64º C	2,156.7 Ω	89º C	944.8 Ω
15º C	15,714.0 Ω	40º C	5,324.9 Ω	65º C	2,082.3 Ω	90º C	916.1 Ω

Table 4 – Thermistor Resistance Values by Temperature

For a temperature in between two consecutive whole number temperature values listed in *Table 4*, interpolate linearly for a close approximation of the thermistor resistance. For example, to determine the resistance at 25.7º C, assume that each 0.1º C increment above 25º C is equal to { 10,000 - [7 · ((10,000 Ω - 9,572 Ω) / 10)] }, or 9,700.4 Ω. Since the temperature correlates directly to the temperature and there is no convenient polynomial fit, the temperature is not part of either equation listed below. Use *Equation 7* and *Equation 8* along with *Table 4* to convert the degree Celsius temperature to a resistance when converting a quantization step number to a temperature or a temperature to a quantization step number. Note, there is an inverse relationship between the temperature/resistance and the quantization step number.

$$\text{quantization step number to Ohms} = \frac{\left(\frac{10000\ \Omega}{1}\right)\left(\frac{\text{step \#}}{1}\right)\left(\frac{1.5\ \text{V}}{4096\ \text{steps}}\right)}{1.5\ \text{V} - \left[\left(\frac{\text{step \#}}{1}\right)\left(\frac{1.5\ \text{V}}{4096\ \text{steps}}\right)\right]}$$

Equation 7 – TEC Set Point Temperature – Quantization Step Number to Ohms

$$\text{Ohms to quantization step number} = \frac{(4096\ \text{steps})(1.5\ \text{V})}{\left[\frac{(10000\ \Omega)(1.5\ \text{V})}{\text{resistance in Ohms}}\right] + 1.5\ \text{V}}$$

Equation 8 – TEC Set Point Temperature – Ohms to Quantization Step Number

DAC 5 – Laser Diode Current

The 762 can set the amount of current intended to move through the laser diode during a laser pulse. The DAC controlling the laser diode current has a voltage range between 0 V and +1.25 V/step. This means each user adjustable quantization step from 0 to 4095 is approximately +0.3052 mV.

$$\text{quantization step number to amperes} = \left(\frac{\text{step \#}}{1}\right)\left(\frac{1.25\ \text{V}}{4096\ \text{steps}}\right)\left(\frac{2\ \text{A}}{\text{V}}\right)$$

Equation 9 – Laser Diode Current – Quantization Step Number to Amperes

$$\text{amperes to quantization step number} = \left(\frac{\text{current in A}}{1}\right)\left(\frac{4096\ \text{steps}}{1.25\ \text{V}}\right)\left(\frac{\text{V}}{2\ \text{A}}\right)$$

Equation 10 – Laser Diode Current – Amperes to Quantization Step Number

DAC 6 – Current Limit

To protect the laser diode, the current limit sets the maximum amount of current that is permitted to move through the laser diode. Should the laser diode current exceed the current limit set by the user, the 762 issues an error and shuts down to prevent damage to the laser diode. The DAC controlling the current limit has a voltage range between 0 V and +362 mV. This means each user adjustable quantization step from 0 to 4095 is approximately +0.0884 mV/step.

$$\text{quantization step number to amperes} = \left(\frac{\text{step \#}}{1}\right)\left(\frac{362\ \text{mV}}{4096\ \text{steps}}\right)\left(\frac{\text{A}}{125\ \text{mV}}\right)$$

Equation 11 – Current Limit – Quantization Step Number to Amperes

$$\text{amperes to quantization step number} = \left(\frac{\text{current in A}}{1}\right)\left(\frac{4096\ \text{steps}}{362\ \text{mV}}\right)\left(\frac{125\ \text{mV}}{\text{A}}\right)$$

Equation 12 – Current Limit – Amperes to Quantization Step Number

DAC 7 – Laser Diode Bias Current (Offset Adjustment)

The DAC controlling the laser diode bias current (offset adjustment) has a voltage range between 0 V DC and +1.25 V. This means each user adjustable quantization step from 0 to 4095 is approximately +0.3052 mV/step.

$$\text{quantization step number to milliamperes} = \left(\frac{-659.7\ \text{mA}}{\text{V}}\right)\left[0.8261\left(\frac{\text{step \#}}{1}\right)\left(\frac{1.25\ \text{V}}{4096\ \text{steps}}\right) - 0.4347\ \text{V}\right]$$

Equation 13 – Laser Diode Bias Current – Quantization Step Number to Milliamperes

$$\text{milliamperes to quantization step number} = \left(\frac{286.8\ \text{mA} - (\text{current in mA})}{545.0\ \text{mA}}\right)\left(\frac{4096\ \text{steps}}{1.25}\right)$$

Equation 14 – Laser Diode Bias Current – Milliamperes to Quantization Step Number

DAC 8 – Test Point

As a test point for measurement, DAC 8 controls a voltage range between 0 V and +4.63 V. This means each user adjustable quantization step from 0 to 4095 is approximately +1.2207 mV/step.

$$\text{quantization step number to volts} = \left(\frac{\text{step \#}}{1}\right)\left(\frac{4.63\ \text{V}}{4096\ \text{steps}}\right)$$

Equation 15 – Test Point – Quantization Step Number to Volts

$$\text{volts to quantization step number} = \left(\frac{\text{voltage in V}}{1}\right)\left(\frac{4096\ \text{steps}}{4.63\ \text{V}}\right)$$

Equation 16 – Test Point –Volts to Quantization Step Number

Memory Map

The SLDD contains four 128-byte memory banks. This allows for four separate configurations to be stored on the SLDD. By default, all banks are identical and contain the configuration settings found in *Table 5*. The system automatically loads memory bank 0 on power up. In order to ensure the SLDD operates properly, memory addresses 0x00 to 0x1F cannot be edited by the host system. Data contained at these addresses are programmed at the factory.

	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x000	DAC 1 Minimum (0x0000)		DAC 2 Minimum (0x0000)		DAC 3 Minimum (0x0000)		DAC 4 Minimum (0x01B0)		DAC 5 Minimum (0x0000)		DAC 6 Minimum (0x0000)		DAC 7 Minimum (0x0000)		DAC 8 Minimum (0x0000)	
0x010	DAC 1 Maximum (0x0FFF)		DAC 2 Maximum (0x0FFF)		DAC 3 Maximum (0x0FFF)		DAC 4 Maximum (0x0CAF)		DAC 5 Maximum (0x0FFF)		DAC 6 Maximum (0x0FFF)		DAC 7 Maximum (0x0FFF)		DAC 8 Maximum (0x0FFF)	
0x020	DAC 1 Value (0x07FF)		DAC 2 Value (0x07FF)		DAC 3 Value (0x07FF)		DAC 4 Value (0x072F)		DAC 5 Value (0x07FF)		DAC 6 Value (0x07FF)		DAC 7 Value (0x068C)		DAC 8 Value (0x07FF)	
0x030	DAC Enable Config (0x0000)		Serial Select Config (0x0000)		Monitor Gain Config (0x0000)		Amp Sync Output Polarity (0x0000)		Trigger Config (0x0008)		Pulse Output Enable Config (0x0004)		I^2C Address (0x0050)		TEC Shutdown Config (0x0000)	
0x040	Trigger / PW Delay (0x00000000)				Reserved	Reserved	Reserved	Trigger / PW Pulsewidth (0x0000000000)					Internal Trigger Period (0x00000000)			
0x050	Amp Sync 1 Delay (0x00000000)				Reserved	Reserved	Reserved	Amp Sync 1 Pulsewidth (0x0000000000)					Reserved	Reserved	Reserved	Reserved
0x060	Amp Sync 2 Delay (0x00000000)				Reserved	Reserved	Reserved	Amp Sync 2 Pulsewidth (0x0000000000)					Reserved	Reserved	Reserved	Reserved
0x070	Serial Number (Unique)				Firmware Version Number				Reserved	Reserved	Enable Select Config (0x0000)		Fingerprint			

Table 5 – SLDD Memory Map

DACxMIN – DAC x Minimum Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	DACxMIN[11..0]											
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 6 – DACxMIN – DAC x Minimum Register

The DACxMIN register sets the minimum value for the digital-to-analog converter for the specified DAC. Only the lower 12 bits are significant as the DAC has 12 bits of resolution. DACs 1 through 3 and 5 through 7 all have default values of 0x000, as described in *Table 6 – DACxMIN – DAC x Minimum Register*, while DAC 4 has a minimum value of 0x1B0. Due to this singularity, the DAC4MIN register is shown in *Table 7 – DAC4MIN – DAC 4 Minimum Register Description*. To ensure the functionality of the 762, all minimum DAC values are factory set and cannot be changed by the user.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	DAC4MIN[11..0]											
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	0

Table 7 – DAC4MIN – DAC 4 Minimum Register Description

DACxMAX – DAC x Maximum Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	DACxMAX[11..0]											
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Table 8 – DACxMAX – DAC x Maximum Register

The DACxMAX register sets the maximum value for the digital-to-analog converter for the specified DAC. Only the lower 12 bits are significant as the DAC has 12 bits of resolution. DACs 1 through 3 and 5 through 7 all have default values of 0xFFF, as described in *Table 8 – DACxMAX – DAC x Maximum Register*, while DAC 4 has a minimum value of 0xCAF. Due to this singularity, the DAC4MAX register is shown in *Table 9 – DAC4MAX – DAC 4 Maximum Register*. To ensure the functionality of the 762, all maximum DAC values are factory set and cannot be changed by the user.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	DAC4MAX[11..0]											
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1

Table 9 – DAC4MAX – DAC 4 Maximum Register

DACxVAL – DAC x Value Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	DACxVAL[11..0]											
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Table 10 – DACxVAL – DAC x Value Register

The DACxVAL register sets the working value for the digital-to-analog converter for the specified DAC. Only the lower 12 bits are significant as the DAC has 12 bits of resolution. DACs 1 through 3 and 5 through 7 all have default values of 0x7FF, as described in *Table 10 – DACxVAL – DAC x Value Register*, while DAC 4 has a default value of 0x72F and DAC 7 has a default value of 0x68C. Due to these singularities, the DAC4VAL register is shown in *Table 11 – DAC4VAL – DAC 4 Value Register* and the DAC7VAL register is shown in *Table 12 – DAC7VAL – DAC 7 Value Register*.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	DAC4VAL[11..0]											
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	1	1	1	0	0	1	0	1	1	1	1

Table 11 – DAC4VAL – DAC 4 Value Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	DAC7VAL[11..0]											
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0	0

Table 12 – DAC7VAL – DAC 7 Value Register

DACEN – DAC Enable Configuration Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	X	X	X	X	DEN8	DEN7	DEN6	DEN5	DEN4	DEN3	DEN2	DEN1
Read/Write	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 13 – DACEN – DAC Enable Configuration Register

The DACEN register enables or disables specified DACs. Each DAC is controlled by one bit in the register, and by default, DEN[8..1] are set to 0 (off) as described in *Table 13 – DACEN – DAC Enable Configuration Register*. Write a 1 to the specific DAC enable bit location in order to enable the DAC output. When switched on, the DAC will immediately take the value specified in the corresponding DAC value register.

SERSEL – Serial Select Configuration Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	X	X	X	X	X	X	X	X	X	X	SSC	SS
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 14 – SERSEL – Serial Select Configuration Register

The SERSEL register controls how serial commands are handled by the 762. The serial select control (SSC) bit determines if the external 762 serial select input or the 762 memory controls whether the 762 reads I^2C or asynchronous serial as the method of communication. When SSC is set to 0, the external input controls this setting and when SSC is set to 1, the digital controller selects I^2C or asynchronous serial based on the value of the serial select (SS) bit. The SS bit, which only has an effect when SSC is set to 1, determines whether asynchronous serial (SS set to 1) or I^2C (SS cleared to 0) is used to communicate with a host.

MONGAIN – Monitor Gain Configuration Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	X	X	X	X	X	X	X	X	X	MGC	MG[1..0]	
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 15 – MONGAIN – Monitor Gain Configuration Register

The MONGAIN register controls how the inputs to the monitor gain are handled by the 762. The monitor gain control (MGC) bit determines if the two external 762 monitor gain inputs or the 762 memory control the monitor gain settings. When MGC is set to 0, the external inputs control this setting and when MGC is set to 1, the digital controller dictates the settings based on the value of the monitor gain bits (MG[1..0]). Monitor gain bits MG[1..0] only have an effect when MGC is set to 1. Whether monitor gain is set from external inputs or by 762 memory, highest gain is achieved with both monitor gain bits set to 1 and lowest gain occurs when both monitor gain bits are set to 0.

AMPSYNC – Amplifier Synchronization Polarity Configuration Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	X	X	X	X	X	X	X	X	X	X	ASP2	ASP1
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 16 – AMPSYNC – Amplifier Synchronization Polarity Configuration Register

The AMPSYNC register controls the polarity of the amplifier synchronization signals. Both signals behave the same, with the only difference being the corresponding amplifier synchronization signal. When the ASPx bit is set to 1, the polarity is logic 1 during the amplifier synchronization pulse and logic 0 when idle. Oppositely, when the ASPx bit is set to 0, the polarity is signal is logic 0 during the amplifier synchronization pulse and logic 1 when idle.

TRIGGER – Trigger Pulse Configuration Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	X	X	X	X	X	X	X	X	FTEN	ETEN	TGEN	TSEN
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Table 17 – TRIGGER – Trigger Pulse Enable Configuration Register

The TRIGGER register consists of four bits that control the operating mode of the 762. The follow trigger enable (FTEN) bit, when set to 1, forces the output pulse to follow the input pulse. As a result, the input pulse dictates the pulse width and repetition rate of the output pulse. The output pulse is identical to the input pulse. By default, FTEN is set to 1.

The ETEN bit allows the internal pulse generator to trigger action according to the rising edge of the external pulse input. If ETEN is set to 1 and the external pulse input is logic 1, the internal pulse generator executes one cycle using the stored delay and pulse width settings. The internal pulse generator does not execute without a rising edge external pulse input or when the ETEN bit is set to 0.

The use external trigger as gate enable (TGEN) bit allows the internal pulse generator to start or stop execution according to the state of the external pulse input. If TGEN is set to 1 and the external pulse input is logic 1, the internal pulse generator

executes using the stored delay and pulse width settings. The pulse generator continues to operate in this fashion until either the external pulse input state is driven low or the TGEN bit is set to 0.

The use external trigger as synchronization enable (TSEN) allows for the 762 pulse generator and delay timers to be reset, or synchronized, with the rising edge of the external pulse input. Following a rising edge trigger, the timers are set to 0 and the internal pulse generator begins counting. The amplifier synchronization and output pulses continue indefinitely using stored delay and pulse width settings until the TSEN bit is set to 0.

To enable the internal pulse generator to freely function without the need for external stimulus, set all TRIGGER register bits to 0 and set the desired delay, pulse width, and period settings for the trigger pulse and the desired delay and pulse width for the amplifier synchronization pulses.

PULEN – Pulse Output Enable Configuration Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	X	X	X	X	X	X	X	X	X	TEN	A1EN	A2EN
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Table 18 – PULEN –Pulse Output Enable Configuration Register

The trigger/pulse output enable (TEN), amplifier synchronization 1 output enable (A1EN), and amplifier synchronization 2 output enable (A2EN) all behave the same. When the TEN, A1EN, or A2EN bits are set to 1 the signals will be output by the 762. When set to 0, no output will occur. By default, only the trigger/pulse output is enabled.

I2CADDR – I^2C Address Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	X	X	X	X	X	I2C6	I2C5	I2C4	I2C3	I2C2	I2C1	I2C0
Read/Write	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0

Table 19 – I2CADDR – I^2C Address Register

Bits I2C[6..0] represent the 7-bit I^2C address of the 762. This address can be changed to any address within the 127 device I^2C address space. The unit does not respond to the I^2C global address of 0x00 unless the unit is specifically set to address 0x00 (this may cause interference with other devices who, by default, respond to commands on address 0x00). By default, the address is 0x50.

TECSEL – TEC Shutdown Select Configuration Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	X	X	X	X	X	X	X	X	X	X	TSC	TS
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20 – TECSEL – TEC Select Configuration Register

The TECSEL register controls how the TEC shutdown operates on the 762. The TEC shutdown select control (TSC) bit determines if the TEC never shuts down or the 762 memory controls the TEC shutdown. When TSC is set to 0, the TEC is always on and when TSC is set to 1, the digital controller selects the TEC shutdown value based on the TEC shutdown select

(TS) bit. The TS bit, which only has an effect when TSC is set to 1, determines whether the TEC is disabled/shut down (TC set to 1) or operating (TC cleared to 0).

ENSEL – Enable Select Configuration Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Designator	X	X	X	X	X	X	X	X	X	X	X	X	X	X	ENSC	ENS
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 21 – ENSEL – Enable Select Configuration Register

The ENSEL register controls how the enable bit is handled by the 762. The enable select control (ENSC) bit determines if the external 762 serial enable input or the 762 memory controls the enable. When ENSC is set to 0, the external input controls this setting and when ENSC is set to 1, the digital controller selects the enabled value based on the enable select (ENS) bit. The ENS bit, which only has an effect when ENSC is set to 1, determines whether the enable is disabled (ENS set to 1) or enabled (ENS cleared to 0). As a safety precaution, the enable signal can only be controlled by software when the enable input is driven low (device is enabled). When the enable input is driven high (disabled), the software control of the enable line is ignored and the unit is disabled.

Trigger/PW, and Amplifier Synchronization 1 and 2 Delay Registers

All three possible delay registers are 32 bits wide. Each increment of the counter represents one 100 MHz clock cycle, or 10 nanoseconds. The delay therefore has 10 nanosecond resolution over the range of time between 10*0 nanoseconds (0 seconds) to $10*2^{32}$ nanoseconds (42.94967296 seconds).

Trigger/PW, and Amplifier Synchronization 1 and 2 Pulse Width Registers

All three pulse width registers are 40 bits wide with only the upper 35 bits being significant. The upper 32 bits (bits 39..8) represent the number of 10 nanosecond intervals possible and the next 3 bits (bits 7..5) represent eight approximately 1.25 nanosecond intervals within each 10 nanosecond interval. In this way, it is possible to be 8 times more precise with the pulse width than with the delays. The pulse width therefore has approximately 1.25 nanosecond resolution over the range of time between 10*0 nanoseconds (0 seconds) to $10*2^{32}$ nanoseconds (42.94967296 seconds).

Trigger/PW Period Register

The internal trigger period specifies the time between internal trigger generated pulses. Similar to the way the delay and pulse width registers operate, the 32-bit wide period register represents the number of 10 nanosecond intervals between output pulses. The period therefore has 10 nanosecond resolution over the range of time between 10*0 nanoseconds (0 seconds) to $10*2^{32}$ nanoseconds (42.94967296 seconds).

Serial Number Register

The 32-bit serial number is written to the device at the time of programming at the factory.

Firmware Version Number Register

The 32-bit firmware version number is written to the device at the time of programming at the factory.

Fingerprint Register

The fingerprint register is a 32-bit register that allows the 762 to determine when invalid data is written to or read from the EEPROM. An invalid fingerprint, in most cases, means that other data is also corrupt and cannot be trusted. Since certain safety settings of the 762 are critical to proper operation, if the fingerprint check fails, the unit reverts to a factory default

setting. If the default setting cannot be attained, the unit will issue an internal memory error and will not operate until the memory error is resolved.

Commands and Responses

Host to SLDD transfer occurs via transistor-transistor logic (TTL) level asynchronous serial or I^2C synchronous serial. All transactions are initiated by the host system. When communicating with asynchronous serial, the SLDD automatically transmits command responses; however, with I^2C synchronous serial, the host must explicitly initiate the reading of the response.

The format of asynchronous serial packets is human readable, and therefore the 762 may be controlled using a standard terminal. All commands are ASCII characters, each one-byte parameter is transmitted and received as two ASCII characters, and the command is terminated with a carriage return (0x0D). For example, if the host attempts to write data 0xD3 to memory address 0x54, the string is as follows: `W54d3<CR>`. Please note values are sent using characters 0-9 and a-f (case sensitive). The SLDD response is immediate, and, in the case of the sample Write Data Command, identical to the issued command. If an error occurs, an error message will be returned instead of the Write Data Command Response. See the section titled *Error Response (Asynchronous Serial Only)* on page 35 for more information regarding asynchronous serial error response codes and descriptions.

If the host system communicates via I^2C, the same Write Data Command message to the 762 would be as follows: 0x50, 0x57, 0x54, 0xD3. The first byte is the default I^2C slave address. The 762 will respond and acknowledge all I^2C read or I^2C write commands addressed to 0x50 unless the I^2C address is changed in memory prior to communication. In contrast with asynchronous serial, all single byte parameters of an I^2C transaction are single bytes. No error message will be returned, but instead an error flag is raised in the command response to advise the host system an error occurred.

Command Overview

Serial control of the 762 is achieved using six commands. Each command is supported for asynchronous serial and I^2C. Commands are shown in *Table 22 – Description of Serial Commands*, and are described in detail over the following sections.

Command	ASCII Command	Hex Command	Parameter 1 (if necessary)	Parameter 2 (if necessary)
Write Data	W	0x57	Memory Address	Data to Write
Read Data	R	0x52	Memory Address	N/A
Load from EEPROM	L	0x4C	N/A	N/A
Save to EEPROM	S	0x53	N/A	N/A
Get Status	T	0x54	N/A	N/A
Bank Switch	B	0x42	Bank Number	N/A

Table 22 – Description of Serial Commands

Write Data

Writing an 8-bit data value anywhere between memory address 0x20 to 0x6F can be completed using a Write Data Command ('W', 0x57). A write operation requires a 7-bit memory address and a data byte. At the completion of the write cycle, the data received by the SLDD is saved in the received SRAM address. Note, the Write Data Command stores the data byte in volatile SRAM. If the value being written needs to be saved to non-volatile memory, use the Save to EEPROM command following the data write sequence.

To verify a Write Data Command has been successful, the contents at the memory address must match what was written. This can be achieved in several ways, but is most easily checked immediately following a Write Data Command by reading and processing of the command response.

Asynchronous Serial Command

Following an asynchronous serial Write Data Command byte, the host system must transmit the memory address to where the data will be written. The data to write is sent next, and then the command is terminated with a carriage return (0x0D). The carriage return forces the 762 to write the data to the memory address, and to transmit a response. See *Figure 2 – Asynchronous Serial Write Data Command Timing Diagram* for a bit level depiction of the data to be sent. MAUN represents the upper nibble of the memory address and MALN represents the lower nibble of the memory address, while DUN is the upper nibble of the data byte and DLN is the lower nibble.

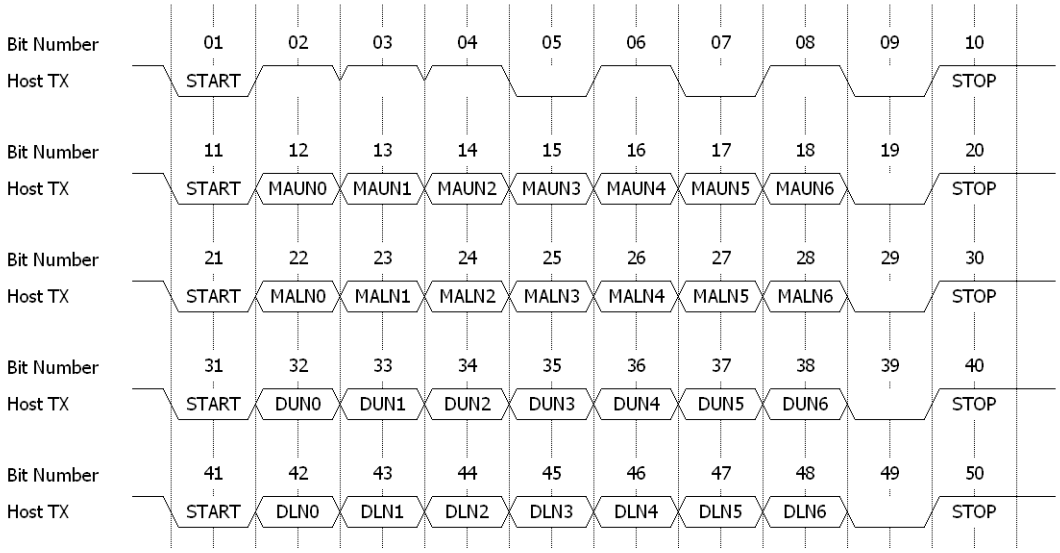

Figure 2 – Asynchronous Serial Write Data Command Timing Diagram

Asynchronous Serial Response

After an asynchronous serial Write Data Command, the host system replies with a 5-byte response. The first byte is the received command, bytes 2 and 3 are the ASCII representation of the hexadecimal upper and lower nibbles of the received memory address, and bytes 4 and 5 make up the ASCII representation of the hexadecimal upper and lower nibbles of the data contained at the memory address. Bytes 2 through 5 are ASCII values between '0' and '9' or between 'a' and 'f'. The response is terminated with a carriage return. Refer to *Figure 3 – Asynchronous Serial Write Data Command Response Timing Diagram*.

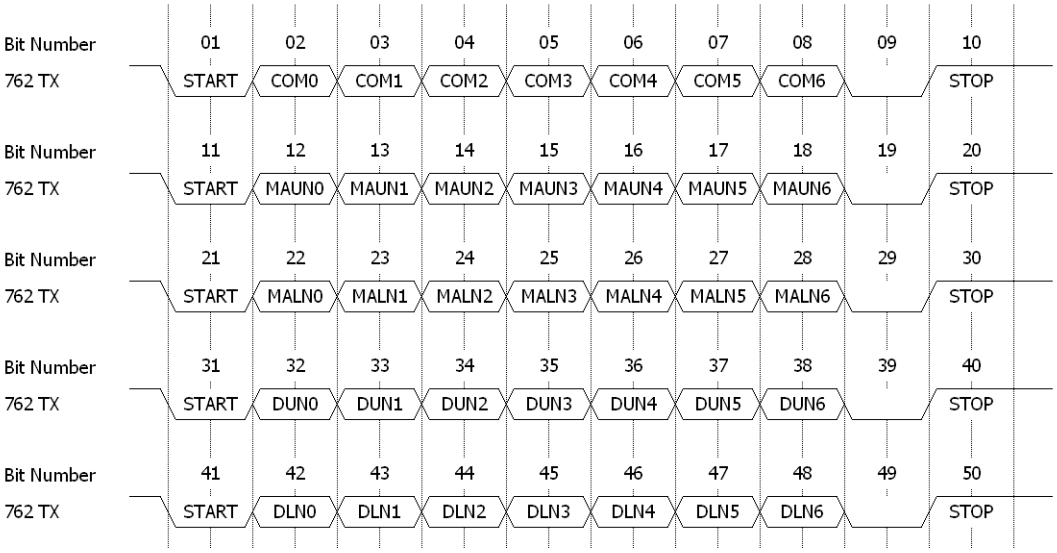

Figure 3 – Asynchronous Serial Write Data Command Response Timing Diagram

I²C Command

An I²C write begins with an I²C start condition followed by the I²C slave address (SA[6..0]) of the SLDD. If the slave address received matches the programmed slave address, the 762 acknowledges receipt and expects to receive a command with the next eight bits – the lower seven of which are meaningful. Should a Write Data Command be received, the SLDD acknowledges and is ready to receive the memory address (MA[6..0]). Similar to the command reception, only the lower seven bits of the memory address are significant. Once the memory address is received, an acknowledge is issued and the 762 waits to receive the 8-bit data value (D[7..0]). When finished, the host system issues an I²C stop condition and the data received is stored at the received memory address. The data is only written to memory when a valid stop condition is detected by the SLDD. See *Figure 4 – I2C Write Data Command Timing Diagram* for additional direction.

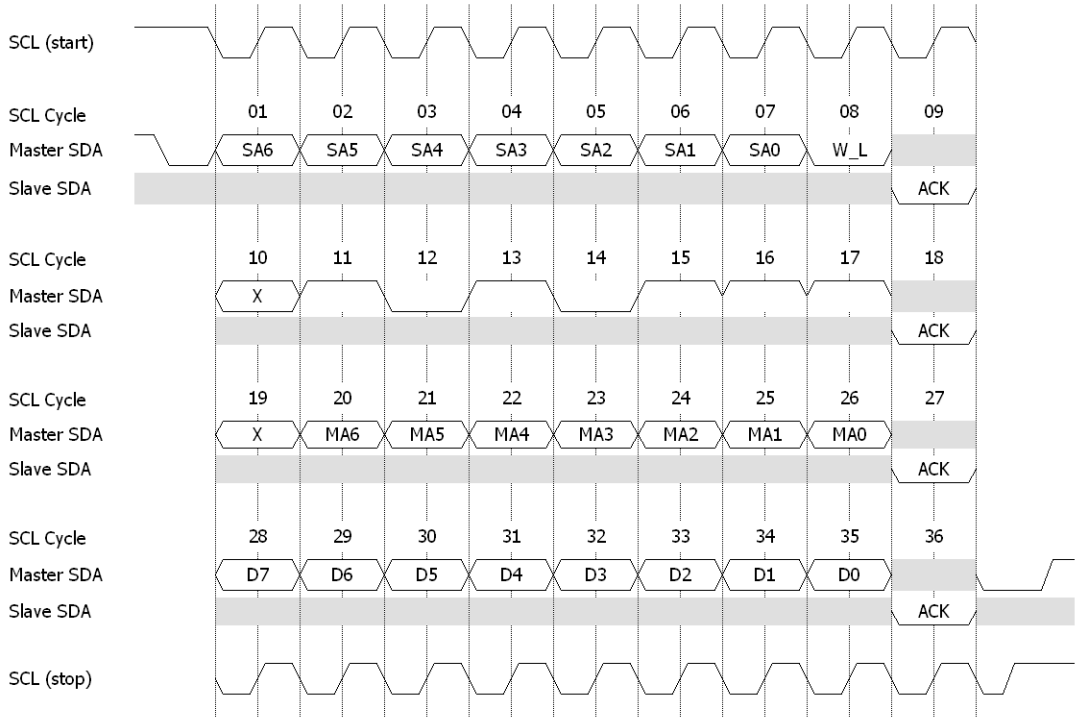

Figure 4 – I²C Write Data Command Timing Diagram

I²C Response

Since I²C is a host driven protocol, the host system must request to read the Write Data Command response. The I²C read begins with an I²C start condition followed by the I²C slave address of the SLDD. If the slave address received matches the programmed slave address, the 762 acknowledges receipt and sends the most previously received command (COM[6..0]). The lower seven bits represent the command the 762 received, and the most significant bit indicates whether the command was recognized (0 if recognized, 1 if not recognized). If recognized and following a master acknowledge, the SLDD sends the memory address received, followed by the 8-bit data value as read from SRAM. It is up to the host system to ensure the command, memory address, and data match expected values.

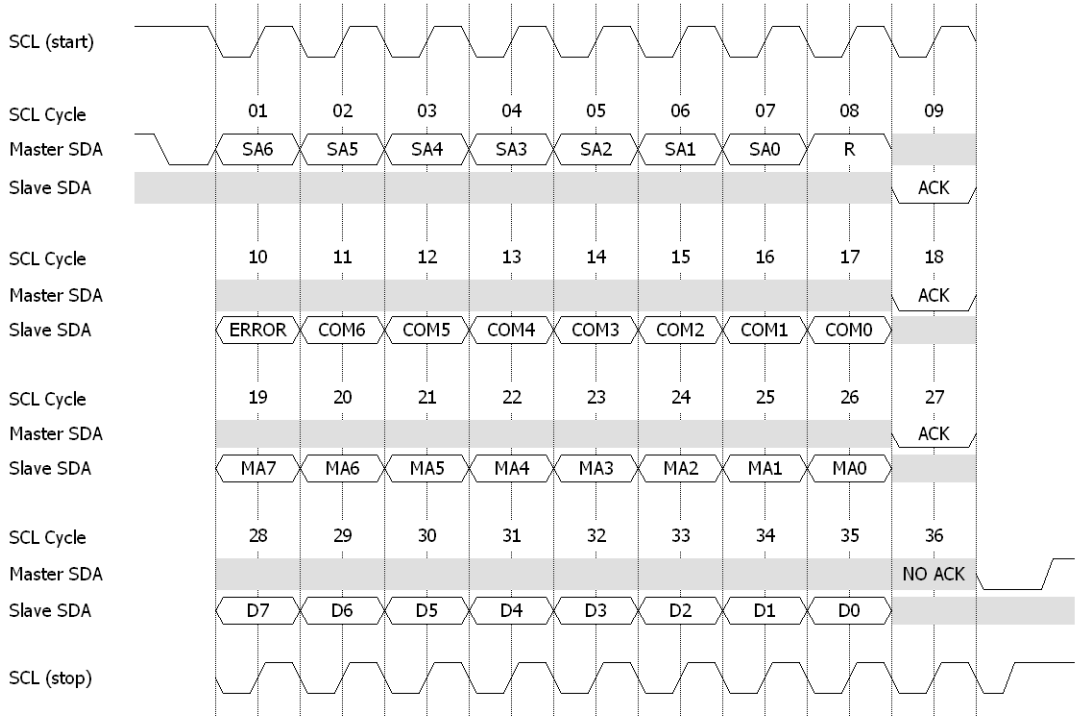

Figure 5 – I²C Write Data Command Response Timing Diagram

As can be seen in *Figure 5 – I2C Write Data Command Response Timing Diagram*, the host does not need to acknowledge the receipt of the incoming data. If the host chooses to acknowledge, the 762 increments the memory address, reads the data contents at the incremented memory address, and transmits the data to the master. This process continues indefinitely until the host does not acknowledge receipt of the data and issues an I²C stop condition. This behavior is useful because it allows the host to read the entire SRAM block contents without having to issue separate Read Data Commands.

Read Data

Reading an 8-bit data value anywhere in the memory space can be completed using a Read Data Command ('R', 0x52). A read operation requires only a 7-bit memory address parameter. Immediately after the command is issued, the SLDD reads the contents of the memory address and prepares to serially transmit the response.

Asynchronous Serial Command

Following an asynchronous serial Read Data Command byte, the host system must transmit the memory address from where the data will be read. After the memory address is sent, the command is terminated with a carriage return (0x0D). The carriage return forces the 762 to read the contents of the memory address, and to transmit a response. See *Figure 6 – Asynchronous Serial Read Data Command Timing Diagram* for a bit level depiction of the data to be sent. MAUN represents the upper nibble of the memory address and MALN represents the lower nibble of the memory address.

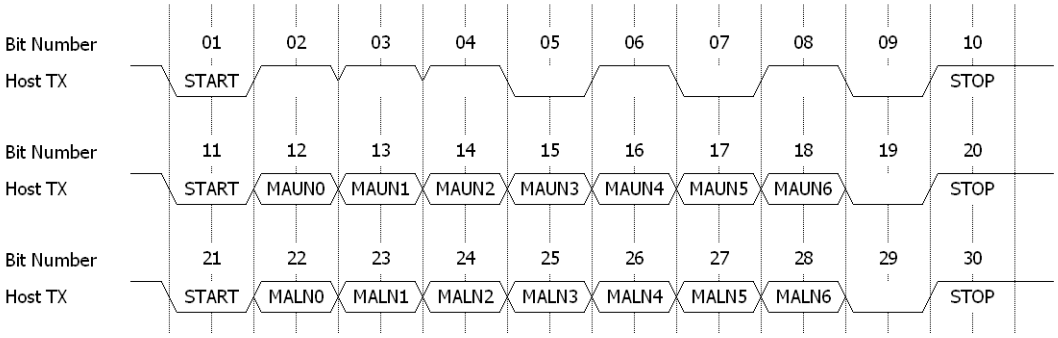

Figure 6 – Asynchronous Serial Read Data Command Timing Diagram

Asynchronous Serial Response

After an asynchronous serial Read Data Command, the host system replies with a 5-byte response. The first byte is the received command, bytes 2 and 3 are the ASCII representation of the hexadecimal upper and lower nibbles of the received memory address, and bytes 4 and 5 make up the ASCII representation of the hexadecimal upper and lower nibbles of the data contained at the memory address. Bytes 2 through 5 are ASCII values between '0' and '9' or between 'a' and 'f'. The response is terminated with a carriage return. Refer to *Figure 7 – Asynchronous Serial Read Data Command Response Timing Diagram*.

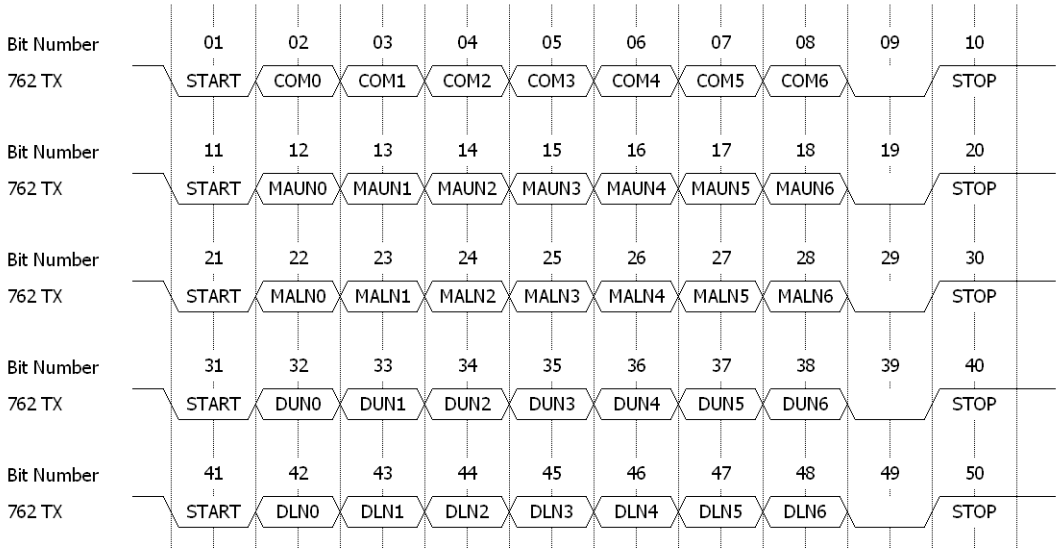

Figure 7 – Asynchronous Serial Read Data Command Response Timing Diagram

I^2C Command

The host initiates an I^2C write with an I^2C start condition followed by the I^2C slave address (SA[6..0]) of the SLDD. If the slave address received matches the programmed slave address, the 762 acknowledges receipt and expects to receive a command with the next eight bits – the lower seven of which are meaningful. Should a Read Data Command be received, the SLDD acknowledges and is ready to receive the memory address (MA[6..0]). Similar to the command reception, only the lower seven bits of the memory address are significant. Once the memory address is received, an acknowledge is issued and the 762 waits for an I^2C stop condition. See *Figure 8 – I2C Read Data Command Timing Diagram* for additional direction.

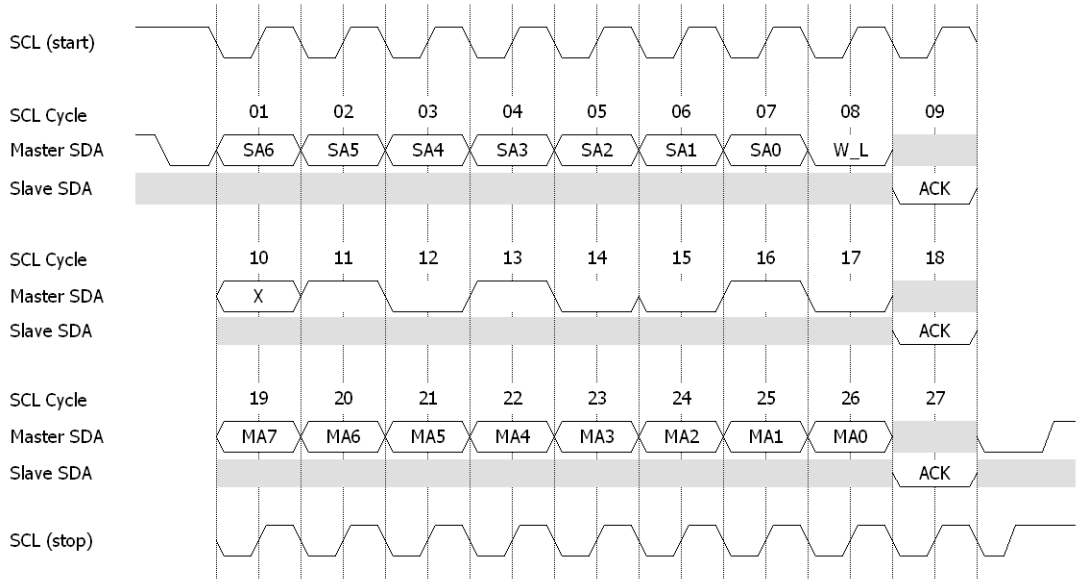

Figure 8 – I²C Read Data Command Timing Diagram

I²C Response

The host system must request to read the Read Data Command response. The I²C read begins with an I²C start condition followed by the I²C slave address of the SLDD. If the slave address received matches the programmed slave address, the 762 acknowledges receipt and sends the most previously received command (COM[6..0]). The lower seven bits represent the command the 762 received, and the most significant bit indicates whether the command was recognized (0 if recognized, 1 if not recognized). Following a master acknowledge, the SLDD sends the memory address received, followed by the 8-bit data value as read from SRAM. It is up to the host system to ensure the command, and memory address match.

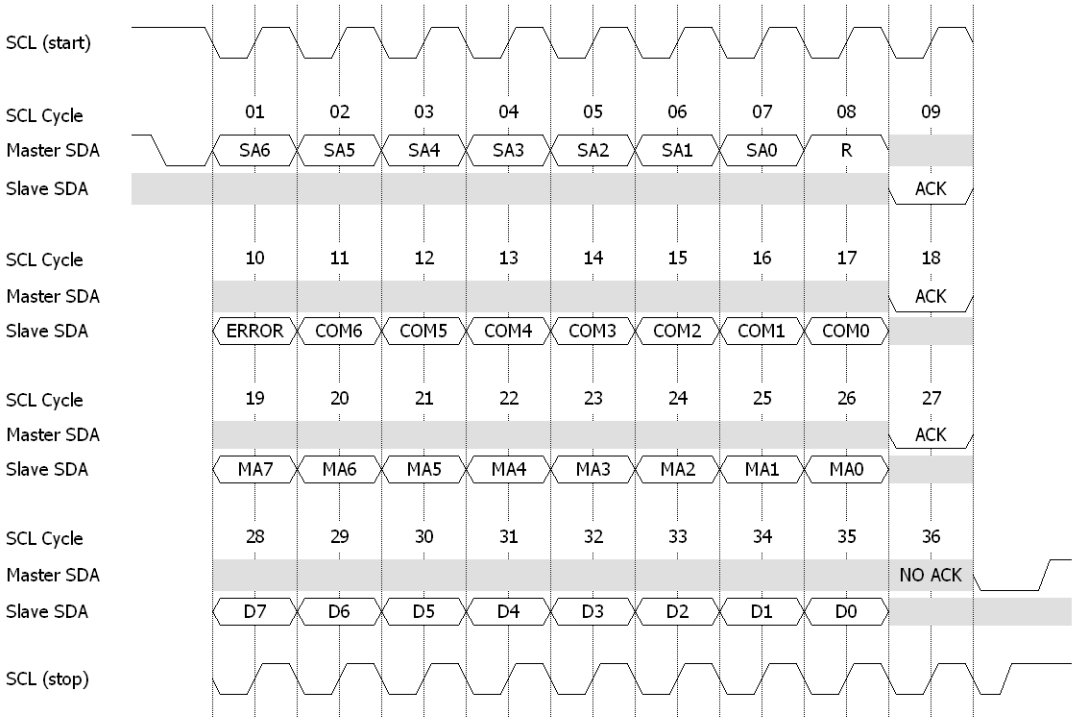

Figure 9 – I²C Read Data Command Response Timing Diagram

As can be seen in *Figure 9 – I2C Read Data Command Response Timing Diagram*, the host does not need to acknowledge the receipt of the incoming data. If the host chooses to acknowledge, the 762 increments the memory address, reads the data

contents at the incremented memory address, and transmits the data to the master. This process continues indefinitely until the host does not acknowledge receipt of the data and issues an I^2C stop condition. This behavior is useful because it allows the host to read the entire SRAM block contents without having to issue separate Read Data Commands.

Load from EEPROM

On power up, the EEPROM contents are automatically copied to SRAM. Should the host require the EEPROM to overwrite SRAM at any point thereafter, the host can use the Load from EEPROM Command. The command overwrites all four 128 byte SRAM blocks.

Asynchronous Serial Command

Following an asynchronous serial Load from EEPROM Command, the host system must terminate the command with a carriage return (0x0D). The carriage return forces the 762 to load the contents of SRAM with the contents of the EEPROM, and to transmit a status response. See *Figure 10 – Asynchronous Serial Load from EEPROM Command Timing Diagram* for a bit level depiction of the data to be sent.

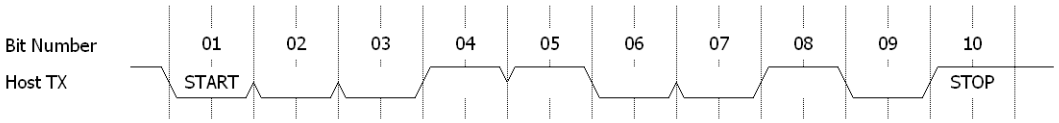

Figure 10 – Asynchronous Serial Load from EEPROM Command Timing Diagram

Asynchronous Serial Response

After an asynchronous serial Load from EEPROM Command, the host system replies with a 5-byte status response. The first byte is the received command, bytes 2 and 3 are the ASCII representation of the hexadecimal upper status byte, and bytes 4 and 5 make up the ASCII representation of the hexadecimal lower status byte. Bytes 4 through 5 are ASCII values between '0' and '9' or between 'a' and 'f'. Refer to *Figure 11 – Asynchronous Serial Load from EEPROM Command Response Timing Diagram*.

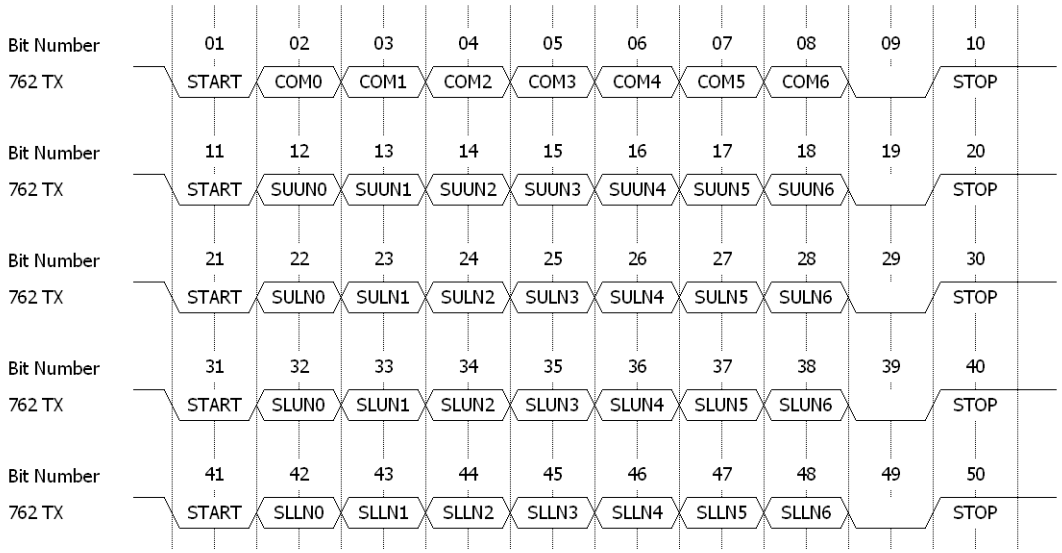

Figure 11 – Asynchronous Serial Load from EEPROM Command Response Timing Diagram

The status register is 16 bits wide, and is broken down in four nibbles. The upper nibble of the most significant status byte is represented above as SUUN, the lower nibble of the most significant status byte is SULN. Similarly, the upper and lower nibbles of the least significant byte are SLUN and SLLN, respectively. The response is terminated with a carriage return.

I²C Command

The host initiates an I²C write with an I²C start condition followed by the I²C slave address (SA[6..0]) of the SLDD. If the slave address received matches the programmed slave address, the 762 acknowledges receipt and expects to receive a command with the next eight bits – the lower seven of which are meaningful. Should a Load from EEPROM Command be received, the SLDD acknowledges and waits for an I²C stop condition. Once the stop condition is received, the SLDD writes SRAM with the contents of the EEPROM. Further commands will not be accepted until the load is finished.

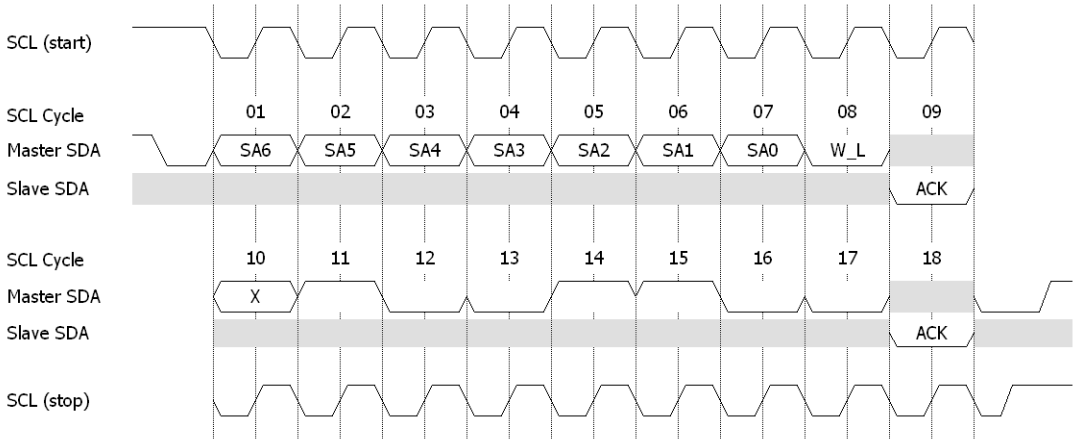

Figure 12 – I²C Load from EEPROM Command Timing Diagram

I²C Response

The Load from EEPROM Command Response is identical to the Get Status Command Response, the Save to EEPROM Command Response, and the Bank Switch Command Response. All four commands generate a status response and the 16-bit status register (S[15..0]) is written to the host system.

The host system must request to read the response. The I²C read begins with an I²C start condition followed by the I²C slave address of the SLDD. If the slave address received matches the programmed slave address, the 762 acknowledges receipt and sends the most previously received command (COM[6..0]). The lower seven bits represent the command the 762 received, and the most significant bit indicates whether the command was recognized (0 if recognized, 1 if not recognized). Following a master acknowledge, the SLDD sends the upper byte of the status register, followed by the lower byte of the status register. It is up to the host system to check the appropriate ready bits of the status register to determine when a new command may be safely issued.

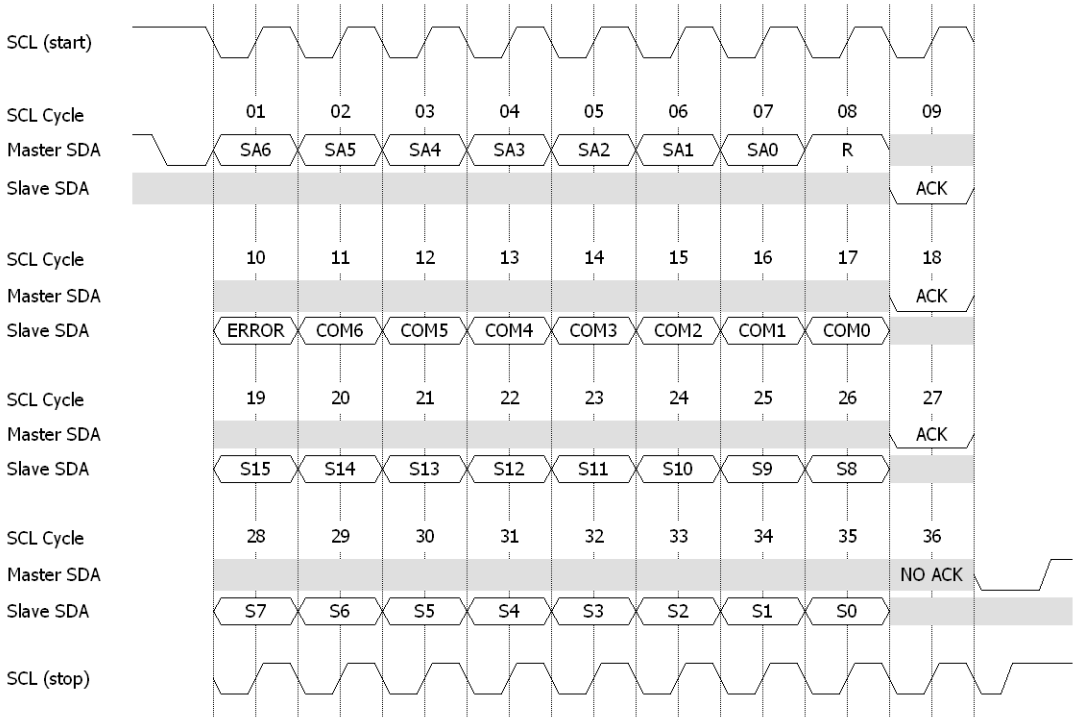

Figure 13 – I²C Load from EEPROM Command Response Timing Diagram

As can be seen in *Figure 13 – I2C Load from EEPROM Command Response Timing Diagram*, the host does not need to acknowledge the receipt of the incoming data. If the host chooses to acknowledge, the 762 updates the status register and repeatedly sends the upper and lower portions of the register to the host. In this way, it is possible to poll the SLDD ready bits and easily determine when another command may be issued.

Save to EEPROM

The 762 only saves the contents of SRAM to the EEPROM when requested to do so. As a result, the Save to EEPROM Command is the mechanism the host system uses to maintain SLDD state through a power cycle. The command overwrites all four 128 byte blocks in the EEPROM with the contents of SRAM.

Asynchronous Serial Command

Following an asynchronous serial Save to EEPROM Command byte, the host system must terminate the command with a carriage return (0x0D). The carriage return forces the 762 to save the contents of SRAM to the EEPROM, and to transmit a status response. See *Figure 14 – Asynchronous Serial Save to EEPROM Command Timing Diagram* for a bit level depiction of the data to be sent.

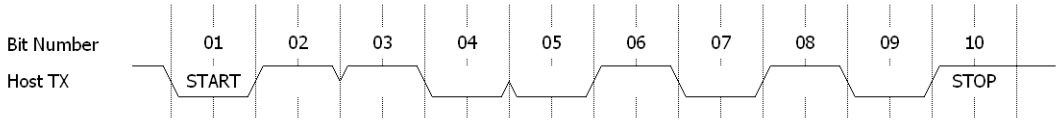

Figure 14 – Asynchronous Serial Save to EEPROM Command Timing Diagram

Asynchronous Serial Response

After an asynchronous serial Save to EEPROM Command, the host system replies with a 5-byte status response. The first byte is the received command, bytes 2 and 3 are the ASCII representation of the hexadecimal upper status byte, and bytes 4 and 5 make up the ASCII representation of the hexadecimal lower status byte. Bytes 2 through 5 are ASCII values between '0' and '9' or between 'a' and 'f'. Refer to *Figure 15 – Asynchronous Serial Save to EEPROM Command Response Timing Diagram*.

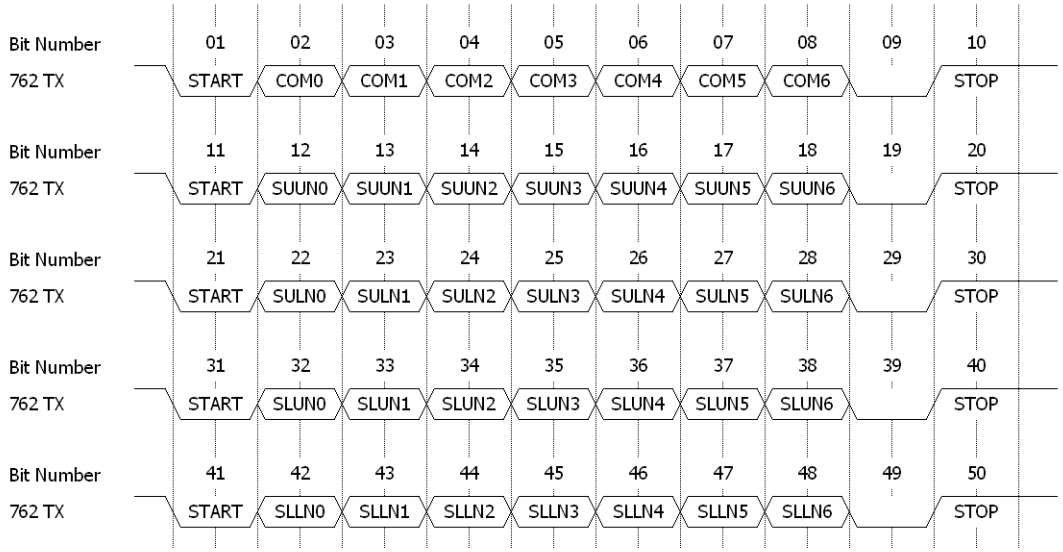

Figure 15 – Asynchronous Serial Save to EEPROM Command Response Timing Diagram

The status register is 16 bits wide, and is broken down in four nibbles. The upper nibble of the most significant status byte is represented above as SUUN, the lower nibble of the most significant status byte is SULN. Similarly, the upper and lower nibbles of the least significant byte are SLUN and SLLN, respectively. The response is terminated with a carriage return.

I²C Command

The host initiates an I²C write with an I²C start condition followed by the I²C slave address (SA[6..0]) of the SLDD. If the slave address received matches the programmed slave address, the 762 acknowledges receipt and expects to receive a command with the next eight bits – the lower seven of which are meaningful. Should a Save to EEPROM Command be received, the SLDD acknowledges and waits for an I²C stop condition. Once the stop condition is received, the SLDD saves the contents of SRAM to the EEPROM, and will not accept further commands until the save is finished.

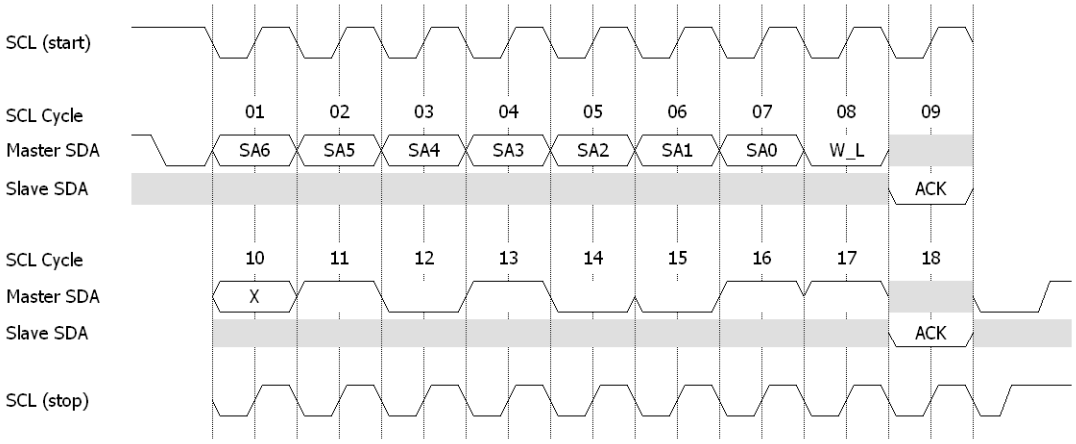

Figure 16 – I²C Save to EEPROM Command Timing Diagram

I²C Response

The Save to EEPROM Command Response is identical to the Get Status Command Response, the Load from EEPROM Command Response, and the Bank Switch Command Response. All four commands generate a status response and the 16-bit status register (S[15..0]) is written to the host system.

The host system must request to read the response. The I²C read begins with an I²C start condition followed by the I²C slave address of the SLDD. If the slave address received matches the programmed slave address, the 762 acknowledges receipt and sends the most previously received command (COM[6..0]). The lower seven bits represent the command the 762 received,

and the most significant bit indicates whether the command was recognized (0 if recognized, 1 if not recognized). Following a master acknowledge, the SLDD sends the upper byte of the status register, followed by the lower byte of the status register. It is up to the host system to check the appropriate ready bits of the status register to determine when a new command may be safely issued.

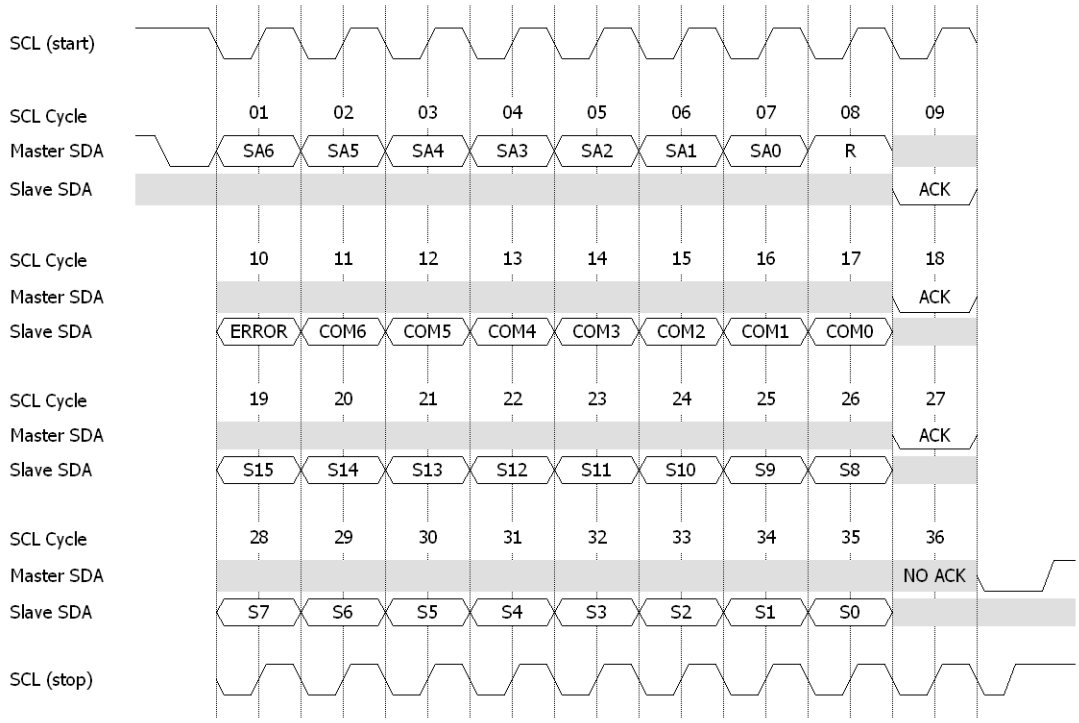

Figure 17 – I²C Save to EEPROM Command Response Timing Diagram

As can be seen in *Figure 17 – I2C Save to EEPROM Command Response Timing Diagram*, the host does not need to acknowledge the receipt of the incoming data. If the host chooses to acknowledge, the 762 updates the status register and repeatedly sends the upper and lower portions of the register to the host.

Get Status

Occasionally, the host system requires a snapshot of the current SLDD status. The Get Status Command ('T', 0x54) is used to retrieve the 16-bit status register described in *Table 23 – 16-bit Status Register Description*.

	7	6	5	4	3	2	1	0
Upper Byte	Reserved	Reserved	Reserved	Enable (Low)	Memory Bank Bit 1	Memory Bank Bit 0	762 Ready	DAC Ready
Lower Byte	EEPROM Ready	Temperature Fault (Low)	Over Current Fault (Low)	TEC Disabled	762 Error	Internal Memory Error	DAC Error	EEPROM Error

Table 23 – 16-bit Status Register Description

Bits 4 and 3 of the upper byte constitute the current memory bank from which the SLDD is currently operating. The lower two bits of the upper byte and the upper bit of the lower byte are positive logic ready flags. If the 762, DAC, and EEPROM are not working on a task and are able to accommodate user commands, the ready bits are driven high; otherwise, the ready bits are driven low. Bit 4 of the upper byte indicates the current state of the 762 enable. When driven low, the unit is enabled and when driven high, the unit is disabled. A temperature or over current fault condition is occurring when the Temperature Fault or Over Current Fault bits are driven low. No fault is occurring when the bits are driven high. The TE cooler is disabled when the TEC Shutdown bit is driven high and is enabled when the bit is driven low. Errors are indicated by the lower four bits of the lower byte. An error has occurred internal to the 762 when the 762 Error flag is driven high. When the 762 fails to

read valid settings from the internal memory, the Internal Memory Flag is driven high. Should a DAC or EEPROM error have occurred, the respective flags are driven high. If there are no errors, all four lower bits of the lower byte are driven low.

Asynchronous Serial Command

Following an asynchronous serial Get Status Command, the host system must terminate the command with a carriage return (0x0D). The carriage return forces the 762 to perform the command and to transmit a response. See *Figure 18 – Asynchronous Serial Get Status Command Timing Diagram* for a bit level depiction of the data to be sent.

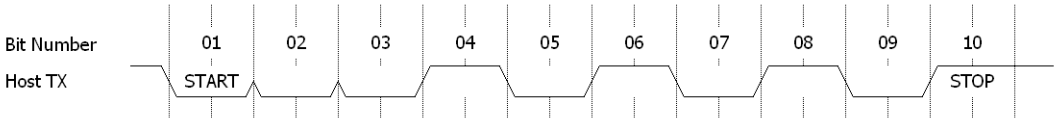

Figure 18 – Asynchronous Serial Get Status Command Timing Diagram

Asynchronous Serial Response

After an asynchronous serial Get Status Command, the host system replies with a 5-byte status response. The first byte is the received command, bytes 2 and 3 are the ASCII representation of the hexadecimal upper status byte, and bytes 4 and 5 make up the ASCII representation of the hexadecimal lower status byte. Bytes 2 through 5 are ASCII values between '0' and '9' or between 'a' and 'f'. Refer to *Figure 19 – Asynchronous Serial Get Status Command Response Timing Diagram*.

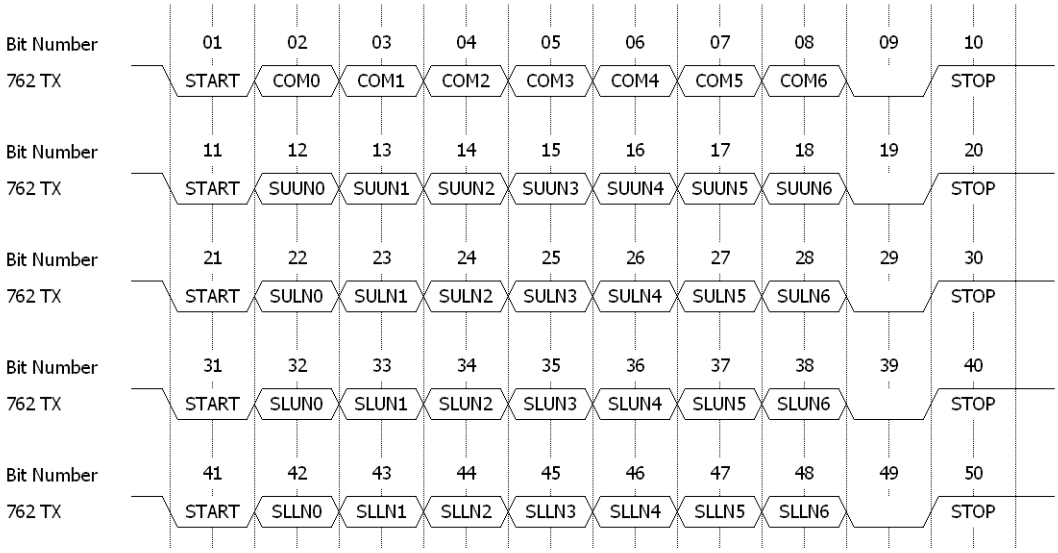

Figure 19 – Asynchronous Serial Get Status Command Response Timing Diagram

The status register is 16 bits wide, and is broken down in four nibbles. The upper nibble of the most significant status byte is represented above as SUUN, the lower nibble of the most significant status byte is SULN. Similarly, the upper and lower nibbles of the least significant byte are SLUN and SLLN, respectively. The response is terminated with a carriage return.

I^2C Command

The host initiates an I^2C write with an I^2C start condition followed by the I^2C slave address (SA[6..0]) of the SLDD. If the slave address received matches the programmed slave address, the 762 acknowledges receipt and expects to receive a command with the next eight bits – the lower seven of which are meaningful. Should a Get Status Command be received, the SLDD acknowledges and waits for an I^2C stop condition.

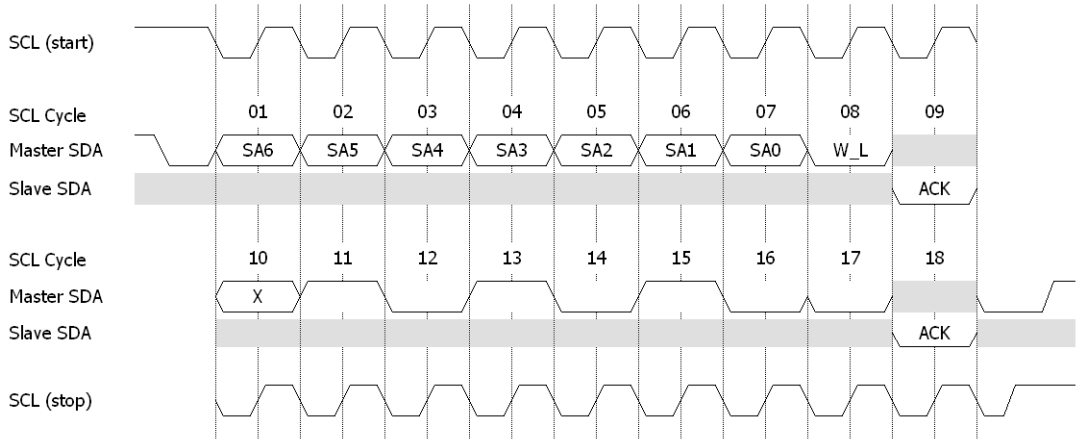

Figure 20 – I²C Get Status Command Timing Diagram

I²C Response

The Get Status Command Response is identical to the Save to EEPROM Command Response, the Load from EEPROM Command Response, and the Bank Switch Command Response. All four commands generate a status response and the 16-bit status register (S[15..0]) is written to the host system.

The host system must request to read the status response. The I²C read begins with an I²C start condition followed by the I²C slave address of the SLDD. If the slave address received matches the programmed slave address, the 762 acknowledges receipt and sends the most previously received command (COM[6..0]). The lower seven bits represent the command the 762 received, and the most significant bit indicates whether the command was recognized (0 if recognized, 1 if not recognized). Following a master acknowledge, the SLDD sends the upper byte of the status register, followed by the lower byte of the status register. It is up to the host system to check the appropriate ready bits of the status register to determine when a new command may be safely issued.

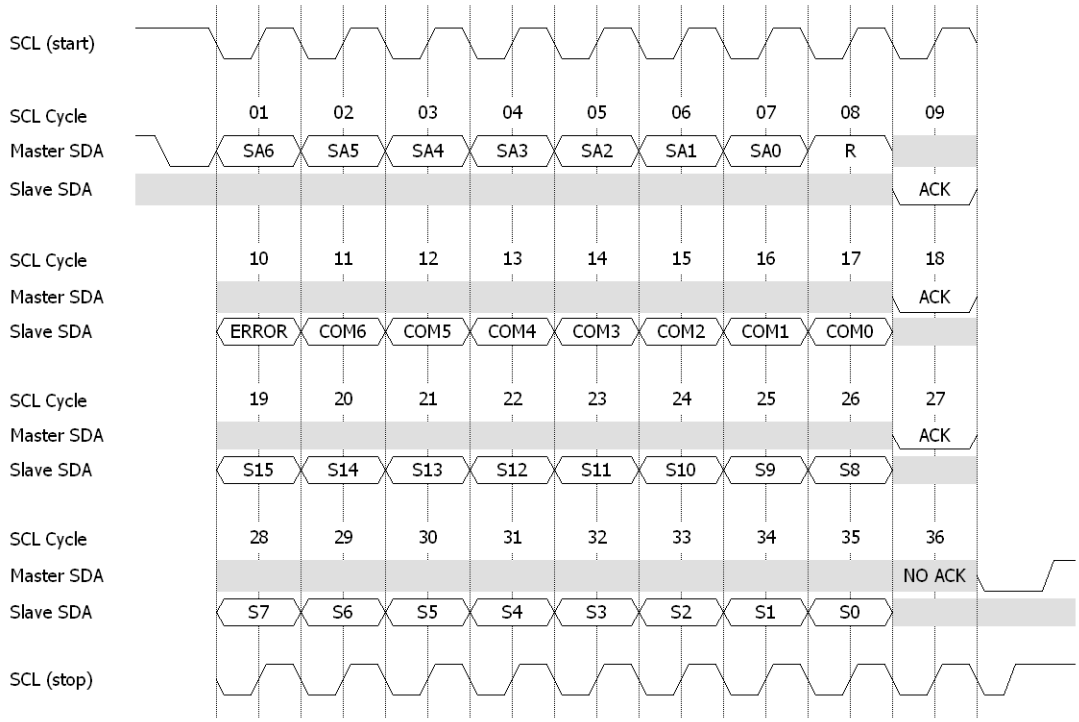

Figure 21 – I²C Get Status Command Response Timing Diagram

As can be seen in *Figure 21 – I2C Get Status Command Response Timing Diagram*, the host does not need to acknowledge the receipt of the incoming data. If the host chooses to acknowledge, the 762 updates the status register and repeatedly sends the upper and lower portions of the register to the host.

Bank Switch

The 762 supports up to four configurations with each being stored in one of the SLDDs four memory banks. By default, the system boots into memory bank 0, but the Bank Switch Command ('B', 0x42) allows the host system to change the memory bank from which the 762 operates. A bank switch is not maintained through a power cycle, so the host system must reissue the command in the event of power failure.

Asynchronous Serial Command

Following an asynchronous serial Bank Switch Command, the host system must specify the memory bank the 762 will use to operate. One data byte between ASCII '0' (0x30) and '3' (0x33) should be transmitted and followed by a carriage return (0x0D) to execute the bank switch. See *Figure 22 – Asynchronous Serial Bank Switch Command Timing Diagram* for a bit level depiction of the data to be sent.

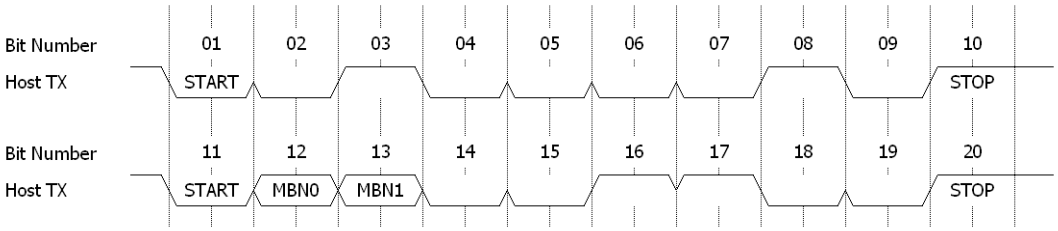

Figure 22 – Asynchronous Serial Bank Switch Command Timing Diagram

Asynchronous Serial Response

After an asynchronous serial Bank Switch Command, the host system replies with a 5-byte status response. The first byte is the received command, bytes 2 and 3 are the ASCII representation of the hexadecimal upper status byte, and bytes 4 and 5 make up the ASCII representation of the hexadecimal lower status byte. Bytes 2 through 5 are ASCII values between '0' and '9' or between 'a' and 'f'. Refer to *Figure 23 – Asynchronous Serial Bank Switch Command Response Timing Diagram*.

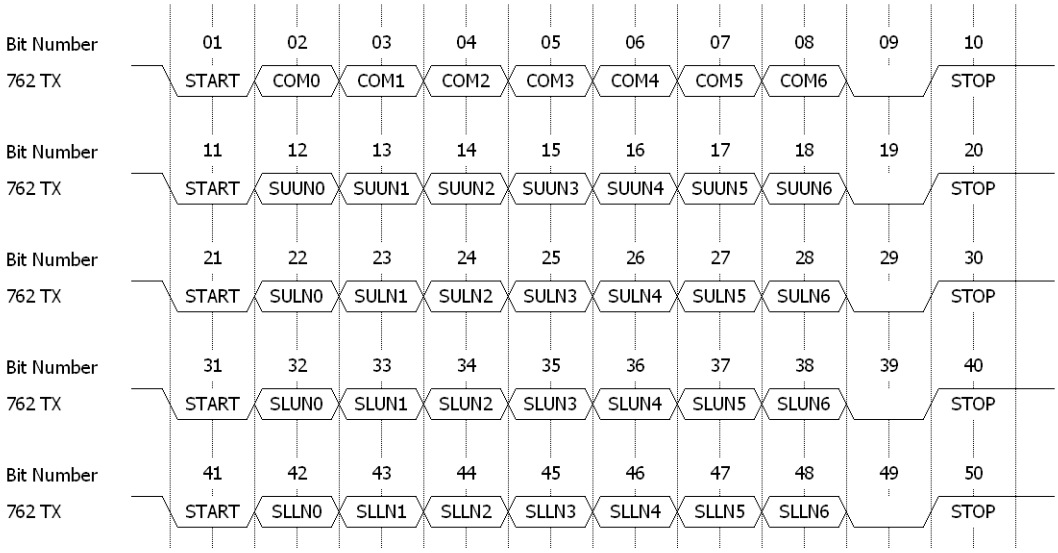

Figure 23 – Asynchronous Serial Bank Switch Command Response Timing Diagram

The status register is 16 bits wide, and is broken down in four nibbles. The upper nibble of the most significant status byte is represented above as SUUN, the lower nibble of the most significant status byte is SULN. Similarly, the upper and lower nibbles of the least significant byte are SLUN and SLLN, respectively. The response is terminated with a carriage return.

I²C Command

The host initiates an I²C write with an I²C start condition followed by the I²C slave address (SA[6..0]) of the SLDD. If the slave address received matches the programmed slave address, the 762 acknowledges receipt and expects to receive a command with the next eight bits – the lower seven of which are meaningful. Should a Bank Switch Command be received, the SLDD acknowledges and waits to receive the new memory bank number. While the host sends an eight bit memory bank number, only the least significant two bits are meaningful (MBN[1..0]). Following receipt of the new memory bank number, the SLDD expects to receive an I²C stop condition. Once the stop condition is received, the SLDD changes the memory bank and updates all internal registers.

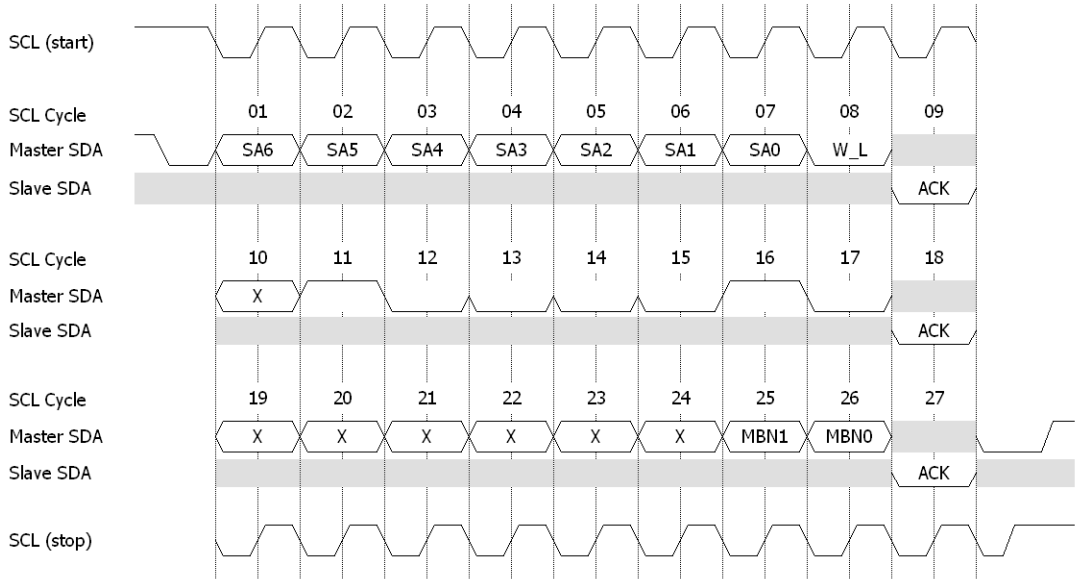

Figure 24 – I²C Bank Switch Command Timing Diagram

I²C Response

The Bank Switch Command Response is identical to the Save to EEPROM Command Response, the Load from EEPROM Command Response, and the Get Status Command Response. All four commands generate a status response and the 16-bit status register (S[15..0]) is written to the host system.

The host system must request to read the response. The I²C read begins with an I²C start condition followed by the I²C slave address of the SLDD. If the slave address received matches the programmed slave address, the 762 acknowledges receipt and sends the most previously received command (COM[6..0]). The lower seven bits represent the command the 762 received, and the most significant bit indicates whether the command was recognized (0 if recognized, 1 if not recognized). Following a master acknowledge, the SLDD sends the upper byte of the status register, followed by the lower byte of the status register. It is up to the host system to monitor the memory bank bits in the status response to ensure the bank switch has been successful.

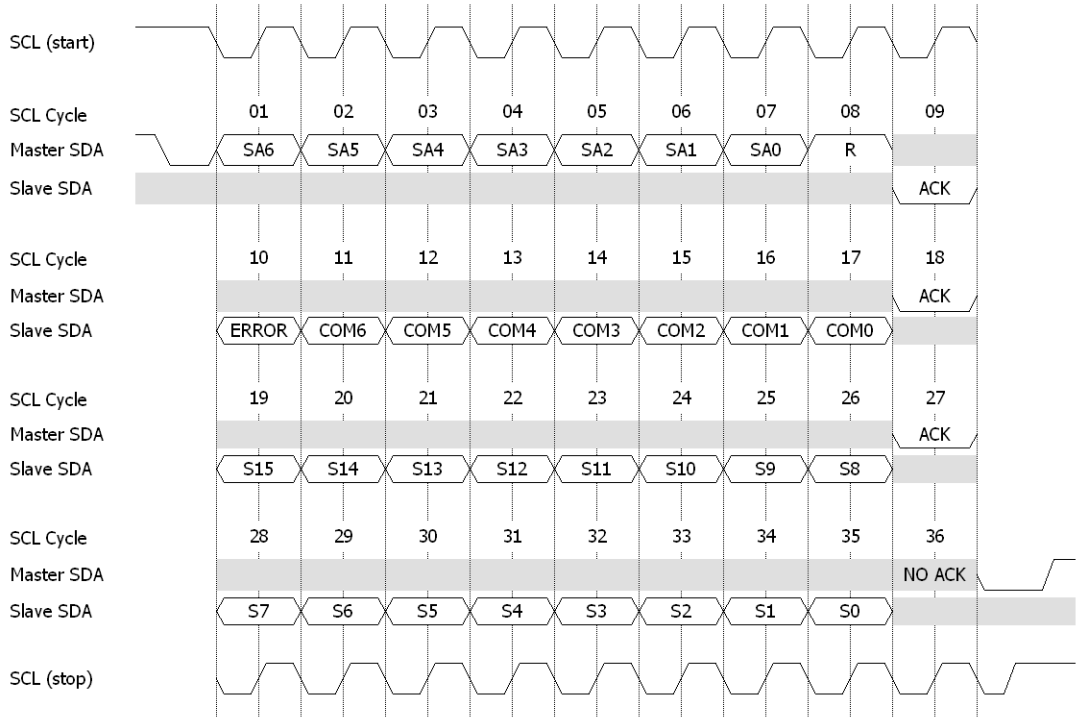

Figure 25 – I²C Bank Switch Command Response Timing Diagram

As can be seen in *Figure 25 – I2C Bank Switch Command Response Timing Diagram*, the host does not need to acknowledge the receipt of the incoming status register. If the host chooses to acknowledge, the 762 updates the status register and repeatedly sends the upper and lower portions of the register to the host.

Error Response (Asynchronous Serial Only)

Asynchronous Serial Response

When using asynchronous serial to interface to the SLDD, the system sometimes advises the user when an unknown command or memory error occurs. Unlike I²C, an asynchronous serial error response is a separate response message starting with an ASCII 'E' (0x45). Bytes 2 and 3 of the 5-byte response are the ASCII representation of the hexadecimal error code data byte, and bytes 4 and 5 make up the ASCII representation of the hexadecimal error data byte. Bytes 2 through 5 are ASCII values between '0' and '9' or between 'a' and 'f'. Refer to *Figure 26 – Asynchronous Serial Error Response Timing Diagram*.

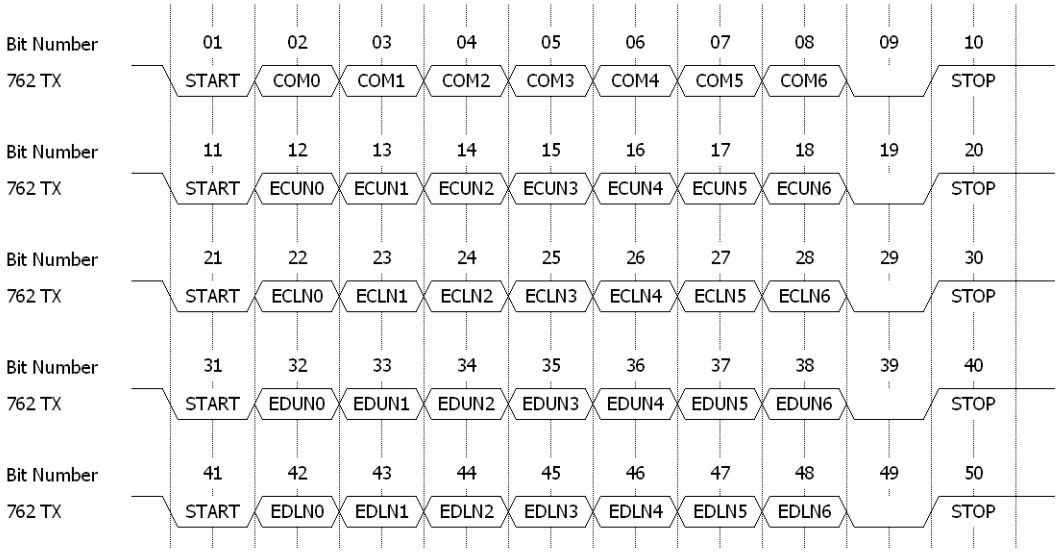

Figure 26 – Asynchronous Serial Error Response Timing Diagram

Both the error code data byte and the error data byte are 8-bits wide, and in the response message are broken down in two nibbles each. The upper nibble of the error code data byte is represented above as ECUN, and the lower nibble is ECLN. Similarly, the upper and lower nibbles of the error data byte are EDUN and EDLN, respectively. The response is terminated with a carriage return.

Error Code	Error Description	Error Data Description
01	Unknown command	Received command
02	Memory error	All zeros

Table 24 – Error Response Descriptions