

Interfacing the X24C44, X24C45 NOVRAMs to the Motorola 6803 Microcontroller

Application Note July 15, 2005 AN25.0

Author: Applications Staff

The following code demonstrates how the Intersil X24C44, X24C45 serial NOVRAMs can be interfaced to the Motorola 6803 microcontroller when connected as shown in Figure 1. The code uses three pins from port 1 to implement the

interface. Additional code can be found on the Intersil web site at http://www.intersil.com that will implement interfaces between several other Motorola microcontroller families and most Intersil serial devices.

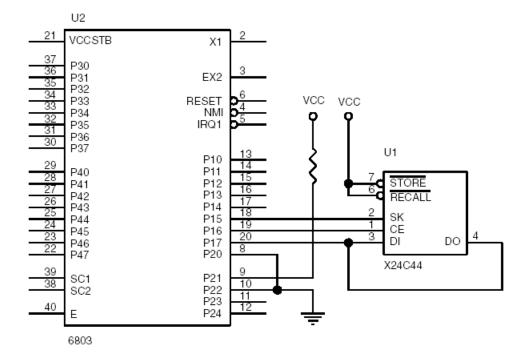


FIGURE 1. INTERFACING AN X24C44 TO A 6803 MICROCONTROLLER

```
*******************
* THIS CODE WAS DESIGNED TO DEMONSTRATE HOW THE X24C44 COULD BE INTERFACED TO *
* THE 6803 MICROCONTROLLER. THE INTERFACE USES 3 LINES FROM PORT 1 (P17,
* P16, AND P15) TO COMMUNICATE. THE DI AND DO PINS ON THE X24C44 ARE TIED
 TOGETHER WHICH ALLOWS 1 LESS PORT LINE TO BE USED.
* THE CODE SHOWN DEMONSTRATES RCL, WREN, READ, WRITE, AND STORE
* INSTRUCTIONS. THE REMAINING INSTRUCTIONS (WRDS AND ENAS) CAN BE ISSUED
* USING THE SAME ROUTINE AS OTHER NON-DATA INSTRUCTIONS.
* THE PROGRAM ISSUES A SEQUENCE OF INSTRUCTIONS TO READ THE CONTENTS OF
 ADDRESS 5 AND STORES THE SAME VALUE IN ADDRESS 9. THE SEQUENCE OF
 INSTRUCTIONS IS AS FOLLOWS :
* 1. RCL SETS THE PREVIOUS RECALL LATCH
* 2. WREN SETS THE WRITE ENABLE LATCH
* 3. READ DATA FROM ADDRESS 5 IS READ
* 4. WRITE THE DATA READ DURING STEP 3 IS WRITTEN TO ADDRESS 9
* 5. STO THE RAM'S CONTENTS IS TRANSFERED TO THE EEPROM
* DATA TRANSFER IS PERFORMED WITH THE MOST SIGNIFICANT BIT FIRST. DURING
* THE READ AND WRITE INSTRUCTIONS THE DATA SEQUENCE IS INVERTED FROM THAT
* SHOWN IN THE DATA BOOK (D15 IS SHIFTED FIRST).
*************************
SKHI
      EOU
               $20
                           MASK TO GENERATE A 1 ON SK
SKLO
      EOU
               $DF
                           MASK TO GENERATE A 0 ON SK
DTHT
      EOU
               $80
                           MASK TO GENERATE A 1 ON DI
DILO
      EQU
              $7F
                           MASK TO GENERATE A 0 ON DI
               $40
                           MASK TO GENERATE A 1 ON CE
CEHI
      EOU
CELO
      EQU
              $BF
                           MASK TO GENERATE A 0 ON CE
WRDS
      EOU
              $80
                           RESET WRITE ENABLE LATCH
STO
              $81
                           TRANSFERS FROM RAM TO EEPROM
      EOU
ENAS
      EOU
              $82
                           PLACES PART INTO POWER DOWN MODE
WRITE
      EQU
              $83
                           RAM WRITE
WREN
               $84
                           SET WRITE ENABLE LATCH
RCL
      EQU
               $85
                           TRANSFERS FROM EEPROM TO RAM, RESETS
                           WRITE ENABLE LATCH
      EOU
             $86
READ
                           RAM READ
DDR 1
      EQU
               $00
                           DATA DIRECTION REGISTER FOR PORT 1
PORT1
     EQU
               $02
                           ADDRESS FOR PORT 1
               $80
                           LOCATION FOR X24C44 ADDRESS TO ACCESS
ADDR
      EOU
INST
      EQU
               $81
                           INSTRUCTION FOR PART
RWDAT
               $82
                          LOCATION FOR X24C44 DATA TRANSFERED
      EOU
               $84
                           DATA TO BE SENT TO DUT
P1DATA EOU
DD1DAT EQU
                           DATA TO BE STORED IN PORT 1 DIRECTION REGISTER
               $85
* RESET VECTOR TO BEGINNING OF PROGRAM CODE *
************
      ORG
               SFFFE
                         RESET VECTOR TO PROGRAM ENTRY POINT
      FDB
               $E000
********
* START OF PROGRAM EXECUTION *
*******
```

BEGINNING OF EXECUTABLE CODE

ORG

\$E000

intersil July 15, 2005

AN25.0

TAA TAA DAA TAA DAA TAA BR BR BR DAA TAA BR BR BR BR BR BR BR BR BR	DDR1 DDIDAT #\$1F PORT1 P1DATA #RCL INST CEHIGH OUTBYT CELOW #WREN	PORT 1 ALL OUTPUTS INITIALIZE PORT1 DIRECTION REGISTER INITIALIZE PORT1 DIRECTION VALUE CE, SK, DI ALL 0S INITIALIZE PORT1 INITIALIZE PORT1 DATA VALUE PERFORM A RECALL TO SET THE RECALL LATCH PERFORM A WRITE ENABLE TO SET THE WRITE ENABLE LATCH
TAA DAA TAA DAA TAA SR SR SR DAA TAA SR	DD1DAT #\$1F PORT1 P1DATA #RCL INST CEHIGH OUTBYT CELOW #WREN INST	INITIALIZE PORT1 DIRECTION VALUE CE, SK, DI ALL 0S INITIALIZE PORT1 INITIALIZE PORT1 DATA VALUE PERFORM A RECALL TO SET THE RECALL LATCH PERFORM A WRITE ENABLE TO SET
DAA TAA DAA TAA SR SR DAA TAA SR	#\$1F PORT1 P1DATA #RCL INST CEHIGH OUTBYT CELOW #WREN INST	CE, SK, DI ALL 0S INITIALIZE PORT1 INITIALIZE PORT1 DATA VALUE PERFORM A RECALL TO SET THE RECALL LATCH PERFORM A WRITE ENABLE TO SET
TAA TAA DAA TAA SR SR DAA TAA SR SR SR SR SA	PORT1 P1DATA #RCL INST CEHIGH OUTBYT CELOW #WREN INST	INITIALIZE PORT1 INITIALIZE PORT1 DATA VALUE PERFORM A RECALL TO SET THE RECALL LATCH PERFORM A WRITE ENABLE TO SET
TAA DAA TAA ER ER BR DAA TAA ER	P1DATA #RCL INST CEHIGH OUTBYT CELOW #WREN INST	INITIALIZE PORT1 DATA VALUE PERFORM A RECALL TO SET THE RECALL LATCH PERFORM A WRITE ENABLE TO SET
DAA FAA ER ER DAA FAA ER	#RCL INST CEHIGH OUTBYT CELOW #WREN INST	PERFORM A RECALL TO SET THE RECALL LATCH PERFORM A WRITE ENABLE TO SET
TAA SER SER DAA TAA SER	INST CEHIGH OUTBYT CELOW #WREN INST	THE RECALL LATCH PERFORM A WRITE ENABLE TO SET
ER ER ER DAA FAA ER	CEHIGH OUTBYT CELOW #WREN INST	PERFORM A WRITE ENABLE TO SET
ER ER DAA FAA ER ER	OUTBYT CELOW #WREN INST	
SR DAA FAA SR	CELOW #WREN INST	
DAA FAA SR SR	#WREN INST	
FAA SR SR	INST	
SR SR		THE MOTTE ENABLE TATCH
SR	CEHIGH	INE WRITE ENABLE LAICH
	OUTBYT	
SR	CELOW	
DAA	#\$05	READ THE CONTENTS OF ADDRESS 5
raa a	ADDR	THE VALUE READ WILL BE IN STORED
3R	RDWRD	IN RWDATA
DAA	#\$09	WRITE THE DATA JUST READ INTO
raa aa	ADDR	ADDRESS 9
3R	WRWRD	
DAA	#STO	PERFORM A STORE OPERATION
ΓAA	INST	
SR	CEHIGH	
3R	OUTBYT	
3R	CELOW	
AS	*	LOOP UNTIL RESET
TTEN IS S	SPECIFIED IN A	ADDR. *
*****	;***********	*****
JSR	CEHIGH	WRITE VALUE IN RWDATA INTO LOCATION
DAA	ADDR	SPECIFIED IN ADDR
SLA		JUSTIFY ADDRESS IN INSTRUCTION
	#WRITE	MASK IN WRITE INSTRUCTION
	INST	
TAA		
		SEND WRITE INSTRUCTION TO DUT
SR	OUTBYT	SEND WRITE INSTRUCTION TO DUT
SR DAA	OUTBYT RWDAT	SEND WRITE INSTRUCTION TO DUT
SR DAA FAA	OUTBYT RWDAT INST	
SR DAA FAA SR	OUTBYT RWDAT INST OUTBYT	SEND WRITE INSTRUCTION TO DUT SEND IN UPPER BYTE OF DATA
SR DAA FAA SR DAA	OUTBYT RWDAT INST OUTBYT RWDAT+1	
ER DAA FAA BR DAA	OUTBYT RWDAT INST OUTBYT RWDAT+1 INST	SEND IN UPPER BYTE OF DATA
SR DAA FAA SR DAA FAA SR	OUTBYT RWDAT INST OUTBYT RWDAT+1 INST OUTBYT	
SR DAA FAA SR DAA FAA SR	OUTBYT RWDAT INST OUTBYT RWDAT+1 INST	SEND IN UPPER BYTE OF DATA
SR DAA FAA SR DAA FAA SR	OUTBYT RWDAT INST OUTBYT RWDAT+1 INST OUTBYT	SEND IN UPPER BYTE OF DATA
	SR DAA FAA SR DAA FAA SR SR SR SR SR THE WORD TTEN IS S ******** JSR DAA SLA SLA SLA SLA	SR RDWRD DAA #\$09 FAA ADDR SR WRWRD DAA #STO FAA INST SR CEHIGH SR OUTBYT SR CELOW RA * *********************************

intersil

	LSLA LSLA	JUSTIFY	ADDRESS TO READ
	LSLA	#DEAD	MACK IN DEAD INCODUCTION
	ORAA STAA	#READ INST	MASK IN READ INSTRUCTION
	JSR		SEND IN 7 BITS OF READ INSTRUCTION
	LDAA	SEND7	MAKE DATA LINE AN INPUT
	ANDA	DD1DAT	MARE DATA LINE AN INPUT
		#DILO	
	STAA STAA	DDR1	
	JSR	DD1DAT CLOCK	SEND EIGHTH CLOCK PULSE FOR READ INSTRUCTION
	LDX	#\$0010	PREPARE TO SHIFT IN 16 BITS
BITX:		#\$0010	ASSUME BIT IS GOING TO BE A ZERO (CLEAR CARRY)
BIIA.	LDAA	PORT1	READ BIT VALUE
	ANDA	#DIHI	MASK BIT OUT OF BYTE READ
	BEQ	NO1	LEAVE CARRY FLAG ALONE IF BIT IS A 0
	SEC	SET	CARRY IF BIT IS A 1
NO1:	ROL	RWDAT+1	ROLL CARRY FLAG INTO DATA WORD
NO1.	ROL	RWDAT	ROLL CARRI FLAG INTO DATA WORD
	JSR	CLOCK	SEND A CLOCK PULSE
	DEX	CLOCK	LOOP UNTIL
	BNE	BITX	16 BITS ARE READ
	LDAA	DD1DAT	MAKE DATA LINE AN OUTPUT
	ORAA	#DIHI	THE BITT BIND IN COILCI
	STAA	DDR1	
	STAA	DD1DAT	
	JSR	CELOW	
	RTS	0220	
* SEN	D DATA OUT	T TO THE PART	**************************************
* SENI * LOCI ****	D DATA OUT	T TO THE PART	. THE DATA TO BE SENT IS *
* SENI * LOCI ****	D DATA OU' ATED IN II	T TO THE PART	. THE DATA TO BE SENT IS * * *******************************
* SENI * LOCI ****	D DATA OUT ATED IN II ******* : LDX BRA	r to the part NST. ************************************	. THE DATA TO BE SENT IS * * *******************************
* SEND * LOCA ***** SEND7	D DATA OUT ATED IN II ******* : LDX BRA	T TO THE PART NST. ************************************	. THE DATA TO BE SENT IS * * ******************************
* SEND * LOCA ***** SEND7	D DATA OUT ATED IN II ******* : LDX BRA T:LDX	T TO THE PART NST. ********* #\$0007 LOOPO #\$0008	. THE DATA TO BE SENT IS * * ******************************
* SEND * LOCA ***** SEND7	D DATA OUT ATED IN II ******** : LDX BRA T:LDX : LDAB	T TO THE PART NST. ******** #\$0007 LOOPO #\$0008 P1DATA	. THE DATA TO BE SENT IS * * ******************************
* SEND * LOCA ***** SEND7	D DATA OUT ATED IN II ******** : LDX BRA T:LDX : LDAB ANDB	T TO THE PART NST. ********* #\$0007 LOOPO #\$0008 P1DATA #DILO	. THE DATA TO BE SENT IS * * ******************************
* SEND * LOCA ***** SEND7	D DATA OUT ATED IN II ******** : LDX BRA T:LDX : LDAB ANDB ROL	T TO THE PART NST. ********* #\$0007 LOOPO #\$0008 P1DATA #DILO INST	. THE DATA TO BE SENT IS * * **************************** SHIFT OUT 7 BITS FOR READ INSTRUCTION PREPARE TO SHIFT OUT 8 BITS
* SEND * LOCA ***** SEND7	D DATA OUT ATED IN II ********* : LDX BRA T:LDX : LDAB ANDB ROL BCC	T TO THE PART NST. ********* #\$0007 LOOPO #\$0008 P1DATA #DILO INST IS0	. THE DATA TO BE SENT IS * * *************************** SHIFT OUT 7 BITS FOR READ INSTRUCTION PREPARE TO SHIFT OUT 8 BITS JUMP IF DATA SHOULD BE 0
* SEND * LOCA ***** SEND7 OUTBY LOOPO	D DATA OUT ATED IN II ********** : LDX BRA T:LDX : LDAB ANDB ROL BCC ORAB	T TO THE PART NST. ********* #\$0007 LOOPO #\$0008 P1DATA #DILO INST IS0 #DIHI	. THE DATA TO BE SENT IS * * ***************************** SHIFT OUT 7 BITS FOR READ INSTRUCTION PREPARE TO SHIFT OUT 8 BITS JUMP IF DATA SHOULD BE 0 MAKE DATA A 1
* SEND * LOCA ***** SEND7 OUTBY LOOPO	D DATA OUT ATED IN II ********* : LDX BRA T:LDX : LDAB ANDB ROL BCC ORAB STAB	T TO THE PART NST. ********* #\$0007 LOOPO #\$0008 P1DATA #DILO INST ISO #DIHI PORT1	. THE DATA TO BE SENT IS * * ***************************** SHIFT OUT 7 BITS FOR READ INSTRUCTION PREPARE TO SHIFT OUT 8 BITS JUMP IF DATA SHOULD BE 0 MAKE DATA A 1
* SEND * LOCA ***** SEND7 OUTBY LOOPO	D DATA OUT ATED IN II ********** : LDX BRA T:LDX : LDAB ANDB ROL BCC ORAB STAB STAB JSR DEX	T TO THE PART NST. ********* #\$0007 LOOPO #\$0008 P1DATA #DILO INST ISO #DIHI PORT1 P1DATA	. THE DATA TO BE SENT IS * *********************************
* SEND * LOCA ***** SEND7 OUTBY LOOPO	D DATA OUT ATED IN II ********** : LDX BRA T:LDX : LDAB ANDB ROL BCC ORAB STAB STAB JSR	T TO THE PART NST. ********* #\$0007 LOOPO #\$0008 P1DATA #DILO INST ISO #DIHI PORT1 P1DATA	. THE DATA TO BE SENT IS * *********************************
* SEND * LOCA ***** SEND7 OUTBY LOOPO	D DATA OUT ATED IN II ********** : LDX BRA T:LDX : LDAB ANDB ROL BCC ORAB STAB STAB JSR DEX BNE	T TO THE PART NST. ********* #\$0007 LOOPO #\$0008 P1DATA #DILO INST ISO #DIHI PORT1 P1DATA CLOCK LOOPO	. THE DATA TO BE SENT IS * *********************************
* SEND * LOCA ***** SEND7 OUTBY' LOOPO IS0:	D DATA OUT ATED IN II ********** : LDX BRA T:LDX : LDAB ANDB ROL BCC ORAB STAB STAB JSR DEX BNE RTS	T TO THE PART NST. ********** #\$0007 LOOPO #\$0008 P1DATA #DILO INST ISO #DIHI PORT1 P1DATA CLOCK LOOPO	. THE DATA TO BE SENT IS * *********************************
* SEND * LOCA ***** SEND7 OUTBY LOOPO ISO: ****	D DATA OUT ATED IN II ********** : LDX BRA T:LDX : LDAB ANDB ROL BCC ORAB STAB STAB JSR DEX BNE RTS ***********************************	T TO THE PART NST. ********** #\$0007 LOOPO #\$0008 P1DATA #DILO INST ISO #DIHI PORT1 P1DATA CLOCK LOOPO	. THE DATA TO BE SENT IS * *********************************
* SEND * LOCA ***** SEND7 OUTBY LOOPO ISO: ****	D DATA OUT ATED IN II ********* : LDX BRA T:LDX : LDAB ANDB ROL BCC ORAB STAB STAB STAB JSR DEX BNE RTS **********************************	#\$0007 LOOPO #\$0008 P1DATA #DILO INST ISO #DIHI PORT1 P1DATA CLOCK LOOPO	**************************************
* SEND * LOCA ***** SEND7 OUTBY LOOPO ISO: ****	D DATA OUT ATED IN II ********* : LDX BRA T:LDX : LDAB ANDB ROL BCC ORAB STAB STAB STAB JSR DEX BNE RTS ********** H:LDAA	#\$0007 LOOPO #\$0008 P1DATA #DILO INST ISO #DIHI PORT1 P1DATA CLOCK LOOPO	**************************************
* SEND * LOCA ***** SEND7 OUTBY LOOPO ISO: ****	D DATA OUT ATED IN II ********* : LDX BRA T:LDX : LDAB ANDB ROL BCC ORAB STAB STAB STAB JSR DEX BNE RTS ********* H:LDAA ORAA	#\$0007 LOOPO #\$0008 P1DATA #DILO INST ISO #DIHI PORT1 P1DATA CLOCK LOOPO	**************************************
* SEND * LOCA ***** SEND7 OUTBY LOOPO ISO: ****	D DATA OUT ATED IN II ********* : LDX BRA T:LDX : LDAB ANDB ROL BCC ORAB STAB STAB STAB JSR DEX BNE RTS ********* H:LDAA ORAA STAA	#\$0007 LOOPO #\$0008 P1DATA #DILO INST ISO #DIHI PORT1 P1DATA CLOCK LOOPO *** HOPO *** HOPO ** HOPO	**************************************

*****	*****		
* BRIN	G CE LOW *		
*****	*****		
CELOW:	ANDA STAA	P1DATA CELO PORT1 P1DATA	BRING CE LOW
*****	*****	*****	
* ISSU	E A CLOCK	PULSE. *	
*****	*****	*****	
CLOCK:	LDAA	P1DATA	PROVIDE A CLOCK PULSE ON SK
	ORAA	#SKHI	
	STAA	PORT1	BRING SK HIGH
	ANDA	#SKLO	
	STAA	PORT1	BRING SK LOW
	STAA	P1DATA	
	RTS		

Intersil Corporation reserves the right to make changes in circuit design, software and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that the Application Note or Technical Brief is current before proceeding.