X76041

Serial Secure E²PROM Target Specification

PURPOSE/SCOPE

Purpose: To provide a target specification for the X76XXY family of Secure Serial EEPROMs. It applies to all X76XXY family members.

Scope: - N/A

REFERENCE DOCUMENTS

General Documents Relating to this Specification:

International Standards: ISO 7816-1; ISO 7816-2; ISO 7816-3.

XICOR Specifications Affected by Changes in this Procedure:

- **Safetv—N/A**
- **Equipment/Materials—N/A**
- **Requirements—N/A**

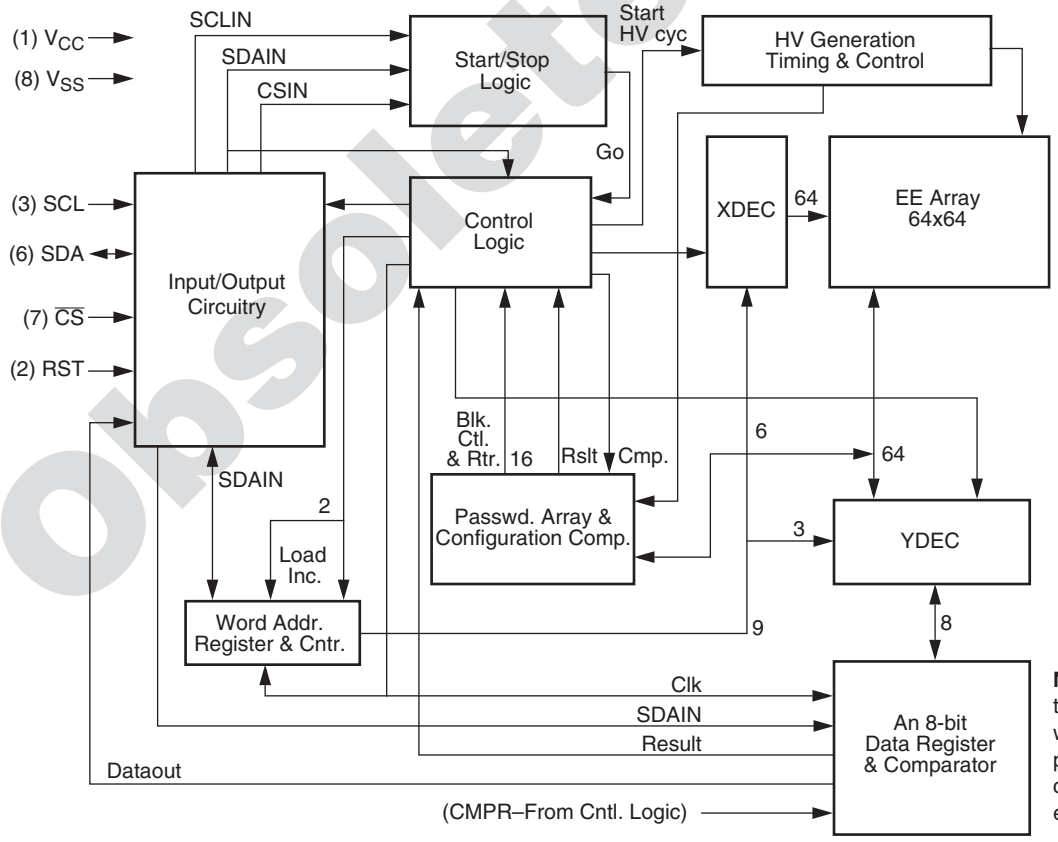
PREPARATION/SETUP

Operation

The X76XXY is a family of secured access serial EEPROMs. On-board non-volatile registers store three 8-byte different passwords which, once set, are used for authentication purposes prior to any read/write operations. The X76XXY uses Two-wire serial interface.

The first part in this family is the X76041, a 4K part with an array of 64 rows by 64 columns (page size of 8 bytes).

BLOCK DIAGRAM



NOTE: There is no way to READ any of the passwords, but there is a possibility to read the configuration registers as explained later.

There are three basic modes of operation with this part:

a) Read—with/without an 8-byte programmable READ password.

b) Write—with/without an 8-byte programmable WRITE password.

c) Configuration—with an 8-byte programmable CONFIGURATION password.

The READ and WRITE operations are similar to current serial EEPROMs and are specified later. The CONFIGURATION option enables the following operations:

a) Program Configuration password

b) Read/Program the Configuration registers

c) Unlimited Read/Write access to the full array

d) Reset of the Read/Write passwords

e) Mass Erase/Program

PIN DESCRIPTIONS

SCL

Serial Clock input.

SDA

Serial data I/O pin.

$\overline{CS}$

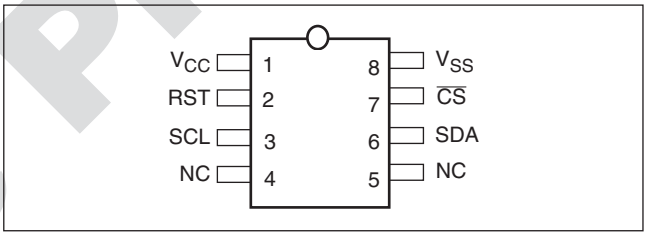
Chip select pin. If this pin is connected to a logic 'O' then communication with the part is enabled and the chip is available for normal serial operation/control. If it is connected to a logic '1', the chip is reset, no communication is possible and it's in stand-by mode.

The state of this pin shouldn't be changed in the middle of a transaction. If the part is deselected in the middle, it will reset and stop unless it's in the middle of a non-volatile write cycle where it will finish the write prior to returning to stand-by mode. The part will not respond to any attempt to reselect it before the end of the write cycle. In addition, if the part is deselected in the middle of any transaction while the clock is low and then reselected while the clock is still low then the protocol will continue normally but in read commands the data that comes out may be corrupted since the output buffer goes to tri-state immediately as the chip is deselected causing undefined values to be shifted into the internal shift register.

RST

A pulse on this input will cause the part internally to output 32 bits of fixed header with respect to 32 clock pulses according to ISO's "synchronous response to reset" (see section 6.5—Timing Diagrams). If the CSN pin isn't low (a logic '0'), no 'response to reset' will occur and if it's changed to a logic '1' in the middle of a 'response to reset', the part will return to a stand-by mode and abort the 'response to reset'. A 'response to reset' will not be available during a non-volatile write cycle. The user should time out for 10ms before issuing a ulse on this pin, if it comes after a non-volatile write cycle. In addition, any attempt to pulse this pin in the middle of a transaction will result with the SDA pin in tri-state. The user will have to issue a stop condition and start the transaction all over again.

PIN CONFIGURATION



This pin configuration is compatible with ISO recommendations with respect to smart cards applications.

DATA SHEET PARAMETERS

DC OPERATING CHARACTERISTICS (Ambient temperature = -55°C to 125°C, V_{CC} = 5v ±10%)

Symbol	Parameter	Limits		Units	Test conditions
		Min.	Max.		
I_{SB}	Stand-by Current The measurement of I_{SB} takes place after a STOP condition that follows a read sequence.		100	µa	SCL = V_{IL}, SDA = High Z, RST = GND, $\overline{CS}$ = V_{CC} - 0.3V
I_{CC1}	V_{CC} write supply current		3	ma	f_{SCL} = V_{IL}/V_{IH}, levels at 1Mhz SDA = V_{IL}/V_{IH}/ High Z, RST = V_{IL}, $\overline{CS}$ = V_{IL}
I_{CC2}	V_{CC} read supply current		2	ma	
I_{LI}	Input leakage current		10	µa	V_{IN} = GND to V_{CC}
I_{LO}	Output leakage current		10	µa	V_{OUT} = GND to V_{CC}
V_{IL}	Input low voltage	-1.0	V_{CC} x 0.3	V	
V_{IH}	Input high voltage	V_{CC} + 0.7	V_{CC} x 0.5		
V_{OL}	Output low voltage		0.4	V	I_{OL} = 2ma
I_{OL}	Output low current		2	ma	0.4v
V_{OH}	Output high voltage	V_{CC} - 0.8		V	I_{OH} = -1ma
I_{OH}	Output high current		-1	ma	V_{CC} - 0.8
C_{IN}	Input capac. RST, SCL, $\overline{CS}$		10	pf	
$C_{I/O}$	Inp/outp capacitance, SDA		10	pf	

AC CHARACTERISTICS

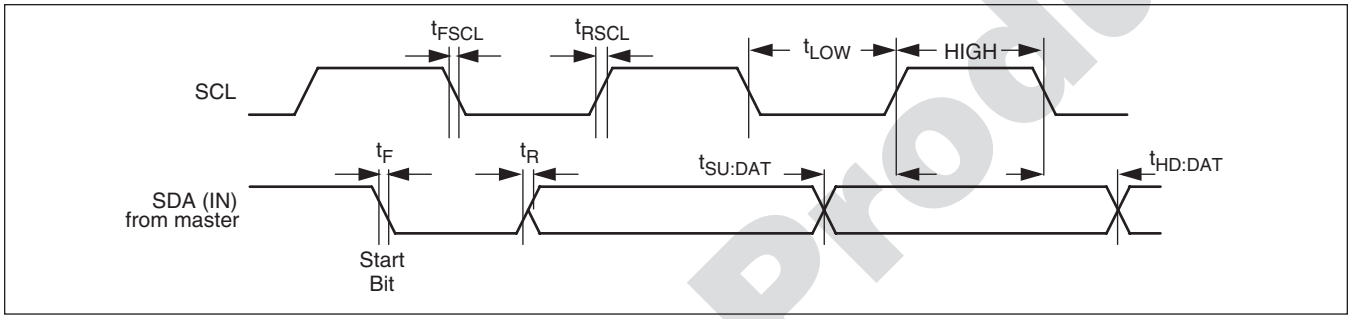
Symbol	Parameter	Min.	Max.	Units
f_{SCL}	SCL clock frequency		1	Mhz
T_I	Noise suppression time constant at SCL & SDA inputs.		20	ns
t_{DV}	SCL high to SDA data out valid		450	ns
t_{LOW}	Clock low period	500		ns
t_{HIGH}	Clock high period	500		ns
t_{STAS1}	Start cond. setup time with respect to RE SCL	150		ns
t_{STAS2}	Start cond. setup time with respect to FE SCL	150		ns
t_{STAH1}	Start cond. hold time with respect to RE SCL	50		ns
t_{STAH2}	Stop cond. hold time with respect to FE SCL	50		ns
t_{STPS1}	Start cond. setup time with respect to RE SCL	150		ns
t_{STPS2}	Stop cond. setup time with respect to FE SCL	50		ns
t_{STPH1}	Stop cond. hold time with respect to RE SCL	150		ns
t_{STPH2}	Stop cond. hold time with respect to FE SCL	50		ns
$t_{HD:DAT}$	Data in hold time	10		ns
$t_{SU:DAT}$	Data in setup time	150		ns
t_{RSCL}	SCL rise time. For slower frequencies, it's a maximum of 9% of the clock period but not more than 500ns. It applies to t_{FSCL} too.		90	ns
t_{FSCL}	SCL fall time		90	ns
t_R	SDA, $\overline{CS}$, RST rise time		90	ns
t_F	SDA, $\overline{CS}$, RST fall time		90	ns
t_{DH}	Data out hold time	0		ns
t_{HZ}	SCL low to High Z output		150	ns
t_{LZ}	SCL high to active output	0		ns
$t_{SU:CS}$	$\overline{CS}$ setup time	200		ns
$t_{HD:CS}$	$\overline{CS}$ hold time	100		ns
$t_{SU:SCL}$	SCL_setup time to CS low after power up	200		ns
t_{RST}	RST high time	50		µs
$t_{SU:RST}$	RST setup time	100		µs
f_{SCL_RST}	SCL frequency during a "synchronous response to reset"		1	Mhz
t_{LOW_RST}	SCL low time during a "synchronous response to reset"	500		ns
t_{HIGH_RST}	SCL high time during a "synchronous response to reset"	500		ns
t_{VCCS}	V_{CC} to $\overline{CS}$ setup time	5		ms
t_{PD}	SCL low to SDA data out valid during 'response to reset'		450	ns
t_{NOL}	REST to SCL non-overlap	500		ns

A.C. TEST CONDITIONS

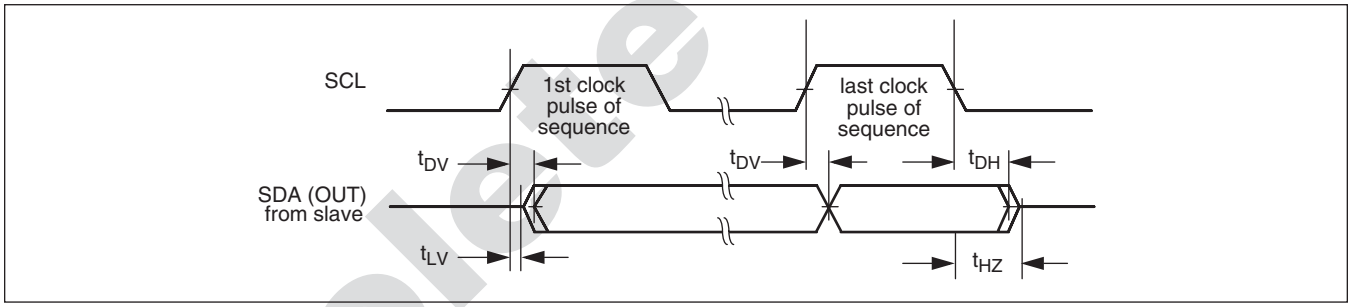
Input pulse levels	V_{CC} x 0.1 to V_{CC} x 0.9
Input rise and fall times	10ns
Input and output timing reference levels	V_{CC} x 0.5
Output Load	1 TTL gate & 100pf

TIMING DIAGRAMS

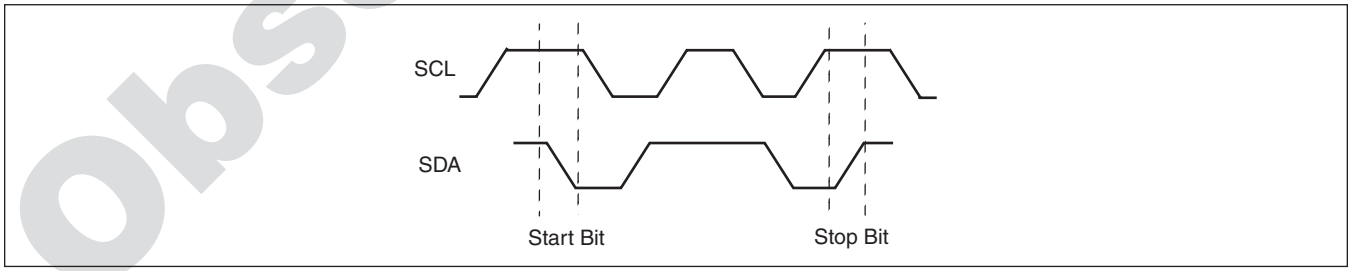
Bus Timing(1)—SDA Driven by the Bus Master



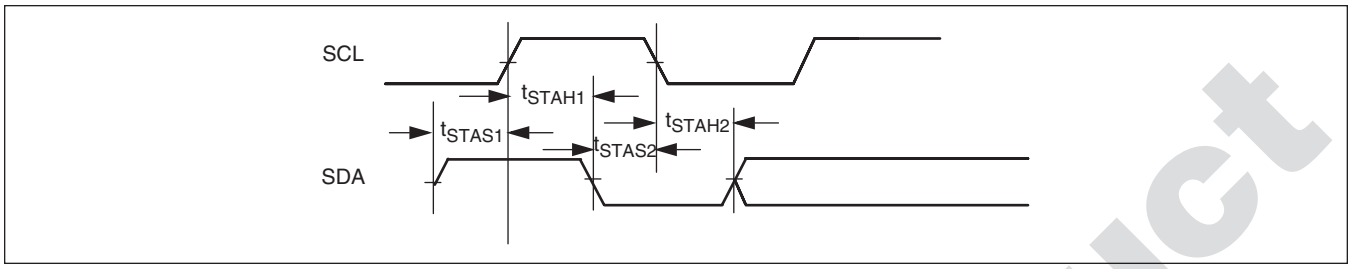
Bus Timing(2)—SDA Driven by the Slave



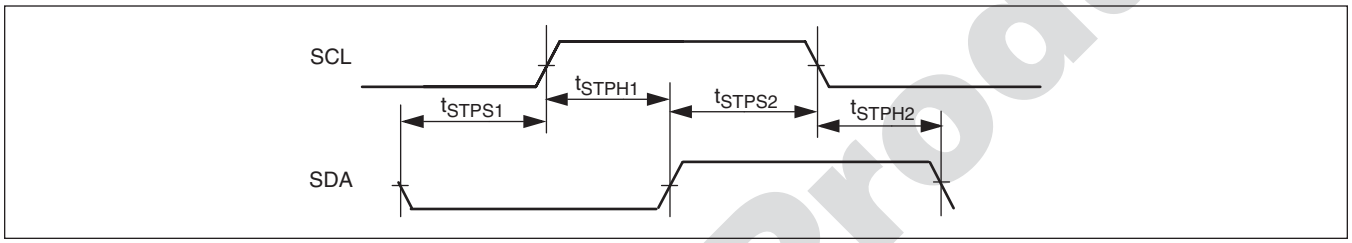
Definition of Start and Stop



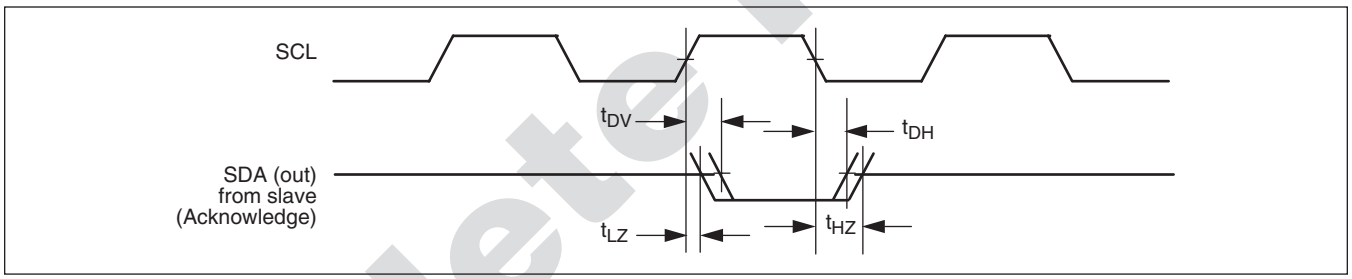
START Condition Timing



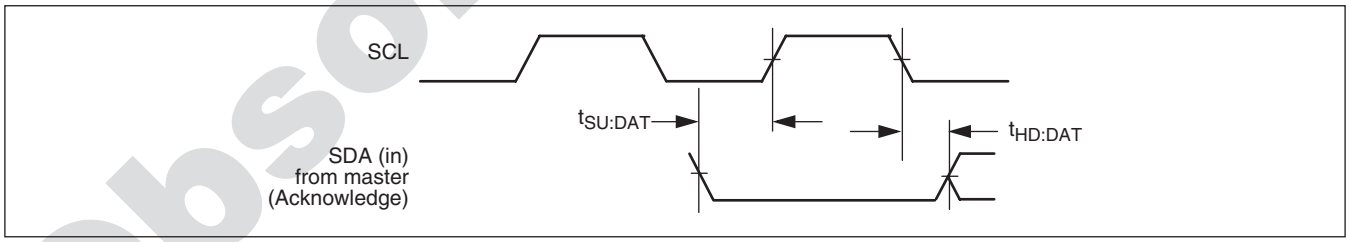
STOP Condition Timing



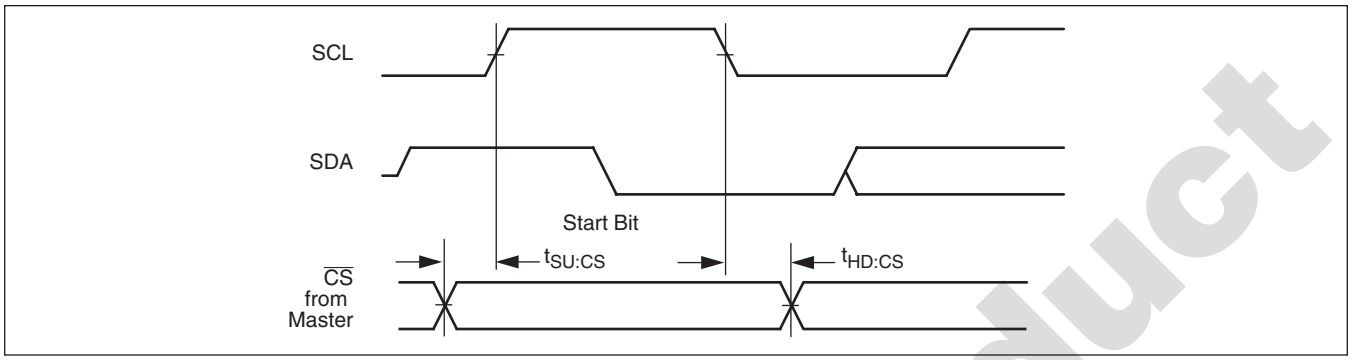
Acknowledge Response from Slave (some timing like data out)



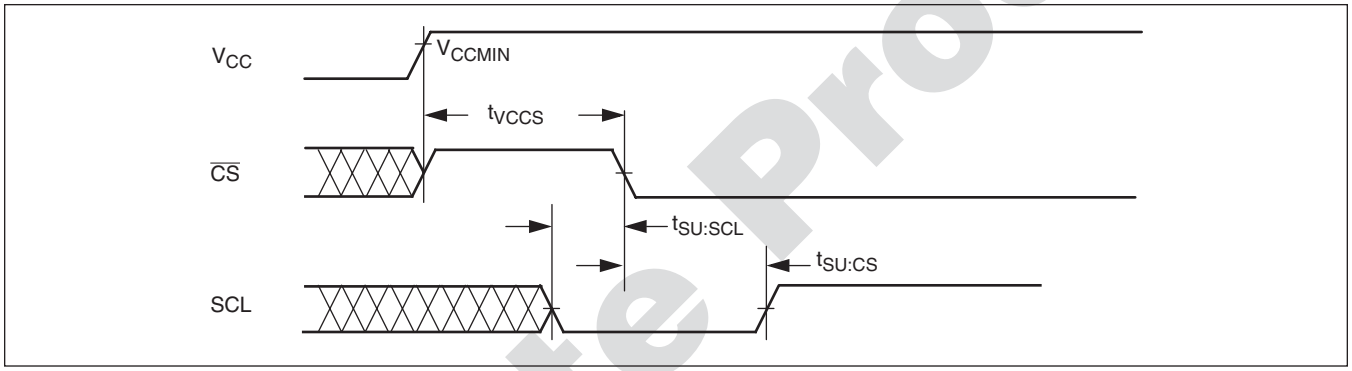
Acknowledge Response from Master (same timing like data in)



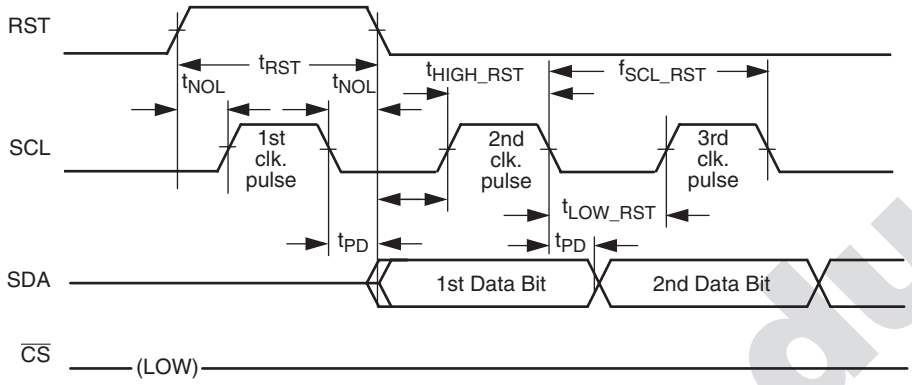
CS Timing Diagram (selecting/deselecting the part)



V_{CC} to $\overline{CS}$ Setup Timing Diagram



RST Timing Diagram—Response to a Synchronous Reset (ISO)



Notes:

* The reset operation results in an answer from the part containing a header transmitted from the part to the master. The header has a fixed length of 32 bits and begins with two mandatory fields of eight bits : H1 and H2.
* The chronological order of transmission of the information bits shall correspond to bit identification bl to b32 with LEAST significant bit transmitted first.
* The first field H1 codes the protocol type. The values of the codes and the corresponding protocol types are :

value (hex)	protocol type
00 and FF	not to be used (reserved)
01 to FE	Each value is assigned by ISO TC97/SC17 to one protocol type.

* The second field codes parameters for the protocol type coded in field H1. The values of H2 are to be assigned by ISO TC97/SC17.
* The remaining fields of the header include 'historical characters' which designate the part manufacturer, the part type (memory size), etc... Since all this information is metal programmable it is T.B.D

The current values that were programmed (metal option) into the device are :

H1: '19' — (Xicor's JEDEC assigned value)

H2: 'AA' — (arbitrarily determined by Xicor; will be changed for each customer)

H3: '55' — (arbitrarily determined by Xicor; will be changed for each customer)

H4: 'AA' — (arbitrarily determined by Xicor; will be changed for each customer)

OPERATIONAL MODES

The protocol for this part is a combination of several sections. The first is a byte that includes three command bits (most significant bits) and five address bits (A12 -:-A8) for extended memory capability (The A8 bit is already used for accessing this “4K” part). The following byte is an “Address/Command” byte which means that it will contain an address or a configuration instruction.

The following section is an 8-byte password that always appears unless accessing a share of the array that requires no password.

The last section is the DATA section which exists only in write or read operation and its length may include one or more bytes.

Each of the bytes is followed by an “ack” signal that is generated by either the master or the slave.

For Example: (write operation with password)

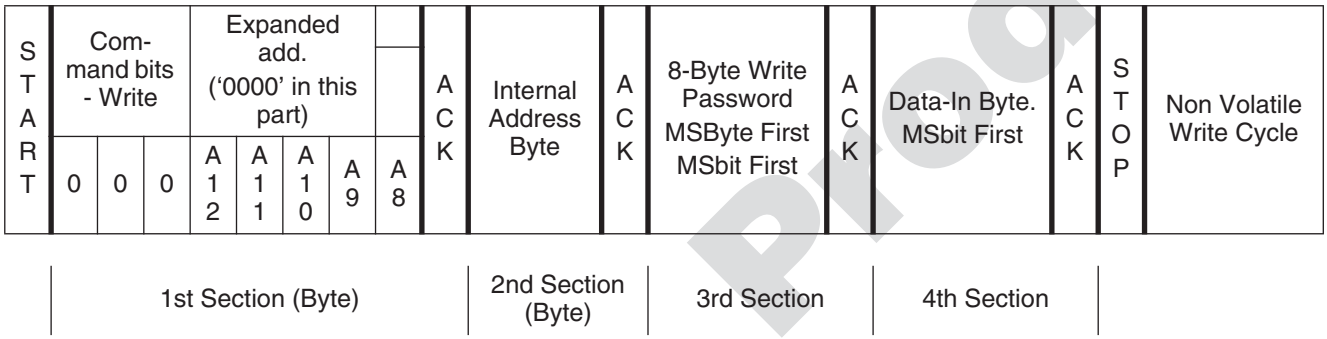


Table 1. For example: (write operation with password)

The following table summarizes the different operational modes that are allowed by this part:

Command Bits	The Second Byte in The Protocol	Command Description	Password Used
0 0 0	Write address	Write (Byte/Page)	Write
0 0 1	Read address	Read (Random/Sequential)	Read
0 1 0	Write address	Write (Byte/Page) to any location	Config.
0 1 1	Read address	Read (Random/Sequential from any location)	Config.
1 0 0	_0_0_0_0_0_0_0_0	Program write-password	Write
1 0 0	_0_0_0_1_0_0_0_0	Program read-password	Read
1 0 0	_0_0_1_0_0_0_0_0	Program configuration-password	Config.
1 0 0	_0_0_1_1_0_0_0_0	Reset write password (all 0’s)	Config.
1 0 0	_0_1_0_0_0_0_0_0	Reset read password (all 0’s)	Config.
1 0 0	_0_1_0_1_0_0_0_0	Program Configuration registers	Config.
1 0 0	_0_1_1_0_0_0_0_0	Read Configuration registers	Config.
1 0 0	_0_1_1_1_0_0_0_0	Mass program	Config.
1 0 0	_1_0_0_0_0_0_0_0	Mass erase	Config.
All the rest		Reserved	

Write Operations

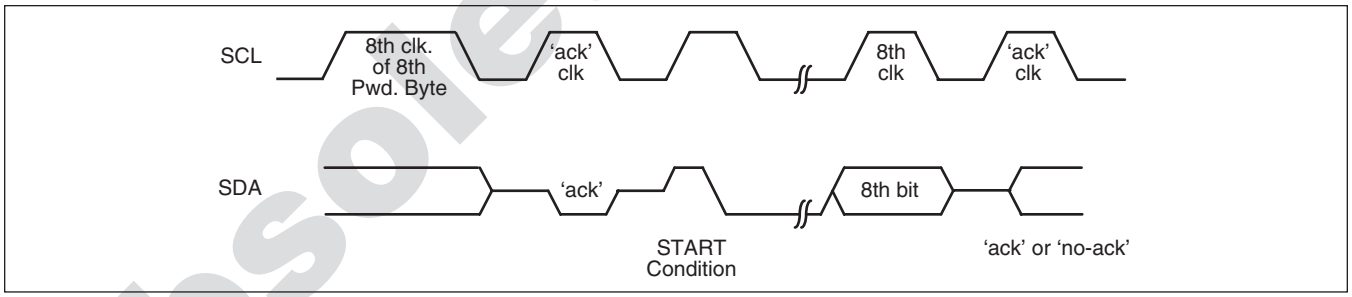
BYTE WRITE—If the device is indeed addressed an acknowledge will be issued and the next section of the protocol is an 8-bit address that enables access to one of the 256 bytes in this half of the memory. After another acknowledge is issued by the slave, the master will send an 8-byte password that will enable the requested write operation. Only then the data-in byte will be sent. The part will acknowledge and the master will terminate the transaction by issuing the STOP condition. The part is also capable of performing WRITE operations without a password, depending upon the value of the appropriate configuration bits that relate to this '1K' share of the array.

START	Command bits - Write			Expanded add. ('0000' in this part)					ACK	Internal Address Byte	ACK	8-Byte Write Password MSByte First MSbit First	ACK	Data-In Byte. MSbit First	ACK	STOP	Non Volatile Write Cycle	(1)
	0	0	0	A12	A11	A10	A9	A8										

Notes: (1) An 'ack' polling must be performed at this point in order to continue the protocol.

General Note:

* After each of the bytes in the password, an ACK signal should be received from the part. The comparison between the received password and the programmed password will be done once all the 64 bits are inside the part and prior to the non-volatile write cycle that will follow.
* The comparison will result, in the case of a wrong password, in a current pulse whose amplitude and position in time is independent of the number of correct/wrong bits in the password that was inserted. An internal noise generator will be activated at this time in order to mask this pulse, regardless of the result of the comparison.
  Following the 'ack' for the 8th byte of the password, the part will execute an internal non-volatile write cycle (which may be a dummy one in case the password was correct) regardless of the comparison result. The user must perform an 'ack' polling routine to determine the validity of the password and invoke the continuation of the required operation. The user may time out for 10ms and then issue the 'ack' polling routine once in order to continue the required operation.
* The following conceptual timing diagram describes this 'Ack' polling scheme:



In the 'ack' polling routine, the user issues a START condition followed by the value 'CO' (hex) or : 11000000 in binary coding and an 'ack' clock. As long as the non-volatile write cycle goes on, the response will be a 'no-ack'. If the password was correct, an 'ack' response will come out following the end of the nonvolatile write cycle. If the password was incorrect, a 'no-ack' response will come out in every 'ack' clock. The user will be able to determine that the password was incorrect only after 10ms have elapsed.

Following any regular non-volatile write cycle (for example for write commands) the user could try to start another transaction by issuing the first byte of the new transaction and then wait for the 'ack'. A 'no-ack' in this case may indicate that the non-volatile write cycle is still on and a new attempt can be performed. However, the user shouldn't use the illegal code 'CO' or any other illegal code because it will be rejected (receive 'no-ack'). The illegal code 'CO' is used only in the polling routine that follows the password insertion.

Page Write

It is initiated in the same manner as the Byte Write operation; however instead of terminating the write cycle after the first data byte is transferred, the master can transmit up to seven more bytes. After the receipt

of each byte, the part will respond with an acknowledge. If more than 8 bytes are transmitted, the internal address counter will "roll over" and the previously written data will be overwritten on the same page.

Read Operations

Read operations are initiated in the same manner as write operations but with a different code. It means that any read operation must begin with an address. If more than one byte is read, then the master may either issue an "ack" for the next successive byte or issue another START condition with another address for a non-successive byte. In other words, the two basic read modes (random & sequential) that are familiar from the two-line interface are available also in this protocol:

Random Read

Every read operation must BEGIN with an address, just like write operations. The last accessed address is kept internally and successive bytes may be read each time from the respective "n+1" address.

START	Command bits - Read			Expanded add. ('0000' in this part)					ACK	Internal Address Byte A7→A0	ACK	8-Byte Read Password MSByte First MSbit First	ACK	Data-Out Byte. MSbit First	STOP	
	0	0	1	A12	A11	A10	A9	A8								(1)

Notes: (1) An 'ack' polling must be performed at this point in order to continue the protocol.

If the user attempts to address an access-limited block of the array the "ack" which follows the address byte will not be issued, letting the bus master know that an address-space violation had occurred. (General note.)

If the user wishes to continue reading additional bytes, successive or random, it can be done without reissuing of the READ password, as long as the STOP condition is not activated. The way to do it, is as follows:

An example of reading Two successive bytes (with the READ password)

START	Command bits—Read			Expanded add. ('0000' in this part)					ACK	Internal Address Byte A7→A0	ACK	8-Byte Read Password MSByte First MSbit First	ACK	Data-Out Byte. MSbit First	ACK	Next Data-Out Byte. MSbit First	STOP	
	0	0	1	A12	A11	A10	A9	A8										(1)

An example of reading Two non-successive bytes (with the READ password)

START	Command bits—Read			Expanded add. ('0000' in this part)					ACK	Internal Address Byte A7→A0	ACK	8-Byte Read Password MSByte First MSbit First	ACK	Data-Out Byte. MSbit First	START	Internal Address Byte A7→A0	ACK	Data-Out Byte. MSbit First	STOP	
	0	0	1	A12	A11	A10	A9	A8												(1)

Notes: (1) An 'ack' polling must be performed at this point in order to continue the protocol.

* In this example, a START condition was issued once again to indicate new random address access. The following byte specifies the new address. (limited to '1K' block that is already pointed at by the 'A8' & 'A7' bits in the very first byte). Therefore, the 'A7' bit that is given in the new address is actually a don't care bit.

* As with the previous example, this mode may be expanded to any number of bytes, within the same '1K' block of the array.

Sequential Read

Sequential read is already accommodated into the normal random read that was previously explained; however it can't go beyond a "1K" block boundary. In other words, the master responds with an acknowledge, as long as it needs additional successive data bytes. The address will be incremented internally and the part will "roll over" to address "0" of the "1K" block boundary if the internal A6-A0 addresses equal "127." The part will CONTINUE to output data for every "ack" that is sent by the master. Thus, the sequential read is limited to the "1K" share of the array from which the first data byte was read.

Configuration Operations

Configuration operations are decoded in the same way like non-configuration operations. In most cases, the MSbit of the command bits is "1" for configuration operations but not always. Please refer to the codes' table which describes each code that is used by this part.

Configuration (Master) Write

This mode allows write access to all the array regardless of the configuration registers' contents. Both Byte and Page writes are available in the following format:

S T A R T	Command bits—Master Write			Expanded add. ('0000' in this part)					A C K	Internal Address Byte A7→A0	A C K	8-Byte Configure Password MSByte First MSbit First	A C K	Data-In Byte. MSbit First	A C K	S T O P	Non Volatile Write Cycle	(1)
	0	1	0	A12	A11	A10	A9	A8										

Notes: (1) An 'ack' polling must be performed at this point in order to continue the protocol.
* For page write, the master continues to send data in instead of terminating with a STOP condition after the first data-in byte.

Configuration (Master) Read

Like in the previous case, all the array is readable with the CONFIGURATION password. The same idea that holds for non-configuration reads is also true now;

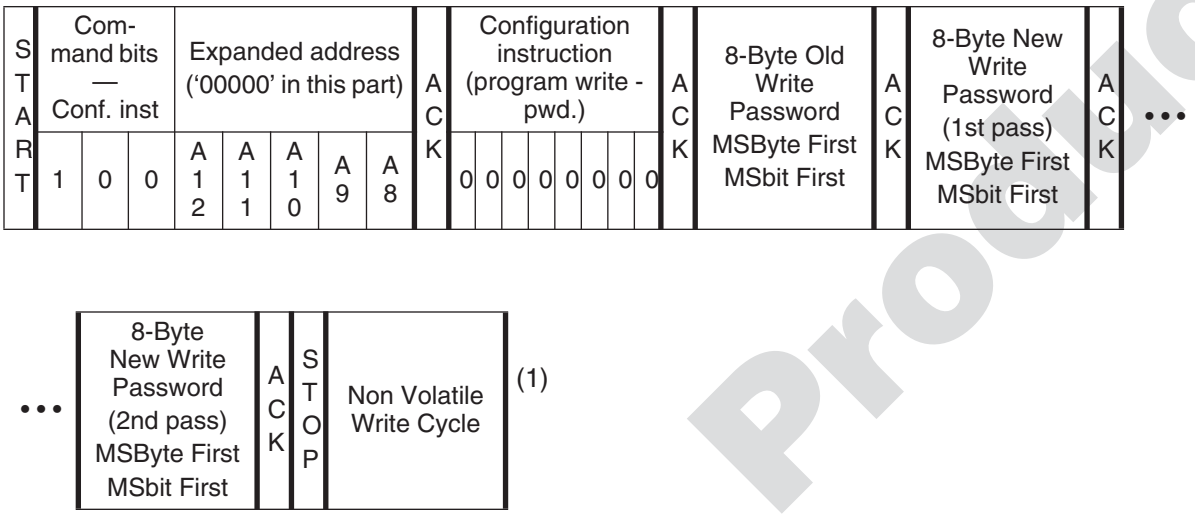
S T A R T	Command bits—Master Read			Expanded add. ('0000' in this part)					A C K	Internal Address Byte A7→A0	A C K	8-Byte Configur. Password MSByte First MSbit First	A C K	Data-Out Byte. MSbit First	S T O P	(1)
	0	1	1	A12	A11	A10	A9	A8								

Notes: (1) An 'ack' polling must be performed at this point in order to continue the protocol.
* As for non-configuration random read, additional bytes from other addresses may be read without entering the configuration password again. Instead of a STOP condition a START condition will be issued followed by an address byte (same '1K' limit for new addresses) and data bytes separated by 'ack' signals.
* Sequential read with the configuration password is performed just like normal sequential-read. Since the configuration password enables access to the whole '4K' array, regardless of the configuration register's value, then a sequential way is actually done in four sessions each enabling a '1K' block.
* In general, the configuration Read/Write operations enable access to any memory location that may otherwise be limited. The configuration password, in this sense, is like a master key that can override the limits caused by the array partitioning.

CONFIGURATION OF PASSWORDS

Program Write-password

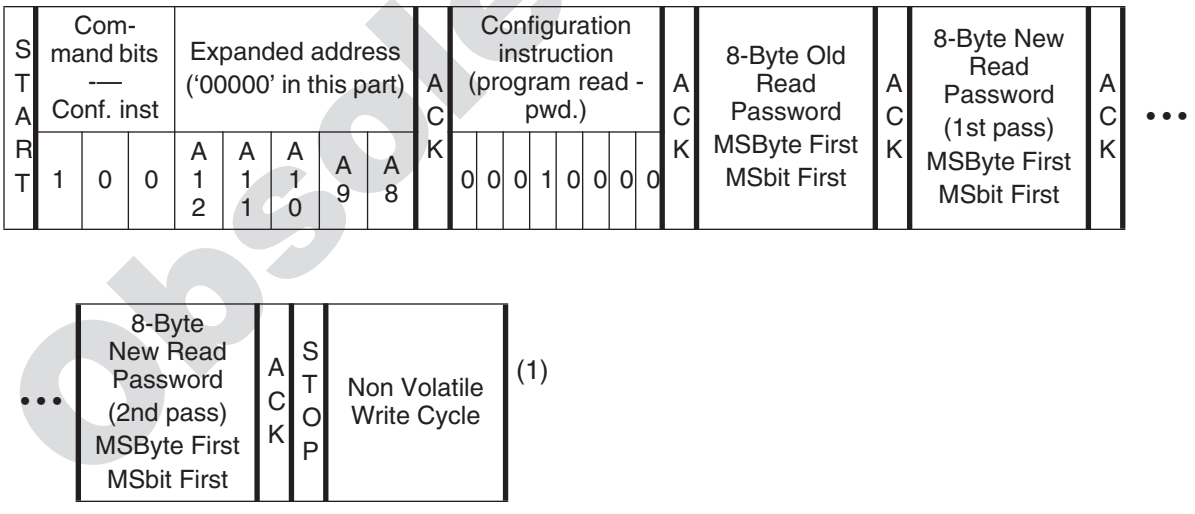
The following sequence will change (program) the write password. Basically, this sequence will repeat itself for Read & Configuration passwords' programming but with a different code.



Notes: (1) An 'ack' polling must be performed at this point in order to continue the protocol.

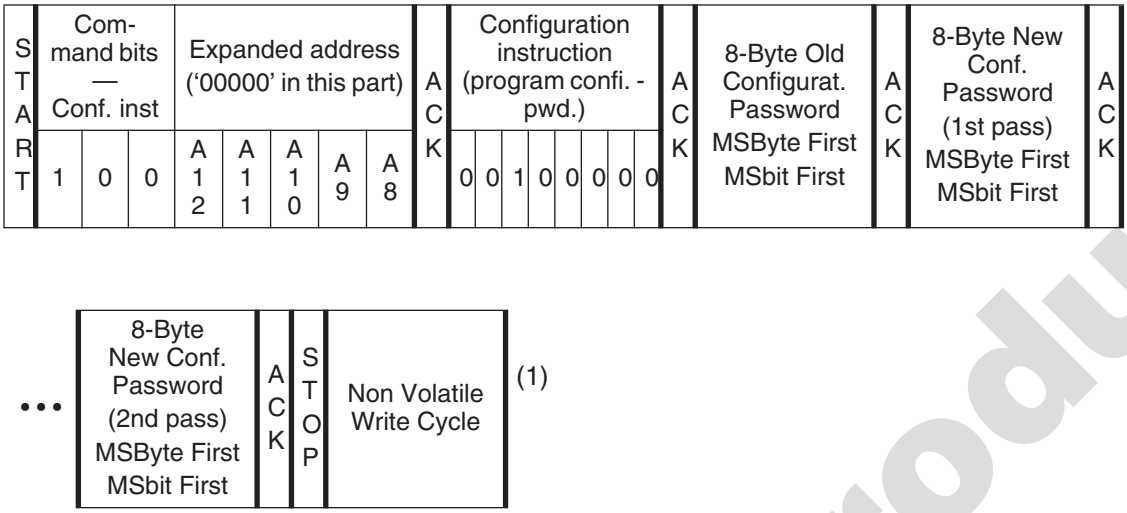
* The programming of passwords is done twice prior to the non-volatile write cycle in order to verify that the user is consistent. In other words, after ALL the Eight bytes are entered in the second pass, a comparison takes place. A mismatch will cause the part to reset and stop (Enter to a stand-by mode) while issuing a 'no-ack' for the master.

Program Read-Password



Notes: (1) An 'ack' polling must be performed at this point in order to continue the protocol.

Program Configuration-Password



Notes: (1) An 'ack' polling must be performed at this point in order to continue the protocol.

Structure of the Special Array

The Special Array consists of four full 64-bi-teach EE rows below the regular 64 rows of the '4k' normal memory. The allocation of the rows is as follows:

64 bit write password

64 bit read password

64 bit configuration password

BCR1	BCR2	CNFG	RETRG	RETCT	reserved (3 EE bytes)

BCR1—Block Control Register 1 (responsible for the lower '2k')

BCR2—Block Control Register 2 (responsible for the upper '2k')

CNFG—Configuration Register (6 bits are used for ABUSE, 16V KILL, RETRY ENABLE & RETRY RESET 2 bits are reserved.)

RETRG—Retry Register (8 bit, allowing up to a maximum of 256 trials)

RETCT—Retry Counter (8 bits, fits the Retry Register)

Note: When performing a 'program configuration' command it will be the user's responsibility to make sure that Retry Register is bigger than the Retry Counter for reasonable device operation. If the user chooses to make the Counter bigger than the Register, the device will treat it as if it was the other way and practically will provide a full 255 wrap-around count.

Configuration of the Block-Control Registers

The block-control registers consists of two EE programmable bytes with four different sections, each of them contains four bits. Each of these sections is responsible for the respective '1K' share of the array. Every '1K' share of the array can be programmed to have the following functionality:

tn n = 0, 1 2, 3	zn n = 0, 1, 2, 3	Function
0	0	Read & Write - unlimited-
0	1	Read only - write limited -
1	0	Program & Read only - Erase limited -
1	1	No Read or Write - Fully limited -

yn n = 0, 1, 2, 3	xn n = 0, 1, 2, 3	Access
0	0	Read pwd. not needed Write pwd. not needed
0	1	Read pwd. not needed Write pwd. needed
1	0	Read pwd. needed Write pwd. not needed
1	1	Read pwd. needed Write pwd. needed

Notes: * The configuration password overrides all functionality limits and virtually allows free access to all the shares of the array.

* If the user wants to erase programmed bits in an address space which is erase limited, the part will reset and stop (enter stand-by mode). The incoming data bytes will be compared, one at a time, with the respective recalled bytes in order to check for that condition. The part will issue a 'no-ack' for a wrong data byte and will automatically reset and stop.

STRUCTURE OF THE BLOCK-CONTROL RESISTERS

1st Block-Control Register (responsible for the lower '2k')

Second '1k'				Lowest '1k'			
x1	y1	z1	t1	x0	y0	z0	t0

MSbit LSbit

2nd Block-Control Resister (responsible for the upper '2k')

Fourth '1k'				Third '1k'			
x3	y3	z3	t3	x2	y2	z2	t2

MSbit LSbit

Array Map

Lowest '1K'	addresses '000' --> 07F' (hex)	High-order addresses
Second '1K'	addresses '080' --> OFF' (hex)	
Third '1K'	addresses '100' --> 17F' (hex)	
Fourth '1K'	addresses '180' --> 1FF' (hex)	

STRUCTURE OF THE CONFIGURATION REGISTER

The configuration register consists of several fields as follows:

Abuse Bits	16v Kill its	Retct Reset	Retct Enable	Reserved (2 Bits)

Notes:

Abuse bits:

1 0 Access is forbidden if retry register equals the retry counter (provided that the retry counter is enabled).

0 1, 0 0, 1 1 Only configuration operations are allowed if the retry register equals the retry counter (provided that the retry counter is enabled).

16v kill bits :

1 0 16v external high voltage testing NOT allowed.

0 1, 0 0, 1 1 16v external high voltage testing allowed

Retry counter reset :

1 Reset the retry counter following a correct password insertion.

0 Don't reset the retry counter following a correct password insertion

Retry counter enable :

1 Retry counter mechanism enabled; An initial comparison between the retry register and retry counter determines whether abuse has been reached; If not, the protocol continues and in case of a wrong password, the retry counter is incremented by '1' and stored in the array for future access reference. If the password was correct then it will either be reset (retry reset bit) or untouched (dummy write cycle)

0 Retry counter mechanism disabled (The result of the comparison between the retry register and the retry counter doesn't affect the continuation of the protocol). The dummy write cycle is done to provide a long delay time for every access, but no non-volatile write to the retry register is performed.

SUMMARY FOR THE RETRY COUNTER MECHANISM

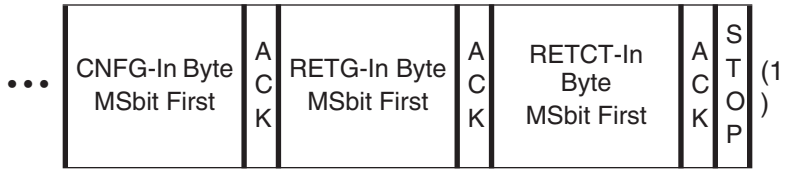
Summary for the Retry Counter Mechanism

Abuse Reached & the Abuse Bits	Passwd Correct	Reset Funct.	Retry Ena.	HV Cycle Type/Comments
Don't allow any access	0	0	0	Entry allowed; Dummy HV(retry disabled)
Don't allow any access	0	0	1	Entry NOT allowed; Bail out after 1st byte
Don't allow any access	0	1	0	Entry allowed; Dummy HV(retry disabled)
Don't allow any access	0	1	1	Entry NOT allowed; Bail out after 1st byte
Don't allow any access	1	0	0	Entry allowed; Dummy HV(retry disabled)
Don't allow any access	1	0	1	Entry NOT allowed; Bail out after 1st byte
Don't allow any access	1	1	0	Entry allowed; Dummy HV(retry disabled)
Don't allow any access	1	1	1	Entry NOT allowed; Bail out after 1st byte
Allow cnfg. op.	0	0	0	Entry allowed; Dummy HV(retry disabled)
Allow cnfg. op.	0	0	1	Entry allowed for cnfg. op. Dummy HV to prevent increment of the retry counter. If NOT abused -> a Real HV cycle
Allow cnfg. op.	0	1	0	Entry allowed; Dummy HV(retry disabled)
Allow cnfg. op.	0	1	1	Entry allowed for cnfg. op. Dummy HV to prevent increment of the retry counter. If NOT abused -> a Real HV cycle
Allow cnfg. op.	1	0	0	Entry allowed; Dummy HV(retry disabled)
Allow cnfg. op.	1	0	1	Entry allowed for cnfg. op. Real HV since reset is required. If NOT abused -> a Real HV as well
Allow cnfg. op.	1	1	0	Entry allowed; Dummy HV (retry disabled)
Allow cnfg. op.	1	1	1	Entry allowed for cnfg. op. Real HV since reset is required. If NOT abused -> a Real HV as well

Program Configuration Registers

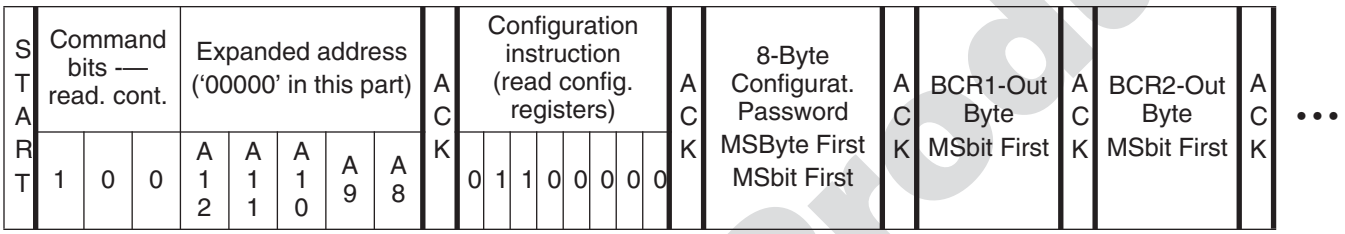
This mode allows programming 5 configuration registers using the Configuration password. It's the user's responsibility to program, among others, the retry counter, and if it is programmed to have a bigger value, then the retry register, there will be a wrap around.

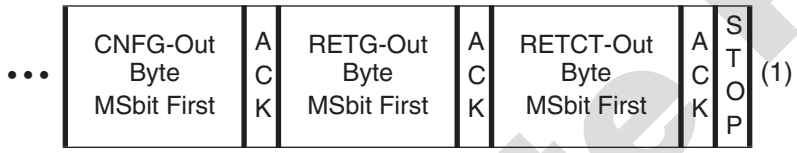
S T A R T	Com-mand bits — read. cont.			Expanded address ('00000' in this part)					A C K	Configuration instruction (program config. registers)								A C K	8-Byte Configurat. Password MSByte First MSbit First	A C K	BCR1-In Byte MSbit First	A C K	BCR2-In Byte MSbit First	A C K	• • •
	1	0	0	A 12	A 11	A 10	A 9	A 8		0	1	0	1	0	0	0	0								



Read Configuration Resister

This mode allows reading of the retry counter with the Configuration password. It may be useful for monitoring purposes.

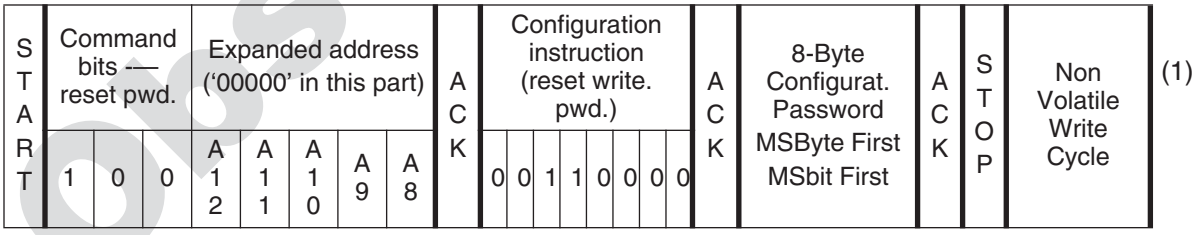




Notes: (1) An 'ack' polling must be performed at this point in order to continue the protocol.
* The user may quit reading by issuing a STOP condition after any byte.

Write-Password Reset

This mode allows the master to reset the WRITE password to all 0's. This is needed if the user wants to re-program the WRITE password and doesn't know the OLD one. This mode will allow the user to begin from a known state. (All 0's).



Notes: (1) An 'ack' polling must be performed at this point in order to continue the protocol.

Read-Password Reset

This mode allows the master to reset the READ password to all 0's. This is needed if the user wants to re-program the READ password and doesn't know the OLD one. This mode will allow the user to begin from a known state. (All 0's).

START	Command bits — reset pwd.			Expanded address ('00000' in this part)					ACK	Configuration instruction (reset read. pwd.)								ACK	8-Byte Configurat. Password MSByte First MSbit First	ACK	STOP	Non Volatile Write Cycle	(1)
	1	0	0	A12	A11	A10	A9	A8		0	1	0	0	0	0	0	0					

Notes: (1) An 'ack' polling must be performed at this point in order to continue the protocol.

MASS Program

This mode allows mass program through a special configuration command. No external High Voltage is needed.

START	Command bits — enable tst			Expanded address ('00000' in this part)					ACK	Configuration instruction (mass program)								ACK	8-Byte Configurat. Password MSByte First MSbit First	ACK	STOP	(1)
	1	0	0	A12	A11	A10	A9	A8		0	1	1	1	0	0	0	0					

Notes: (1) An 'ack' polling must be performed at this point in order to continue the protocol.

MASS Erase

This mode allows mass erase through a special configuration command. No external high voltage is needed.

START	Command bits -— dis. tst			Expanded address ('00000' in this part)					ACK	Configuration instruction (mass erase)								ACK	8-Byte Configurat. Password MSByte First MSbit First	ACK	STOP	(1)
	1	0	0	A12	A11	A10	A9	A8		1	0	0	0	0	0	0	0					

(1) An 'ack' polling must be performed at this point in order to continue the protocol.

Additional General Notes

a) There is no way to read the Read/Write/ Configuration passwords.

b) After each transaction is completed, the part will stop and reset (Enter a stand-by mode). This will also be the response for every attempt to access any limited block of the array. 'Ack' polling during normal non-volatile write cycle is like the Two-wire serial interface parts: As long as the non-volatile write cycle is on, a no-'ack' response will be sent by the part.

c) The master may issue a STOP condition at any given time in which it is driving the SDA line. In other words, when the part is sending 'ack' or data the master may NOT issue a STOP condition. The part will not respond to any such attempt which also causes bus contention. At any other time, a STOP condition will cause the part to reset and stop (enter a stand-by mode). Write operations will terminate prior to entering the stand-by mode.

d) The retry counter will count and store any illegal attempt, no matter which of the Three passwords is used.

e) When the part drives the SDA line, it will tri-state the bus only after the last bit of the sequence. In other words, after the 8th bit of a byte that is read or after 'ack' between incoming bytes. In all other cases when the part drives the bus (between successive bits) it will continue to drive the bus also during the clock LOW periods.

f) Illegal command codes will be disregarded. The part will not respond and will return to a stand-by mode.

g) The part requires the SCL input to be LOW during non-active periods of operation. In other words, the SCL will need to be LOW prior to any START condition and LOW after a STOP condition. This is also reflected in the timing diagrams of section 6.5.

In addition, the SDA line is not allowed to be floating if the SCL line is active high. In other words, SCL must be active LOW when SDA is floating (in tri-state). Thus, false START conditions will be avoided prior to any beginning of a transaction. Moreover, in this protocol SCL is not allowed to be floating during non-active periods for similar reasons.

h) If the abuse-action bits are '00', '01', or '11', then if the Retry Counter Has reached the value of the Retry Register, only configuration operations are allowed. However, since we don't want to create a situation when the Retry Counter becomes bigger than the Retry register, we must make sure that the Retry Counter is not incremented anymore, even if the master uses a wrong configuration password. The reason for preventing it, is the structure of the comparator between the two. The conclusion is that once these conditions are met, the master can endlessly try to enter the part with wrong configuration passwords and avoid the Retry Counter.

i) The part will be delivered to customers at the mass program state. This means that the Retry Register and the Retry Counter will be 'O' but since the retry mechanism will also be disabled (retry enable = '0'), the part can be accessed in full without any password requirements. The user should configure the part using the 'program configuration registers' command, for which it will have to issue the configuration password that is all '0s'.

TEST MODES

There will be five (5) test modes in this part:

a) Mass erase (for cycling purposes for the full array including the special array).

b) Mass program (for cycling purposes for the full array including the special array)

c) Tunneling oxide stress-test with fast ramp-up during mass erase (by disconnecting the ramp-up limiter, the tunneling oxide will go through the tunneling period in a much shorter time).

d) Tunneling oxide stress-test with fast ramp-up during mass program (by disconnecting the ramp-up limiter, the tunneling oxide will go through the tunneling period in a much shorter time).

e) Polyl stress test (for reverse tunneling susceptibility check).

A test mode routine will begin by asserting an external high voltage of 16v DC on the CS pin followed by a START condition and a command byte (much like the regular protocol). Following an 'ack' clock, the part will wait for a STOP condition to begin a non-volatile write cycle in cases a) -> d) or automatically reset and stop in case e).

Note that test modes require no password, however the user may configure its part in a way that will prevent any test modes attempts. The way to do it is by programming that 16v kill bits in the configuration register to '10' as explained previously. This feature enables a 'clean' alternative for the fuse-blowing techniques.

The following timing diagrams show what code is required for each test mode:

a) mass erase

START	Command bits — mass erase				(Don't Care, '0000' in this part)				ACK	STOP
	0	1	1	1	0	0	0	0		

b) mass program

TABLE 2.

START	Command bits — mass program				(Don't Care, '0000' in this part)				ACK	STOP
	0	0	0	0	0	0	0	0		

c) stress test w/mass erase

START	Command bits - stress test w/m. era.				(Don't Care, '0000' in this part)				ACK	STOP
	0	0	1	0	0	0	0	0		

d) stress test w/mass program

START	Command bits — stress test w/m. prg.				(Don't Care, '0000' in this part)				ACK	STOP
	0	1	0	0	0	0	0	0		

e) poly 1 stress test

START	Command bits — poly 1 stress test				(Don't Care, '0000' in this part)				ACK	STOP
	0	0	0	1	0	0	0	0		

COST ANALYSIS

X76041P Cost Calculation (for a packaged part)

Assumptions:

1) Die size estimate	=	131 mil-square (121 x 141)
2) Net die per wafer	=	1119 d/w (based on 80% yield, 6" wafer)
3) Assembly yield	=	98%
4) Test time	=	15 seconds
5) Test yield	=	90%
6) EQA yield	=	99%
7) Visual yield	=	100%
8) PQA yield	=	100%
Die Cost	=	(Wafer fab cost + wafer sort cost)/Net die per wafer
	=	($900 + $35) /1119
	=	$0.84
Assembled Unit	=	For 8-pin Plastic DIP
	=	(Die cost + assembly cost)/ Assembly yield
	=	($0.84 + $0.1)/0.98
	=	$0.94/0.98
Cost after assembly	=	$1.12
Test	=	$0.15 per unit
	=	($0.96 + $0.15)/0.90
	=	$1.11/0.90
Cost after test	=	$1.23
EQA	=	$0.035 per unit
	=	($1.265)/0.99
Cost after EQA	=	$1.28
Visual inspection	=	$0.06 per unit
	=	($1.34)/1.00
Cost after Visual	=	$1.34
PQA inspection	=	$0.01 per unit
	=	($1.35)/1.00
Cost after PQA	=	$1.35
Product Cost	=	$1.35

7. Procedure—N/A

8. Shutdown—N/A

9. Maintenance/Calibration—N/A

10. Supplementarv Information—N/A

11. Operating Summarv—N/A

12. Quality and Reliability Considerations—N/A

CHANGE SHEET

TITLE: X76041 SERIAL SECURE E^{2}PROM TARGET SPECIFICATION

KEY WORDS: X76041, serial, secure, E^{2}PROM, target

REV. NO.	REV. DATE	REV. AUTHOR	DESCRIPTION OF REVISION
0	10/22/90	G. Peer	Incorporate ECN 51796.
01	01/04/91	G. Peer	Incorporate ECN 52863.
02	01/15/92	G. Peer	Incorporate SCN 54320.
03	07/31/92	G. Peer	Incorporate SCN 57483.
04	09/24/92	G. Peer	Incorporate SCN 57957.
05	11/23/92	G. Peer	Incorporate SCN 58320.

SCN's

SCN. NO.	EFFECTIVE PERIOD FROM	TO	DESCRIPTION OF SCN
51796	04/18/90		New spec.
52863	10/24/90		Change spec no. from 011124CS04 to 011125C24. Change title from X24CS04 Target Specification to X25C24 Serial Secure E^{2}PROM Target Specification. New timing diagram. Data sheet parameters (AC characteristics) change New device number: X25C24 (to be replaced wherever it appears in rough draft).
54320	11/19/91		Change spec no. from 011125C24 to 011176041. Change title from "X25C24 SERIAL SECURE E^{2}PROM TARGET SPECIFICATION" to "X76041 SERIAL SECURE E^{2}PROM TARGET SPECIFICATION". Revised ISO "Response to Reset". AC characteristics change. New set of commands. New test modes.
57483	07/13/92		Definition of expected response to reset; correction of errors in DC test conditions; explanation of the "ack" polling routine after a password sequence; revised cost calculation; revised timing diagram for CS.
57957	08/26/92		Correction of errors; specification of ISO documents (standards for smart cards); missing word in the RST pin description; wrong functional description of the read only mode in the block-control registers.
58320	11/04/92		Correction of errors in response-to-reset timing diagram and in the tables that describe the configuration of the block-control registers.