

The X24C44 NOVRAM Teams up with 8051 Microcontrollers

by Rick Orlando, January 1990

Introduction

The X24C44 is a 256-bit serial NOVRAM internally configured as sixteen 16-bit words of RAM overlaid bit by bit with a nonvolatile E²PROM. The X24C44 has the standard hardware RECALL and STORE inputs plus the ability to perform these same operations under software control, thereby freeing two microcontroller port pins for other tasks. The serial interface allows the X24C44 to be packaged in a low cost space saving 8-pin mini DIP. When teamed with the 8051 family of microcontrollers (Figure 1), the X24C44's small physical size, software instruction set, and serial interface make it an ideal parameter storage and scratch pad memory, while maintaining full use of the 8051 serial port as a UART.

Scope

This application note describes interfacing the X24C44 with the 8051 family of microcontrollers. Emphasis will

be placed on the timing considerations of the interface and explaining the modifications to the instruction words for normal device operation. This note assumes the reader has access to a Xicor Data Book and an Intel Microcontroller Handbook.

Serial Port Operation

Port 3 on the 8051 provides a serial port that can be used in two basic configurations, full duplex and half duplex. This note examines the half duplex (mode 0) operation when interfacing to the X24C44. Port 3 pin 1 (P3.1) is the clock output for both transmit and receive modes and port 3 pin 0 (P3.0) is used for bidirectional data transfers. The clock output frequency is 1/12 of the XTAL oscillator input frequency. To simplify timing calculations, this note will assume an input frequency of 12MHz resulting in a symmetrical 1MHz output on P3.1.

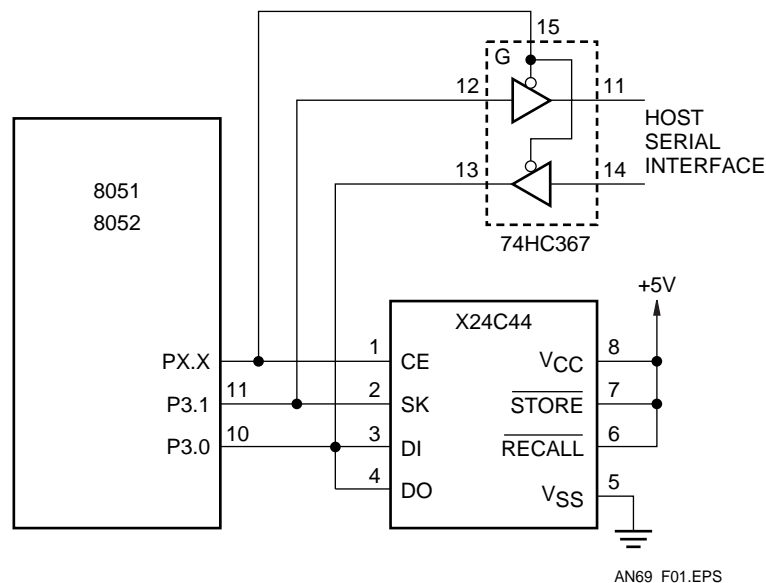


Figure 1. 8051 and X24C44 Interface Adds Scratch Pad RAM and Nonvolatile Parameter Storage via the 8051 Microcontroller Serial Port and Still Maintain Full Use of the Serial Port as a U ART

The P3.1 and P3.0 pins, when inactive (neither transmitting nor receiving) are always a logic "1" (HIGH). When a data transfer commences, P3.1 will be LOW during machine cycle states S3, S4, and S5 and will be HIGH during states S6, S1, and S2. When transmitting, data is shifted out on P3.0 during S6P2 (state 6 phase 2) LSB first. When receiving, data is sampled during S5P2. Refer to Figure 3 for the basic 8051 serial port timing.

Hardware Connections

The X24C44 directly interfaces with the 8051 with no external circuitry required. DI and DO of the X24C44 are both tied to P3.0, SK is tied to P3.1, CE is tied to any free port pin configured as an output, and STORE and RECALL are tied to V_{CC} (see Figure 2).

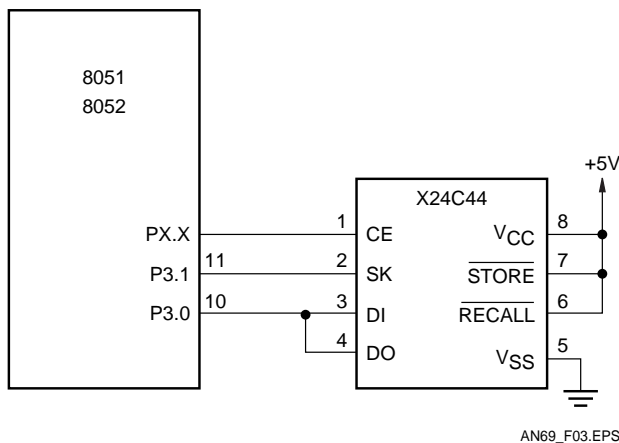


Figure 2. Basic Configuration

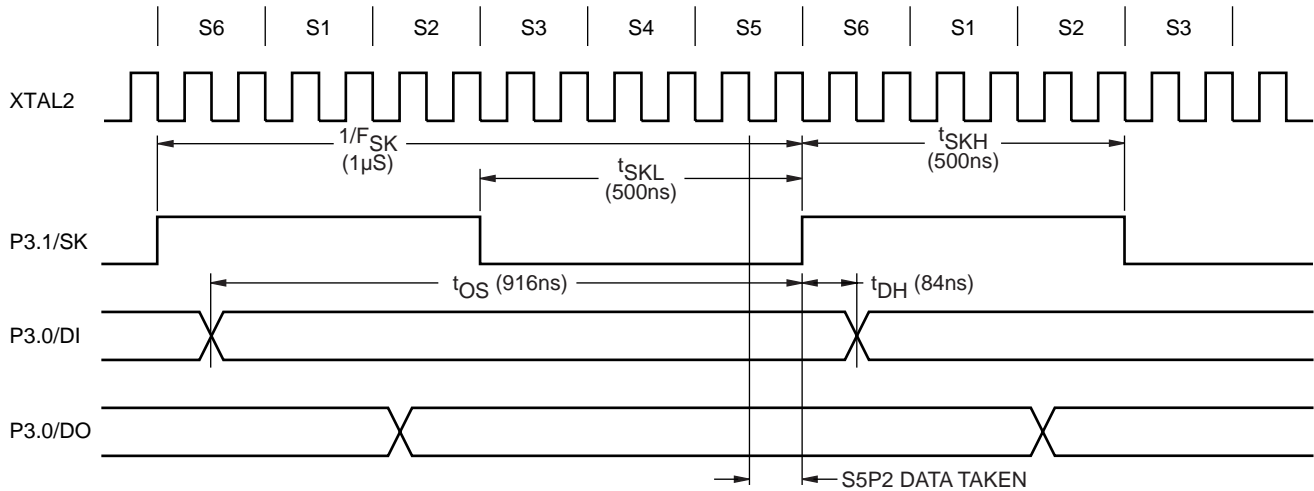
X24C44 Operations Review

The X24C44 is a serial device and in this application, all chip functions are handled via the software instructions. The 8051 transmits data LSB first, but the instruction format for the X24C44 shows the instruction to be transmitted MSB first. This requires a simple transposition of the instruction, MSB for LSB. The memory is effectively a FIFO, so the data to be stored need not be transposed.

Internally the X24C44 increments a bit (clock) counter. This is used to indicate the end of an instruction and, if a read or write instruction is received, to increment a bit position pointer. This pointer enables individual RAM cells for writing and reading. The counter for the pointer increments from zero to fifteen. If CE remains HIGH and SK continues to clock, the counter will rollover from fifteen to zero. The word address does not increment, therefore during a write operation, if SK continues to clock and CE is HIGH, a 25th rising clock edge (8 edges for instruction + 16 edges for the data word + 1) would cause bit position zero to be overwritten.

System Characteristics

Under normal operating conditions, the X24C44 expects CE to transition LOW to HIGH when SK is LOW in order that the first bit of data can be clocked into the X24C44 on the first rising edge of SK. The data is sampled to see if it is "0" (a don't care state) or a "1" which is recognized as an instruction start. The 8051, however, places both P3.1 and P3.0 in the HIGH state when not actively transmitting. THIS IS OK! The X24C44 internally gates CE and SK; therefore, toggling the port pin controlling CE to a HIGH effectively generates the first rising edge of SK, and also clocks in the HIGH present at P3.0 (DI).



AN69_F02.EPS

Figure 3. 12MHz 8051 Serial Port Mode 0 and X24C44 Timing

What this does is clock a "1" into the X24C44 indicating the start of an instruction prior to any shifting operation by the 8051 serial port. This will require dropping the leading "1" from the instruction. See Table 1 for the WAS/IS conditions for the equivalent instructions to be used by the 8051.

The 8051 will still generate eight rising clock edges on P3.1 for each byte loaded into the shift register (SBUF), effectively providing the X24C44 with nine clocks for the first byte. For the single byte instructions, the ninth clock and data are ignored by the X24C44. Refer to Figure 4 for the single byte instruction timing.

Writing

Writing to the RAM array is straightforward. The write instruction is issued by the 8051 in the same manner as the single byte instructions. The MSB (eighth bit) of the instruction byte is clocked in on the equivalent ninth clock rising edge. This bit is recognized as the first data bit of the transfer and is initially written into the addressed word's bit position zero. The 8051 will continue to transmit two more bytes of actual data. The LSB (bit zero) of the first byte will be physically located in bit position one and all subsequent bits will also be offset by one. The MSB (sixteenth data bit) of the word will be written into bit position zero, overwriting the last bit of the instruction byte. Refer to Figure 5 for the sequence of operations.

Reading

Reading data back from the RAM array is quite similar. The X24C44 begins to shift data out during the instruction cycle (more on this later). After the instruction is shifted out, the 8051 must turn around P3.0 and configure it as an input. CE and SK are static during this period and the DO output will remain unchanged until after the rising edge of the first 8051 receive data clock. Therefore, the first data shifted into the 8051 will be from bit position one, equivalent to the LSB originally written. Refer to Figure 5 for the sequence of operations.

Bus Contention

There will not be any bus contention for single byte instructions or the WRITE command. However, for the READ command there could be contention. While the 8051 is still shifting out the instruction byte, the X24C44 begins to output data on the same line. Refer to figure 5, just after the falling edge of clock eight.

The 8051 shifts out data at S6P2. If the data changes state from "0" to "1" a high current enhancement FET is turned on for two 8051 system clock cycles. This is used to provide a fast rise time. At the end of this two cycle period, the enhancement FET is turned off and the output is held HIGH by a depletion mode FET that essentially looks like a resistor pull-up (refer to Intel's Microcontroller Handbook [1984] pages 6-6 and 6-7).

Note that the high drive circuit is enabled only for data state changes from "0" to "1"; therefore, if the output is already a "1" and another "1" is shifted out on P3.0, the high drive will not be turned on. This depletion FET can source a maximum of 250 μ A if the port pin is grounded.

The instruction table indicates that bit seven for the READ instruction should be a "1". The reason for this is to guarantee that the high drive period is off before the X24C44 begins to output data. If bit seven were a "0", the 8051 would turn on the high drive circuit to return P3.0 to the inactive state, possibly generating a high current contention problem with the DO output of the X24C44. Figure 6 illustrates the timing involved during clock eight. The high drive period of the 8051 is turned off well before the X24C44 begins to output data.

Versatility

The DO output of the X24C44 is always in the high impedance state unless it is outputting data in response to a READ command. Therefore, the serial port of the 8051 need not be dedicated solely to a serial memory

interface.

Figure 1 illustrates the versatility that this affords. This figure depicts the basic system components required in a remote location controller. Notice that the 8051 serial port has access to both the X24C44 and through the use of the CE control line maintains full use of the serial port as a UART. Therefore, it can receive downloaded parameters from a host, re-enable the serial port for X24C44 communication, then store the data either temporarily in the X24C44 RAM array or permanently in the X24C44 E²PROM array.

Conclusion

This application note has shown that with no extra hardware, the X24C44 interfaces directly with the 8051 family of microcontrollers, providing a nonvolatile memory storage and scratch pad memory, while maintaining full 8051 UART capabilities. It is the ideal solution for applications where extra memory is required but few port pins are available for implementation.

Table 1. Reconfigured Instruction Format

INSTRUCTION	WAS								IS								
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
WRDS	1	X	X	X	X	0	0	0		X	0	0	0	X	X	X	X
STO	1	X	X	X	X	0	0	1		X	1	0	0	X	X	X	X
RESERVED	1	X	X	X	X	0	1	0		X	0	1	0	X	X	X	X
WRITE	1	A	A	A	A	0	1	1		X	1	1	0	A	A	A	A
WREN	1	X	X	X	X	1	0	0		X	0	0	1	X	X	X	X
RCL	1	X	X	X	X	1	0	1		X	1	0	1	X	X	X	X
READ	1	A	A	A	A	1	1	X		1	X	1	1	A	A	A	A

*Note: bit 7 of the READ command should be a "1" to avoid bus contention.

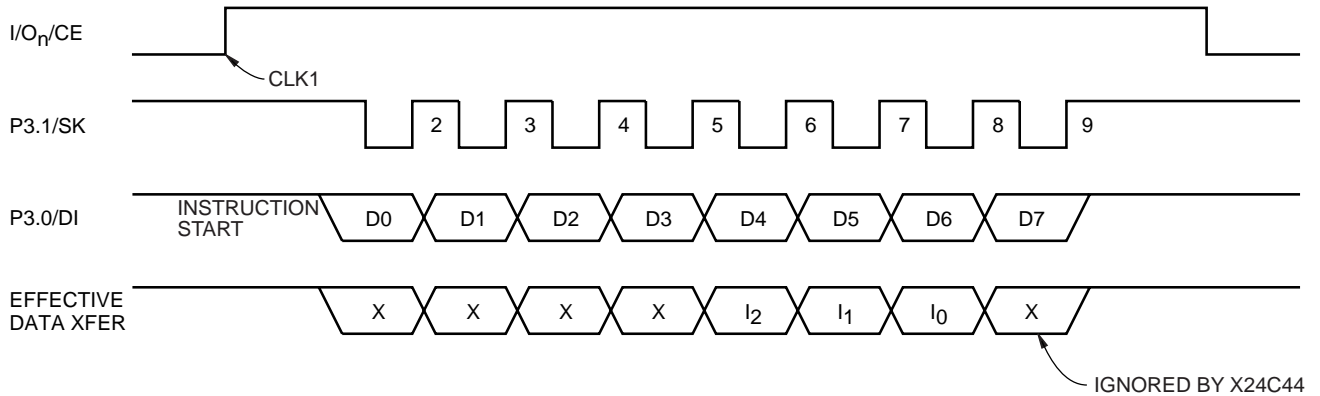


Figure 4. Single Byte Instructions

AN69_F04.EPS

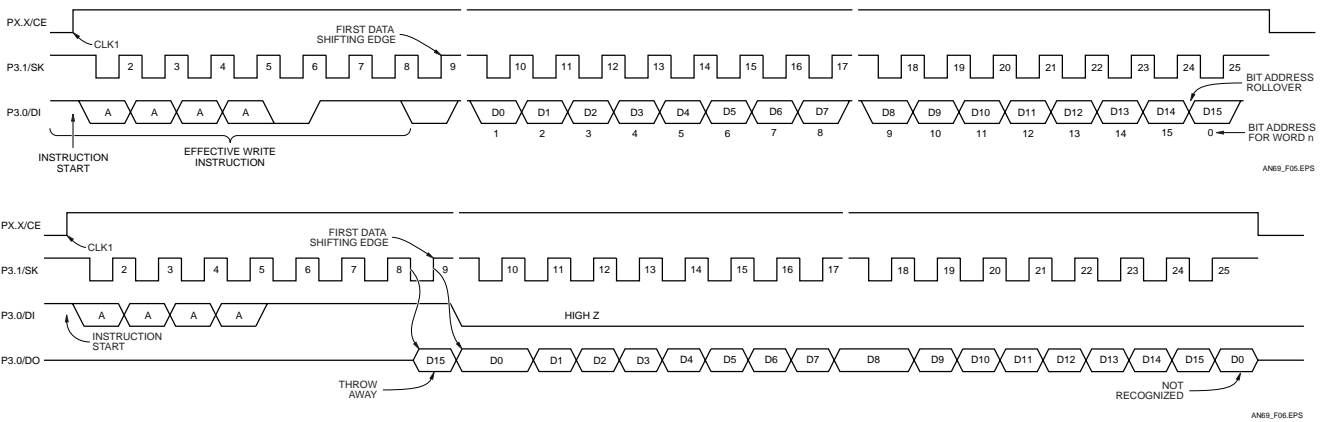


Figure 5. Read Cycle Sequence

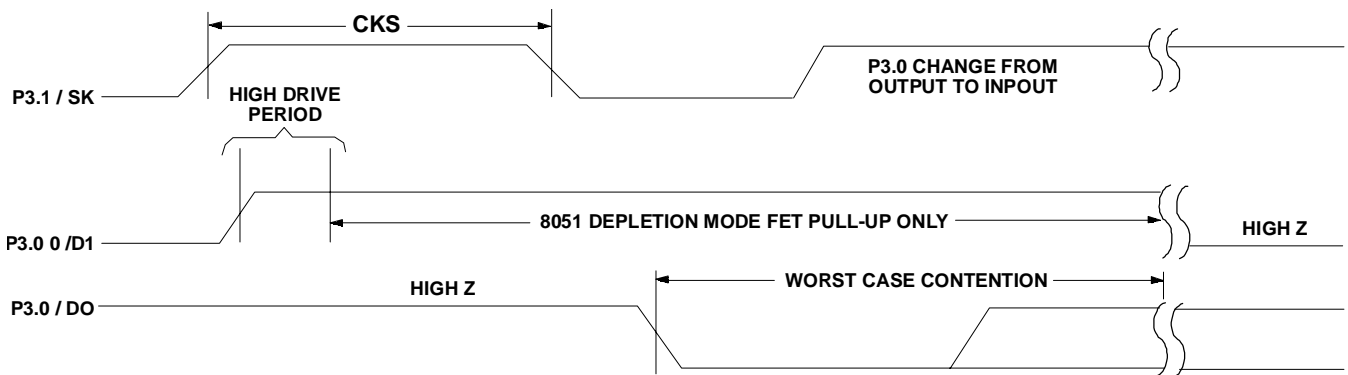


Figure 6. Worst Case Bus Contention

AN69F06.BMP