

# ***Transferring 16-Bit Data Using the TMS320C17 DSP 8-Bit Serial Port***

---

---

*APPLICATION REPORT: SPRA093*

*Oh Hong Lye  
Field Applications Engineer  
Texas Instruments Singapore Pte Ltd*

*Digital Signal Processing Solutions  
January 1997*



## **IMPORTANT NOTICE**

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain application using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

## **TRADEMARKS**

TI is a trademark of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.



## Contents

<b>Abstract.....</b>	<b>7</b>
<b>Product Support on the World Wide Web.....</b>	<b>8</b>
<b>Introduction .....</b>	<b>9</b>
<b>Concatenation .....</b>	<b>10</b>
The Polling Method.....	10
<b>The Counter Method.....</b>	<b>14</b>
<b>Summary.....</b>	<b>17</b>

## Figures

Figure 1.	Modified Serial Port Receive Timing Diagram.....	10
Figure 2.	Receiver Section Hardware Connections .....	11
Figure 3.	Receiver Flow Chart.....	12
Figure 4.	Counter Hardware Connection .....	14
Figure 5.	Flow Chart for Improved Solution .....	16

# Transferring 16-Bit Data Using the TMS320C17 DSP 8-Bit Serial Port

---

---

---

## Abstract

This application report describes two methods of using an 8-bit serial port to receive a 16-bit data word. The first method uses the Texas Instruments (TI™) TMS320C17 ('C17) digital signal processor (DSP) 8-bit serial port to receive the 16-bit data word. The second method improves the first method by adding a counter to the hardware. This report describes the hardware and software modifications required for the 16-bit to 8-bit data conversion.



---

## Product Support on the World Wide Web

TI's World Wide Web site at [www.ti.com](http://www.ti.com) contains the most up-to-date product information, revisions, and additions. New users must register with TI&ME before they can access the data sheet archive. TI&ME allows users to build custom information pages and receive new product updates automatically via email.



---

## Introduction

The TI 'C17 fixed-point DSP is designed for telecommunication applications with a hardware A-law/U-law compander and a host-interface. 'C17 functional code is readily available for line echo cancellation and the encoding/decoding for dual tone multifrequency (DTMF) generation.

A limiting factor in designs for the 'C17 DSP is the 8-bit data serial port. The 8-bit serial port poses a problem for DECT cordless phone implementations because of the 16-bit data format of the phone.

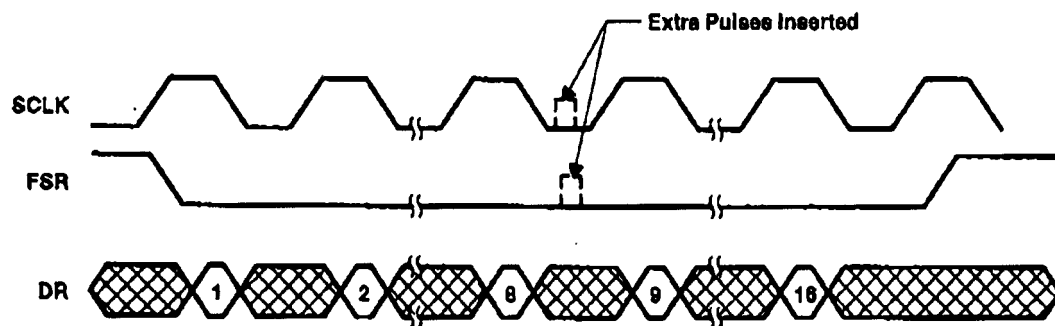
To solve the data format problem, an extra pulse is generated into the Frame Sync (FSR) and Data Clock (SCLK) input signals of the DSP. This extra pulse creates a second receive operation, which allows the DSP to receive sixteen bits of data during one BMC frame interval.

## Concatenation

Since the 'C17 does not support continuous mode transfer, a glueless interface is impossible. To allow the 'C17 serial port to accept sixteen bits of data during one frame interval, the signals into the FSR and the SCLK pin are altered. At the end of transmission of the first eight bits, an extra pulse is generated onto these pins. The extra pulse creates an additional FSR, which forces the DSP to latch in another eight bits of data. The two data sets are concatenated to form a 16-bit data word.

Figure 1 displays the eighth bit being latched into the serial port. An extra pulse is generated on the FSR pin, causing the serial port to undergo another receive operation. This latches in the second eight bits of the incoming data word. The extra pulse on the SCLK pin is used to transfer the first eight bits from the receive shift register (RSR) to the data receive register (DRR).

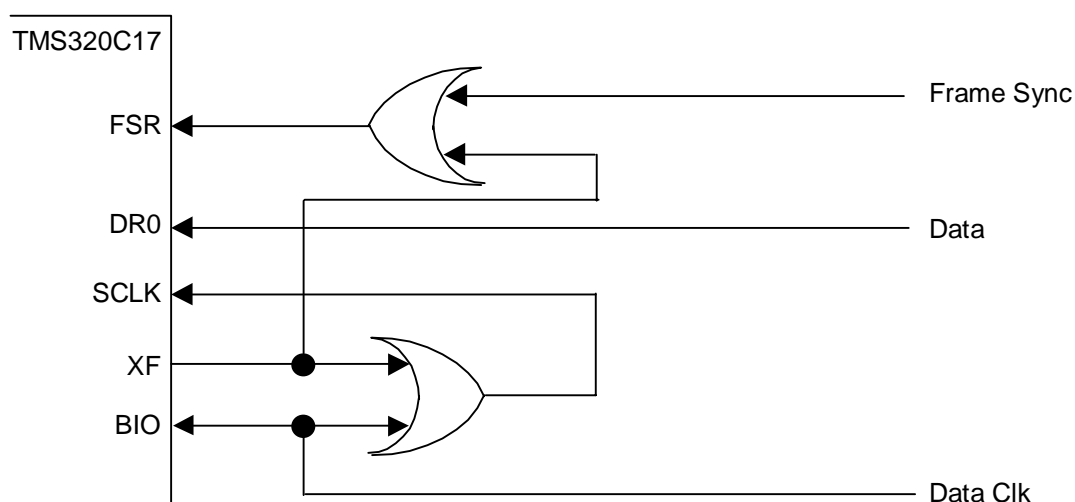
Figure 1. Modified Serial Port Receive Timing Diagram



## The Polling Method

The TI 'C17 DSP contains both an XF pin, used to generate the extra pulse for latching in the data, and a BIO pin, used to track the number of bits latched into the circuit. Figure 2 displays the receiver hardware connections. The transmitter section is similar to the receiver section.

*Figure 2. Receiver Section Hardware Connections*

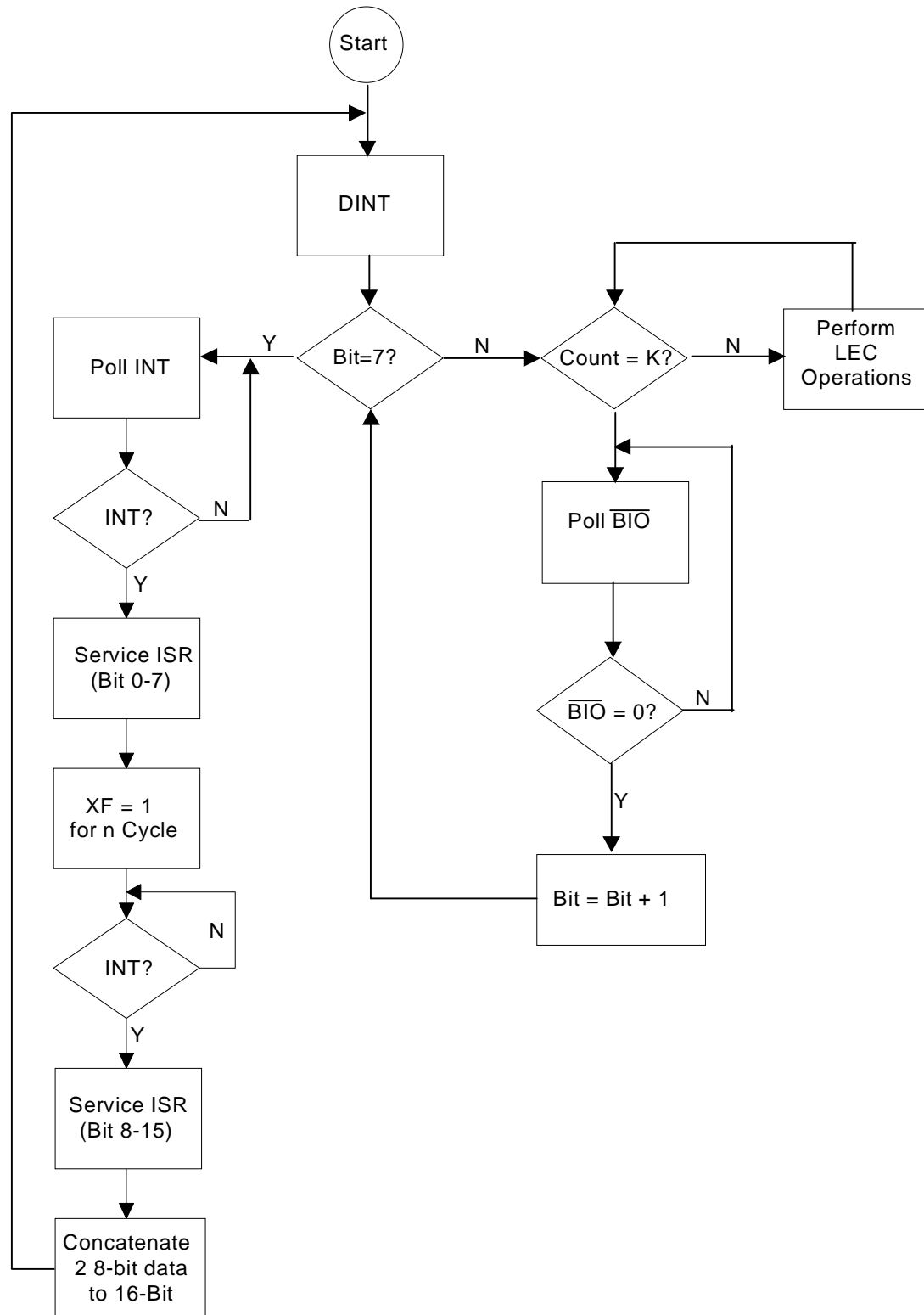


In the receiver, the data is clocked in on the rising edge of the data clock. When FSR is low, the data passes from Data to the DR0 input. The transmit section contains similar connections, with FSX and DX0 replacing FSR and DR0 respectively.

Figure 3 displays the receiver flow chart for the software of this circuit. The Bit variable monitors the number of data bits received, while the Count variable monitors the number of instruction cycles executed since the previous receive operation.



Figure 3. Receiver Flow Chart





For the program to poll at the appropriate time, the programmer needs to know how many instructions to execute between polls. At a predetermined time, the  $\overline{BIO}$  pin is polled for a high to low transition.

When eight bits of data are received, the receive interrupt (RINT) is polled to latch in the bits. The XF pin is then pulsed high to generate extra FSR and SCLK transitions. The first RINT is polled to reduce interrupt latency. Hardware interrupts are used to receive the second eight bits of data. The two data bytes are concatenated to form a 16-bit word. Using the hardware interrupts for the second 8-bits allows the DSP to perform other operations while waiting for the second data byte.

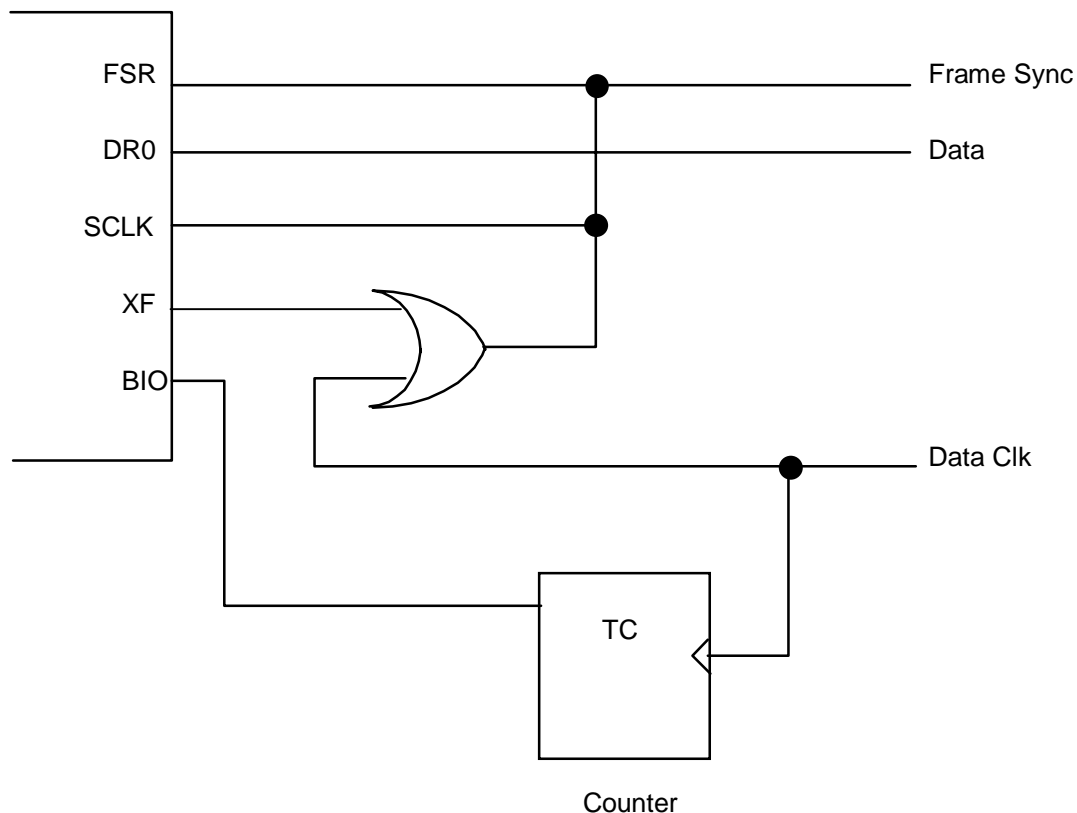
On the transmit side, a pulse is output through the XF pin after the eight bits are transferred.

## The Counter Method

Two problems arise when using the polling solution. The first problem occurs when the computation polling for the various signals is not synchronized, causing recovery to be difficult. The second problem occurs during the polling of the  $\overline{BIO}$  signal after each bit is received, causing CPU resources to be wasted for each unnecessary poll.

Adding a counter to the hardware improves the polling method. Instead of the DSP tracking the number of bits transferred, the counter performs this function. After detecting an 8-bit transfer, the counter outputs a pulse to the  $\overline{BIO}$  pin of the DSP. The DSP then generates an extra pulse on the XF pin. This allows the DSP to poll the  $\overline{BIO}$  for every byte received versus every bit and also frees the CPU to perform other tasks. Figure 4 displays the hardware connections between the DSP and the counter.

Figure 4. Counter Hardware Connection

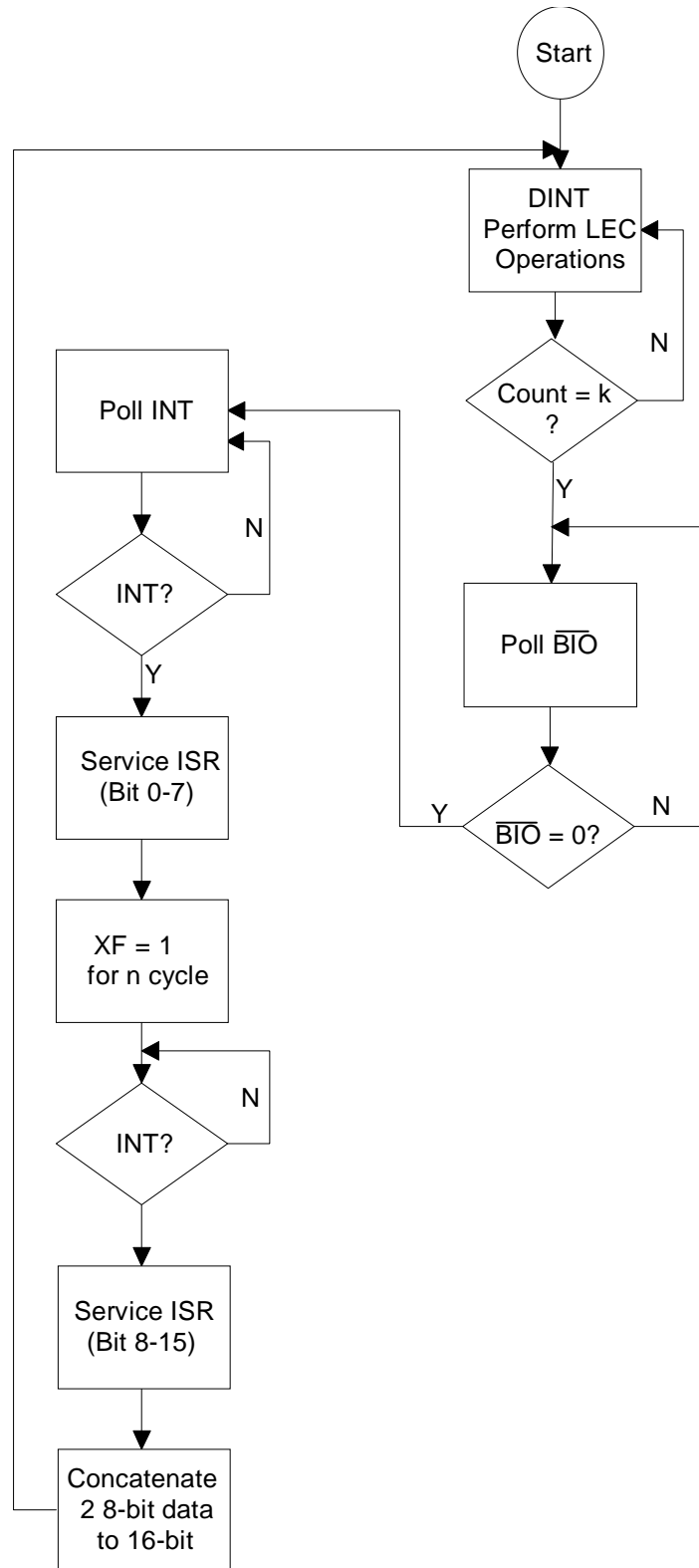


---

Figure 5 displays the flow chart for the improved method. This flow chart differs from the flow chart of Figure 3 as the data is detected as a byte versus bit by bit. The Count variable still keeps track of the number of instructions executed since the last receive operation, but the value 'k' represents the number of instructions that can be executed before eight bits of data are latched in. Conversely, in the first method the value 'k' represents the number of instructions executed before one bit of data is latched in. The solution allows the DSP to perform more operations without interruption.



Figure 5. Flow Chart for Improved Solution





---

## Summary

Two methods of interfacing an 8-bit serial port to 16-bit data are discussed. The first method is a customer's solution. The second method, however, improves on some of the problems found with the former, such as excessive usage of polling method. This excessive usage results in unnecessary wastage of CPU time and also reduces the system's capability to recover from tracking error. The proposed method uses a hardware counter to relieve the DSP from some of the tracking function, making the system more efficient and versatile.