Designing the TMS320C203 DSP Development Board for TMS320C203 Evaluation

APPLICATION BRIEF: SPRA348

Authors: Art Chen
Kai-Ming Chung
TMS320 DSP Applications
SC Application Specific Products
Texas Instruments Asia Pacific Region

Digital Signal Processing Solutions
August 1997

TEXAS INSTRUMENTS

TRADEMARKS

TI is a trademark of Texas Instruments Incorporated.

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

Contents

Abstract **7**
Product Support **9**
Related Documentation 9
World Wide Web 9
Introduction **10**
Hardware Description **11**
TMS320C203 Digital Signal Processor 11
TLC320AC01 Analog Interface Circuit 13
TLE2064 Operational Amplifier 15
External Memory 16
Asynchronous Serial (UART, RS-232) Port 18
JTAG (XDS-510) 19
Power System 20
Layout Issues 21
Software Description **22**
Memory Configuration 22
Echo Program 23
UART Program 28
Appendix A. TMS320C203 Development Board Schematics **33**
Appendix B. Bill of Materials **41**

Figures

Figure 1. TMS320C203 DVB Block Diagram 10
Figure 2. TMS320C203 DSP PZ Package – Top View 12
Figure 3. TLC320AC01 Functional Block Diagram 14
Figure 4. TLC320AC01 AIC FN and PM Packages – Top View 15
Figure 5. Differential Input 15
Figure 6. Differential Output 16
Figure 7. Block Diagram of External Memory 17
Figure 8. Bus Conflict in Data Bus 17
Figure 9. RS-232 9-Pin Connector (DB9) 19
Figure 10. JTAG Cable Header and Signals 19
Figure 11. Ground System 20
Figure 12. Memory Configuration 22
Figure 13. Signal Communication Path 23
Figure 14. Connecting the Synchronous Serial Port with Other Devices 24
Figure 15. Synchronous Serial Port Block Diagram 25
Figure 16. Asynchronous Serial Port Block Diagram 29

Designing the TMS320C203 DSP Development Board for TMS320C203 Evaluation

Abstract

This application brief describes the design of the TMS320C203 development board (DVB) from both the hardware and software approach. The DVB is a simple stand-alone application board used to evaluate the performance and characteristics of the TMS320C203 digital signal processor (DSP) hardware and software.

The DVB contains the Texas Instruments (TI™) TMS320C203 DSP and provides full-speed verification of TMS320C203 codes. Communication with the host PC is configured through an RS-232 interface via the built-in UART (Universal Asynchronous Receiver and Transmitter) port. The DVB transmits and receives the analog signals via the analog interface circuit (AIC) and operational amplifier (OP Amp, OPA) port. The DVB connects with the XDS-510 (JTAG - Joint Testing Action Group, IEEE1149.1 Standard) and uses the TMS320C2xx ('C2xx) emulation software as a debugging tool.

This application brief discusses the technologies behind the power system, clock, AIC/OPA, UART, and DSP. We focus on the memory configuration and code development for direct communication between on-chip synchronized/ asynchronized serial port and serial devices AIC/RS-232.

NOTE:

You may see the term *DSK* in the circuits described in this application report. The DVB was originally named *TMS320C203 DSP Starter Kit (DSK*), which is a formal name at TI, then renamed *DVB* on completion of the project.

The authors extend their deep appreciation to Chaucer Kuo, William Chen, Kevin Chang, Gene Lin and Jeffery Lai from the ASP-DSP Application Team; Jerry Chen from the ASP-ASIC Team; Ryan Hsiao from MSLP; and Vivian Shao from the Taiwan CAC Team for their invaluable help in this project.

Product Support

Related Documentation

TMS320C203, TMS320C209, TMS320VC203 Digital Signal Processors, Literature number SPRS025

TMS320C2xx User's Guide, Literature number SPRU127A, Preliminary Edition

TMS320C1x Evaluation Module Analog Interface Application Report, Literature number SPRA029

TMS320C5x DSP Starter Kit User's Guide, Literature number SPRU101

TLC320AC02C, TLC320AC02I Single-Supply Analog Interface Circuit Data Manual, Literature number SLAS084A

MOS Memory Commercial and Military Specifications Data Book, Literature number SMYD095

Operational Amplifiers and Comparators Data Book, Volume B, Literature number SLYD012

F Logic (SN54/74F) Data Book, Literature number SDFD001B

ABT Advanced BiCMOS Technology A High Performance Line of 5-V and 3.3-V Products Data Book, Literature number SCBD002B

Data Transmission Circuits Data Book, Volume 1, Literature number *SLLD001A*

Semiconductor Group Package Outlines Reference Guide, Literature number SSYU001B

UMC UM61256G Series 32Kx8 CMOS SRAM

World Wide Web

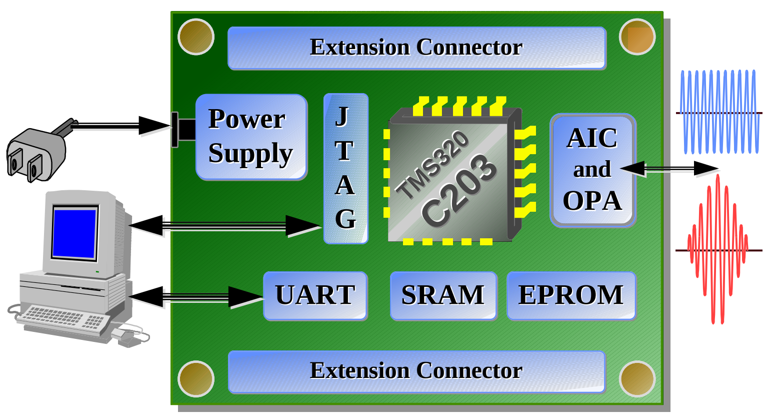
TI's World Wide Web site at www.ti.com contains the most up-to-date product information, revisions, and additions. New users must register with TI&ME before they can access the data sheet archive. TI&ME allows users to build custom information pages and receive new product updates automatically via email.

Introduction

The TMS320C203 DVB enhances the ability to create your own project by implementing software codes, building daughter boards, and expanding your system as desired. Figure 1 shows the DVB block diagram, which includes the following components:

- TMS320C203 DSP
- TLC320AC01 AIC and TLE2064 operational amplifier for the analog I/O port
- 32K SRAM and 32K EPROM configurable for program and data memory
- UART RS-232 port
- JTAG (Joint Testing Action Group, XDS-510) port
- Power supply for +5 V, -5 V, +12 V, -12 V, analog ground, and digital ground
- 10 MHz oscillator for both the TMS320C203 DSP and TLC320AC01 AIC

Figure 1. TMS320C203 DVB Block Diagram



Hardware Description

The TMS320C2xx generation of the TI TMS320 DSP is fabricated using static CMOS integrated circuit technology. The combination of advanced Harvard architecture, on-chip peripherals, on-chip data memory, and a highly specialized instruction set is the basis for the operational flexibility and speed of this device.

TMS320C203 Digital Signal Processor

The TMS320C203 DSP, packaged in a 100-pin PZ TQFP, includes the following features:

- T320C2xLP core CPU
- Source code downwardly compatible with the TMS320C1x and TMS320C2x and upwardly compatible with the TMS320C5x
- 544 words of on-chip, dual-access data RAM for B0, B1, and B2 blocks
- Input clock options of x1, x2, x4 or /2
- On-chip 16-bit timer
- 0-7 wait states software programmable to each space
- One synchronous serial port with four-level deep FIFOs
- Full-duplex asynchronous serial port (UART)
- XDS-510 (JTAG) port fully supported

Figure 2 shows the top view of the 100-pin TMS320C203 DSP PZ package. The package includes the following pin groups:

- Parallel data (D0-D15), address bus (A0-A15), and memory control signals

  Used for data transfer between the TMS320C203 DSP and external memory

- Initialization, interrupts, and reset operation control pins

  Provides direct control of the DSP

- Synchronous serial port and asynchronous serial (UART) port signals

  Communicates with the host or other devices having the same kind of the port

- Multiprocessing signals

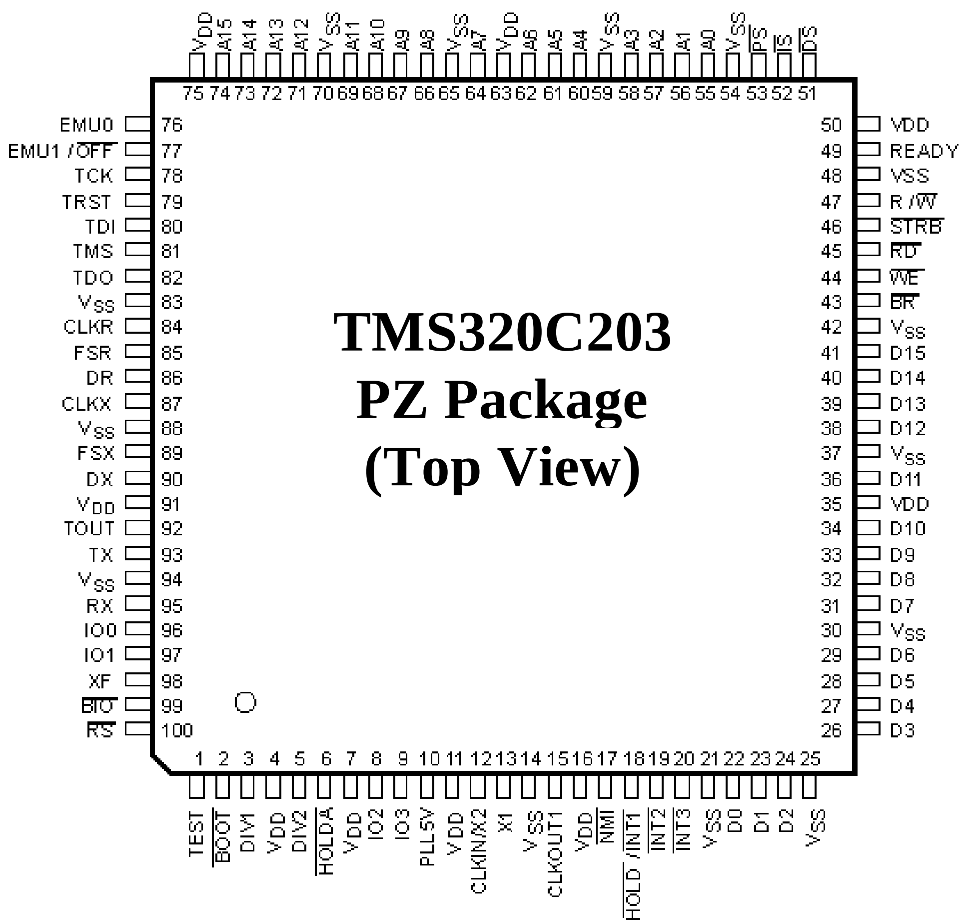
  Cooperates with other DSPs

- Oscillator, phase-locked loop (PLL), and timer signals
- Power supply pins
- JTAG signals

  Defined in the IEEE1149.1 standard and accessed by the emulator

Appendix A shows circuit schematics designed using these pin definitions.

Figure 2. TMS320C203 DSP PZ Package – Top View



The TMS320C203 is the most cost-effective DSP chip with high MIPS in the fixed-point DSP family. The device is built on the high-performance T320C2xLP core and integrates on-chip peripherals that make it well suited for a variety of applications, including:

- Set-top boxes
- Power line monitors
- Solid state relays
- Hard disk drives
- CD-ROMs
- Feature phones
- Phone-like data modems for LCD phone displays
- Caller ID
- DTMF
- Voice mail
- Centrex modems.

The TMS320C203 DSP is designed so that manufacturers of high-volume applications can reap the benefits of high performance DSPs without paying the higher prices historically associated with them. System code and hardware development for the T320C2xLP core is supported using JTAG scan-based emulation. The serial scan interface to the core is bonded out of the device so that the XDS-510 system emulator can interface with the DSP core. In this way, the system tested and verified using the TMS320C203 DVB can be designed with or without a daughter board.

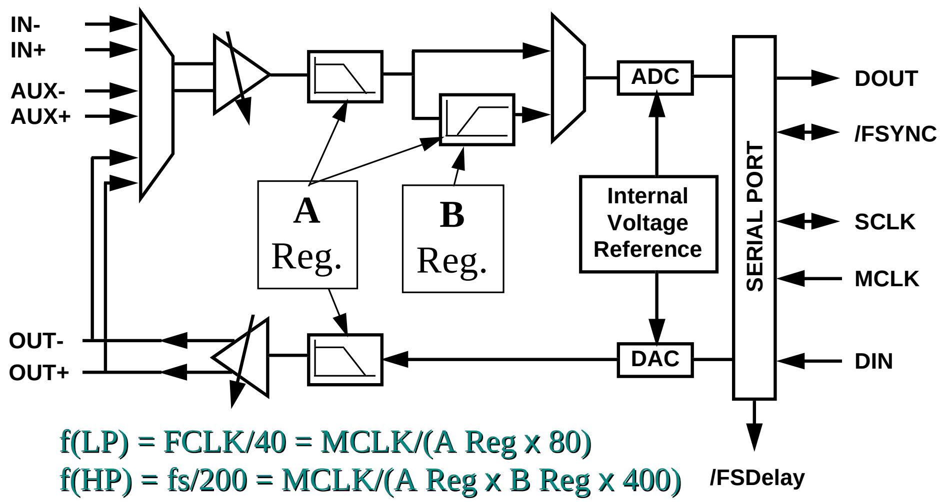
TLC320AC01 Analog Interface Circuit

The TLC320AC01 AIC is an audio-band signal processor that simultaneously provides an analog-to-digital/digital-to-analog input/output interface system on a single, monolithic CMOS chip. The A/D, D/A, and AIC combination is useful to the DSP solution design. The AIC includes the following features:

- Needs only a single 5 V power supply
- Synchronous serial port Interface
- General-purpose, signal-processing analog front end (AFE)
- 14-bit dynamic-range ADC and DAC in 2s-complement data format
- (Sin X)/X compensation supported
- Programmable filter bandwidths (up to 10.8 KHz)
- Programmable output gain

Figure 3 shows the TLC320AC01 AIC functional block diagram.

Figure 3. TLC320AC01 Functional Block Diagram



Use the asynchronous serial port to send information controlling the configuration and performance parameters by eight available data registers. The data in the registers set up the device for a given mode of operation and application. The anti-aliasing input low-pass filter is a switched-capacitor filter with a sixth-order elliptic characteristic, followed by a second-order (Sin X)/X correction filter, followed by a three-pole continuous-time filter to eliminate images of the filter clock signal.

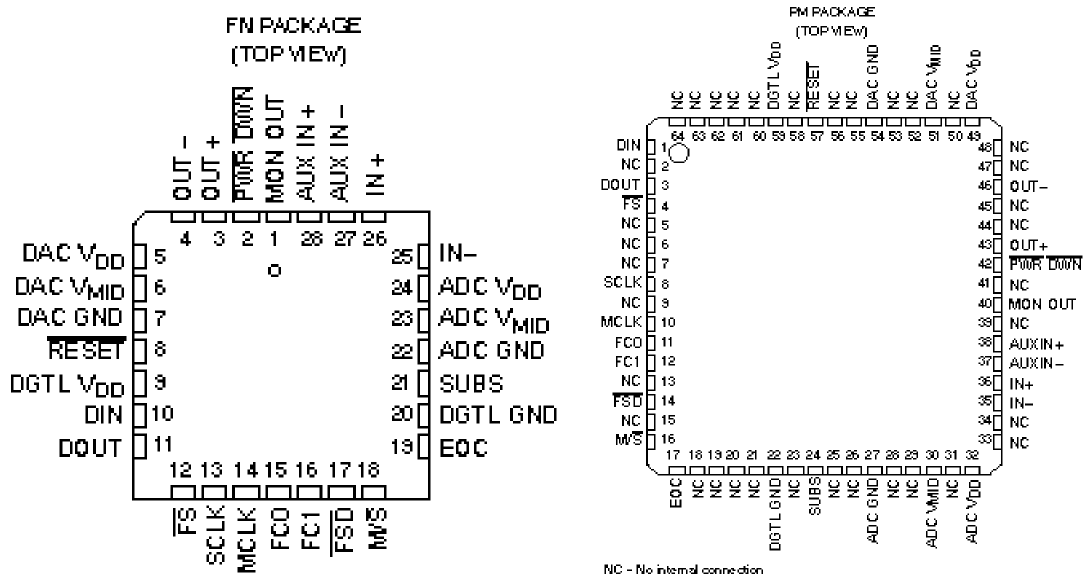
The high-pass filter is a single-pole filter that preserves low-frequency response as the low-pass filter cutoff is adjusted by the parameters in the related register (see Figure 3). Since the TLC320AC01 is a one-frame, synchronous signal only, we should connect the ’C203/FSR and ’C203/FSX with the ’AC01/FS pin. The transmit and receive clock are of the same design.

Two packages are available with the TLC320AC01 AIC (Figure 4 shows the top views of both packages.):

- ❑ 28-pin FN PLCC
- ❑ 64-pin PM TQFP

The TMS320C203 DVB uses the28-pin FN PLCC.

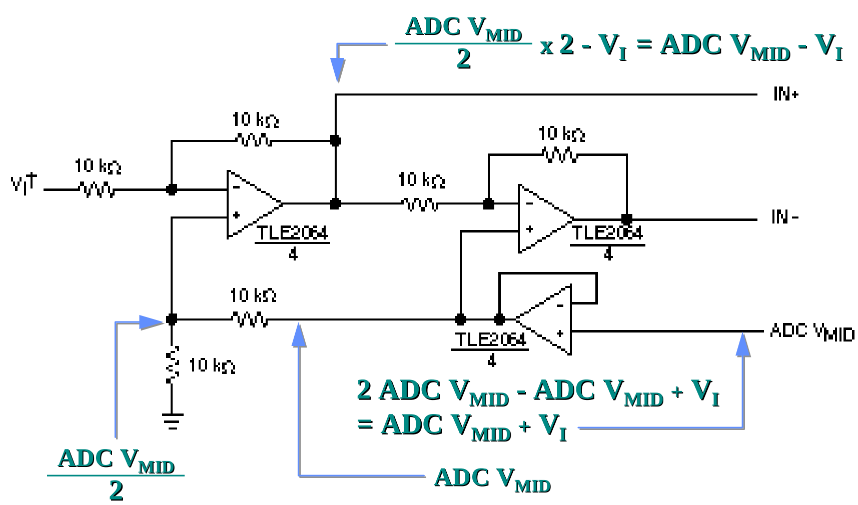
Figure 4. TLC320AC01 AIC FN and PM Packages – Top View



TLE2064 Operational Amplifier

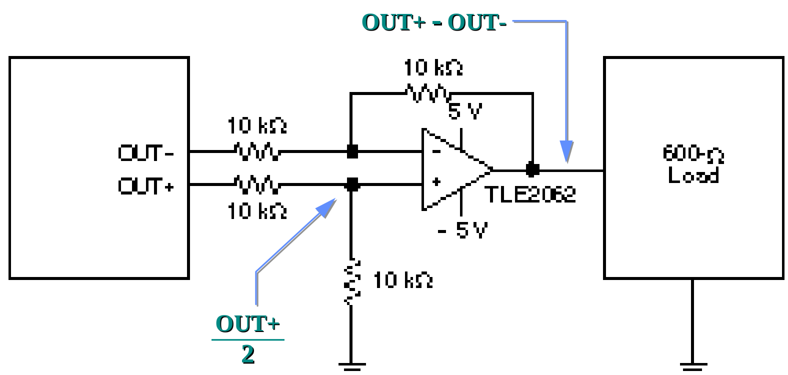
The AIC uses differential input and output and thus requires an operational amplifier (OP Amp, OPA). Because the AIC uses a single 5 V power supply only, we should take care of the middle point voltage (V_{MID}). Figure 5 and Figure 6 show the differential input with V_{MID} and differential output designs.

Figure 5. Differential Input



The TLE2064 is an audio band operational amplifier. Both 5~15 V and -5~-15 V should be offered to this chip, and 12 V and -12 V are chosen for the DVB. The TLE2064 combines outstanding output drive capability with low power consumption, excellent DC precision, and wide bandwidth. In addition to maintaining traditional JFET advantages of fast slew rates, low input bias, and offset currents, the TI Excalibur process offers outstanding parametric stability over time and temperature. The result is a precision device that remains precise, even with changes in temperature and long time in use.

Figure 6. Differential Output



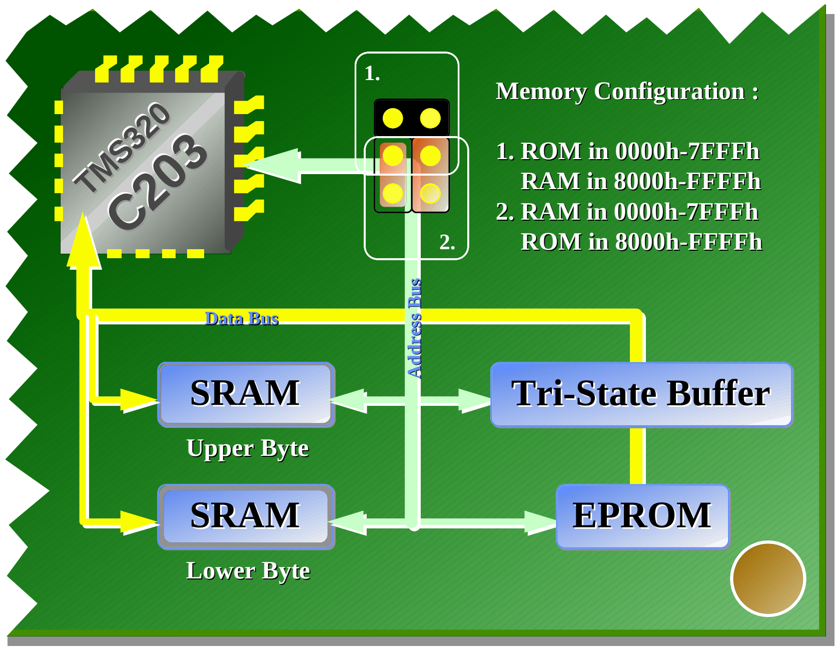
To enlarge the input and the output signals, semi-variable resistors are used as input resistors of the operational amplifiers. Unfortunately, because the DC signals are enlarged along with the AC signals, the operational amplifiers are easily saturated. For this reason, two capacitors are cascaded between the operational amplifiers and the I/O connectors as the AC couplers. Only AC signals that are large enough can be received and transmitted.

External Memory

The TMS320C203 DVB 64K words of external memory are split into SRAM and EPROM. You can configure the SRAM and EPROM as the upper 32K words and the lower 32K words by themselves, or vise versa.

Figure 7 shows the block diagram of the external memory system. These areas can be used as either program or data memory, as defined by the assembly code. The priority of usage in internal, on-chip data memory is higher than that for external data memory, so regardless which memory you select, the B0, B1, and B2 data memory areas are always used in the internal dual access RAM.

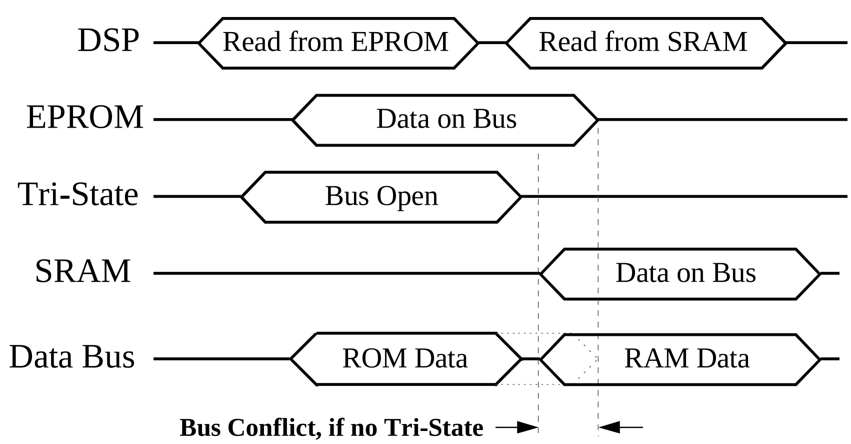
Figure 7. Block Diagram of External Memory



The 32K-word SRAM is combined by two 32K x 8 bits memory devices. The allowable memory access time must be under 15 ns because the TMS320C203 operates at such high speed. We use these two chips of SRAM in parallel, which means that one chip is the upper byte and the other is the lower byte.

The other 32K-word EPROM uses a 64K-word chip with a tri-state buffer made by Advanced Bipolar Technology for faster response. Since SRAM is much faster than EPROM, the tri-state buffer separates these two kinds of memory to avoid bus conflict problems (see Figure 8).

Figure 8. Bus Conflict in Data Bus



The TMS320C203 DSP can start the program only from program address 0000h and thus cannot be designed as the EPROM or the SRAM area in the fixed addressing position for the stand alone demo board design.

If you locate the SRAM in address 0000h-7FFFh and the EPROM in address 8000h-FFFFh, you can use the DVB connected with the XDS-510 (JTAG) as an emulation tool.

If you locate the EPROM in address 0000h-7FFFh and the SRAM in address 8000h-FFFFh, you can use the DVB as a stand-alone demonstration board.

Asynchronous Serial (UART, RS-232) Port

The TMS320C203 DVB provides communication with the PC host or other serial device via the UART/RS-232 port. Two problems must be resolved in the DVB's asynchronous serial port design:

- Voltage caused by the 9 V signals required for the RS-232
- No handshaking pin in the TMS320C203 DSP

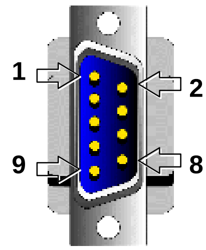
Both the SN75188 and SN75189 chips deal with the first problem. Both chips function as the data buffer for the voltage transfer from 5 V to 9 V and 9 V to 5 V requested by the RS-232 specification. The SN75188 and SN75189 chips invert the signals between the development board and the RS-232 serial port of the PC host.

Only two pins provide TX and RX for the asynchronous serial port in the TMS320C203 DSP. In the communications system, not only is the data transmission used but the handshaking signals also have to be controlled. To build up the handshaking signals, the following four general I/O pins are also supported on the TMS320C203 DSP:

- IO0 for DTR (Data Terminal Ready)
- IO1 for DSR (Data Set Ready)
- IO2 for RTS (Request to Send)
- IO3 for CTS (Clear to Send)

Connecting these pins along with the power supply pins are designed for asynchronous serial port to the DB9 connector, as shown in Figure 9 (see Appendix A for circuit schematics).

Figure 9. RS-232 9-Pin Connector (DB9)



DB-9

1. DCD - Data Carrier Detector
2. RXD - Received Data
3. TXD - Transmitted Data
4. DTR - Data Terminal Ready
5. GND - Signal Ground
6. DSR - Data Set Ready
7. RTS - Request to Send
8. CTS - Clear to Send
9. RI - Ring Indicator

JTAG (XDS-510)

To perform emulation with the XDS-510 following the IEEE 1149.1 specification, the target system must have a 14-pin header (two 7-pin rows, 0.025" x 0.235") with connections shown in Figure 10. Seven pins on the TMS320C203 DSP chip are used for the JTAG. These pins, as well as the power supply pins, are mapped to the 14-pin header.

Pin 11 on the JTAG pod is the Test Clock (10 MHz output) signal, which is generated from the emulator pod. I have an experience to parallel a diode in the inverse mode for the impedance compatibility problem solving. The TMS320C2xx emulation system may not connect properly if the TMS320C203 DSP (Batch Number 5349653) is used on the DVB.

Figure 10. JTAG Cable Header and Signals

7 x 2 Header

Test Mode Select - **TMS**	1	2	**TRST** - Test Reset
Test Data Input - **TDI**	3	4	**GND** - Ground
Presence Detect - **PD**	5	6	**No Pin (Key)**
Test Data Output - **TDO**	7	8	**GND** - Ground
Test Clock Return - **TCK_RET**	9	10	**GND** - Ground
Test Clock - **TCK**	11	12	**GND** - Ground
Emulation Pin 0 - **EMU0**	13	14	**EMU1** - Emulation Pin 1

0.025" x 0.235"

CAUTION:

Cut the No Pin (pin 6) to avoid plugging the connector in the wrong direction and thus connecting the Presence pin (5) with Ground and the Ground pin (10) with VCC (5 V).

This would connect the Presence pin and VCC with possibly serious results.

Power System

The power system design of the TMS320C203 DVB is a formal design similar to that of the TMS320C50 DSK design. The DVB includes more capacitors for noise bypass and an LED as a power indicator. The AC-9V adapter used with the TMS320C50 DSK can be used with the DVB as well.

The most important design consideration for the DVB power system is that digital ground connects to analog ground via a ferrite bead (a kind of core, 800 ohm/100 MHz = 1.27 μH). The resistance of the inductor in the frequency domain $Z(f) = j2\pi fL$

Where

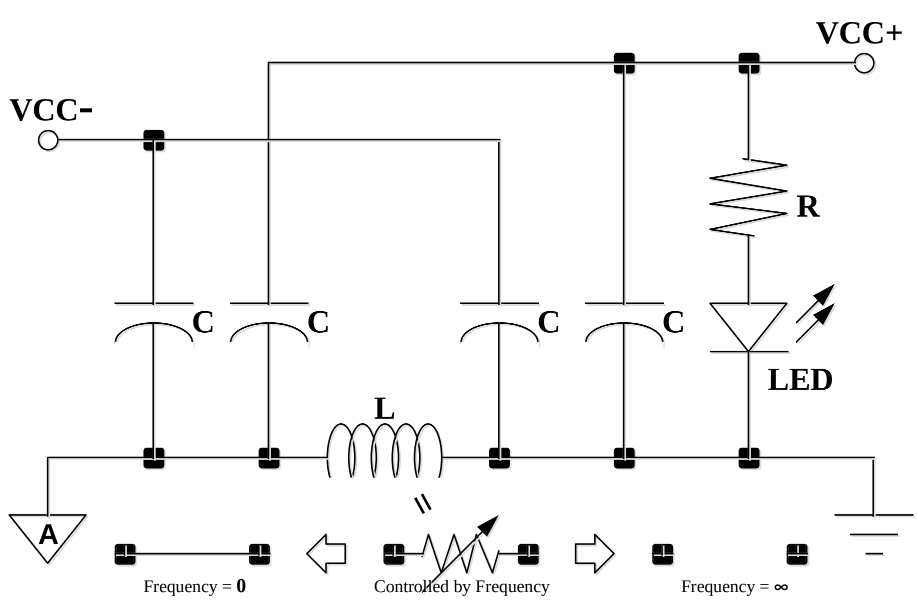
j = the image value

f = the frequency value

L = the inductor value

Thus, the high frequency of the noise generated from the digital ground cannot interfere with the analog ground. Analog devices, such as the AIC and OPA, will work more stable than before. In addition, this approach avoids EMI (electromagnetic interference) problems, because the analog devices are used always as the front-end components.

Figure 11. Ground System



Layout Issues

Because of the high frequencies of the signals running on the data and address buses between the TMS320C203 DSP and SRAMs, these two components must be placed as close as possible to each other. For the same reason, much noise is generated around these buses, which is why a four-layer board is implemented.

The TMS320C203 DVB offers the flexibility DSP application engineers require in an evaluation or development tool. A variety of component packages can be used:

- Either DIP or SMD LED package
- Either 1206 or 0805 resistors and capacitors
- Either full- or half-size oscillator
- Any one of three kinds of semi-variable resistor packages

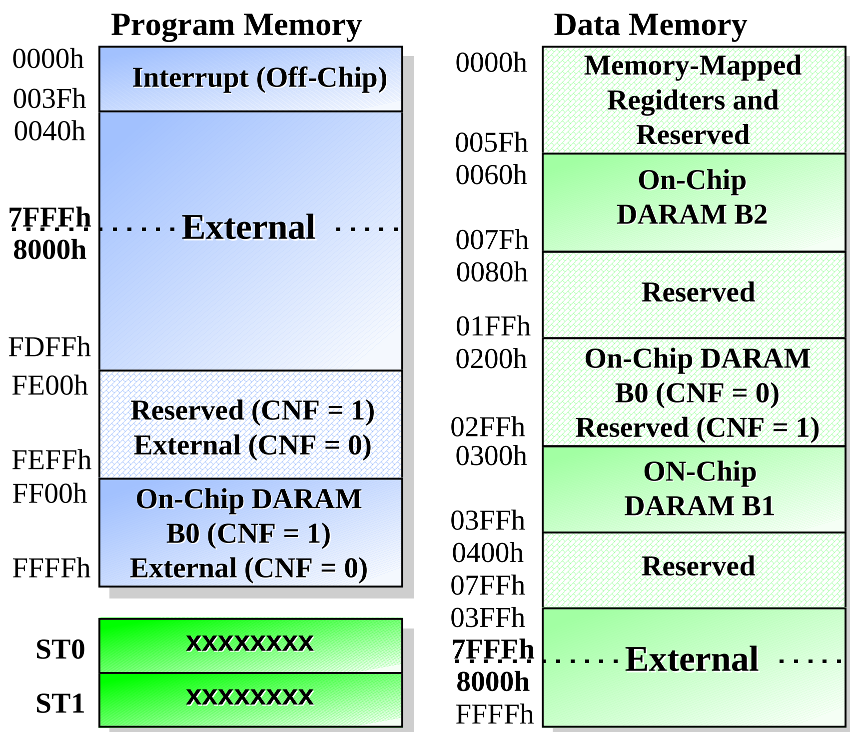
Several connectors for the extension board are reserved; thus, you can make your own design based using the DVB.

Software Description

Memory Configuration

The memory configuration is the most important consideration when writing assembly code (see Figure 12). As mentioned above, the priority of usage to internal data memory is higher than it is for external data memory (see *External Memory*). Regardless which memory you select, the B0, B1, and B2 data memory is always used in the internal RAM (for a discussion of registers, see the TI *TMS320C2xx User's Guide*).

Figure 12. Memory Configuration

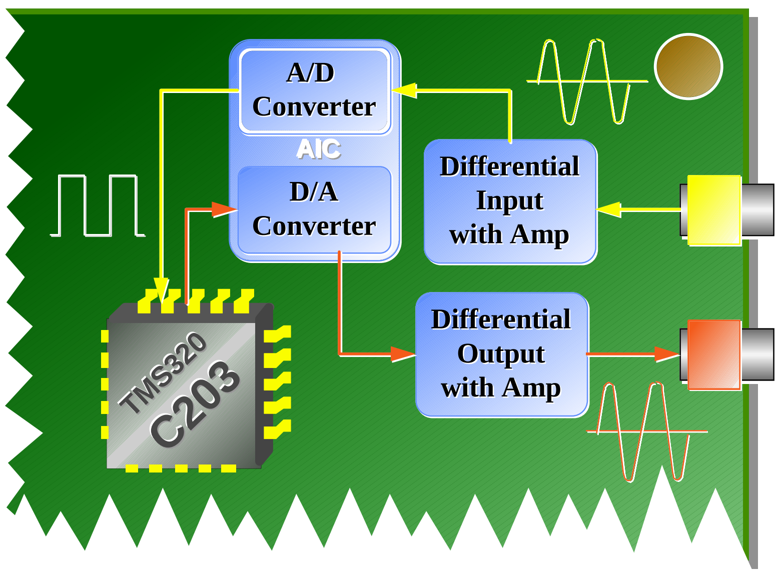


Echo Program

The Echo program enables the synchronous serial port to communicate with the TLC320AC01 AIC. Before the analog signal from the audio input is converted to digital data and then received by the DSP, the TLE2064 operational amplifier enlarges it.

After the AIC passes the digitized data to the DSP via the synchronous serial port, the DSP receives the data and sends it back directly via the synchronous serial port. If the function generator is used as the signal source and the oscilloscope is used as the observer, the same wave shape will be seen on the screen. The hardware signal communication paths are shown in Figure 13.

Figure 13. Signal Communication Path



The TMS320C203 synchronous serial port, which is controlled differently than the TMS320C50 DSP, offers the following features:

- Two four-word-deep FIFO buffers
- Interrupts generated by the FIFO buffers
- Maximum transmission rate of CLKOUT1/2
- Wide range of operating speeds
- Burst and continuous modes of operation

The synchronous serial port requires three kinds of signals:

- Clock signal

  Controls timing during the transfer and can be generated by an internal or external source.

- Frame sync signal

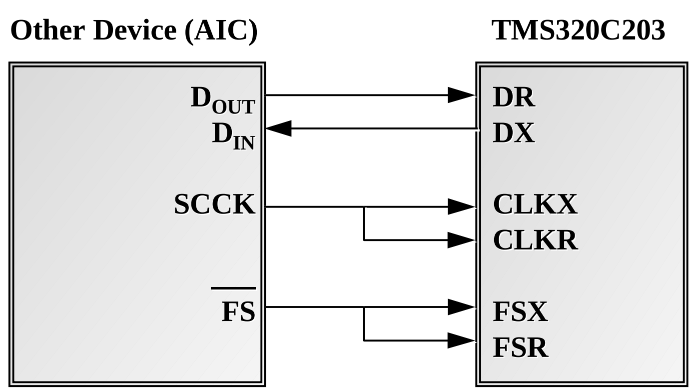
  Synchronizes transmit and receive operations at the start of a transfer and can be generated by an internal or external source.

- Data signal

  Carries the actual data transferred in the transmit/receive operation. The data signal transmit pin (DX) of one device should be connected to the data signal receive (DR) pin of another device.

The synchronous serial port also uses two on-chip, I/O-mapped registers as the synchronous serial port control register (SSPCR, FFF1h@IO) and the synchronous serial port transmit/receive register (SDTR, FFF0h@IO). Figure 14 shows how to connect the synchronous serial port with other devices.

Figure 14. Connecting the Synchronous Serial Port with Other Devices



Transmitting a word through the synchronous serial port is a four-step process as follows:

Step 1: Your software writes up to four words to the transmit FIFO buffer through the SDTR.

Step 2: The transmit FIFO buffer copies the first-written word to the transmit shift register (XSR) when the XSR is empty.

Step 3: The XSR shifts the data bit by bit (MSB first) to the DX pin.

Step 4: The XSR lets the FIFO buffer know when it is empty.

a) If the FIFO buffer is full, the process repeats starting at Step 2.

b) If the FIFO buffer is empty, it sends a transmit interrupt (XINT) to request more data and transmission stops.

Receiving a word from the synchronous serial port is a four-step process as follows:

Step 1: Data from the DR pin is shifted bit by bit (MSB first) into the receive shift register (RSR).

Step 2: When the RSR is full, the RSR copies the data to the receive FIFO buffer.

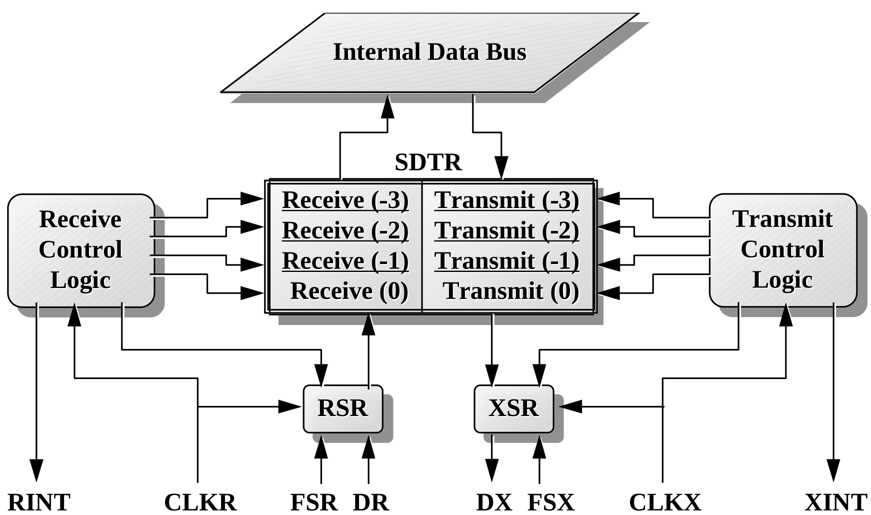
Step 3: One of two actions occur, depending on the state of the receive FIFO buffer.

a) If the receive FIFO buffer is not full, the process starts over at Step 1.

b) If the receive FIFO buffer is full, it sends a receive interrupt (RINT) to the processor to request servicing.

Step 4: The processor reads the received data from the receive FIFO buffer through the SDTR.

Figure 15. Synchronous Serial Port Block Diagram



For more information regarding the synchronous serial port, see the TI *TMS320C2xx User's Guide*.

Example 1 and Example 2 show the Echo.Asm and AIC_Table.Inc files. The polling mode is used in this program in addition to the interrupt mode.

A single-layer FIFO buffer is set up to avoid two problems not discussed in the *TMS320C2xx User's Guide*:

❑ If the transmit FIFO buffer is not full or empty, the status will be unknown.

- If the FIFO buffer is not full, the receive FIFO buffer will not generate the RINT.

At the beginning of the Echo.Asm program, the interrupt vector table should be defined in program address from 0000h to 0040h, and the I/O wait-state and synchronous serial port should be initialized (see Example 1).

Afterward, the AIC can be programmed through the synchronous serial port by the AIC setup table, shown in Example 2. The data can be transmitted and received successfully by RINT, since the CLKX and CLKR are connected together, and the RINT is in a higher priority.

Example 1. Echo.Asm File

```
;;-----------------------------------------------------------------------------------------------------------------------------
;; Program Name: Echo.Asm
;; Description:  1. The Synchronized Serial Port Programming
;;               2. AIC Controlled by the Synchronized Serial Port
;;               3. Get the Data from the Synchronized Serial Port via the AIC and the OPA
;;               4. Return the Same Data to the AIC and OPA from the Synchronized Serial Port
;; Author:       Art Chen in TI-Asia/DSP
;;-----------------------------------------------------------------------------------------------------------------------------

        .title   "Echo Program"

TMVER   .equu    0                              ;; 0 -> TMX, 1 -> TMS

        .mmregs                                 ;; Include Memory Mapped Registers
        .include "AIC_TBLE.INC"                 ;; Include AIC Control Table

SDTR    .set    0FFF0h                          ;; Synchronous Data Transmit/Receiver
SSPCR   .set    0FFF1h                          ;; Synchronous Serial Port Control Reg.
WSGR    .set    0FFFCh                          ;; Wait-State Generator
RFNE    .set    15 - 12                         ;; Receive FIFO Not Empty Bit
TCOMP   .set    15 - 13                         ;; Transmit Complete Bit

        .sect   "vectors"
        B       BEGIN                           ;; Reset Interrupt Vector
        .space  2 * 16                          ;; HOLD/INT1
        .space  2 * 16                          ;; INT2/INT3
        .space  2 * 16                          ;; Timer Interrupt ISR
        B       R_ISR                           ;; SyncSerPort Receive ISR
        B       X_ISR                           ;; SyncSerPort Transmit ISR
        .space  2 * 16                          ;; ...>

        .bss    PORT, 4                         ;; Use Internal RAM
r_port  .set    PORT                            ;; Receive Port
x_port  .set    r_port + 1                      ;; Transmit Port
ctrl    .set    x_port + 1                      ;; Control Signal

        .text
BEGIN:
        SETC    INTM                            ;; Disable Interrupt
        LDP     #PORT

;; Set Wait State
```

```
        SPLK    #0000h, ctrl                ;; 0-Wait State Set
        OUT     ctrl, WSGR

;; Init Serial Port Control Register, User's Guide 9-7

        SPLK    #0000000000000010b, ctrl    ;; Write the Value with Reset
        OUT     ctrl, SSPCR                 ;;
        SPLK    #0000000000110010b, ctrl    ;; Go!
        OUT     ctrl, SSPCR                 ;;

;; AIC initialization Routine

        LACL    #(AICTBL)                   ;; Load Address to AIC Control Table
        LAR     AR1, #AICTBL_LENGTH - 1     ;;
        MAR     *, AR1                      ;;

AIC_LOOP:                                   ;;          <---------------------------------------------\
        TBLR    x_port                      ;; Load Data from AIC Control Table|
        OUT     x_port, SDTR                ;;                                                        |
                                            ;;                                                        |
AIC_WAIT:                                   ;;          <----------------------------------\          |
        IN      ctrl, SSPCR                 ;; Waiting for                                 |          |
        BIT     ctrl, TCOMP                 ;;  Transmit Complete Bit                      |          |
        BCND    AIC_WAIT, NTC               ;;          ------------------------------------/          |
        ADD     #1                          ;; To Next One                                            |
        BANZ    AIC_LOOP, *-                ;; If Pointer != End, Wait      ------------------------/

;; Enable RINT and XINT

        LDP     #0
        SPLK    #0000000000011000b, IMR     ;; Set RINT, XINT
        CLRC    INTM                        ;; Enable Interrupt

LOOP:                                       ;; Waiting for Interrupt       <----------------------\
        NOP                                 ;;                                                    |
        B       LOOP                        ;;          ---------------------------------------------/

;; Interrupt Service Routines

R_ISR:                                      ;; Receive Interrupt Service Routine
        SETC    INTM                        ;; Disable Interrupt
        LDP     #0
        .IF     TMVER = 1                   ;; TMP Chip Version
        LACL    #00018H                     ;; Clear the RINT and XINT Flag
        .ENDIF
        .IF     TMVER = 0                   ;; TMX Chip Version
        LACC    IFR                         ;; Clear the RINT and XINT Flag
        AND     #0FFE7H                     ;;
        .ENDIF
        SACL    IFR                         ;;

        LDP     #PORT
        IN      r_port, SDTR                ;; Get the Data from FIFO

        CALL    PROCESS                     ;; Process

        LDP     #PORT
        OUT     x_port, SDTR                ;;
        CLRC    INTM                        ;; Enable Interrupt
        RET

X_ISR:
```

```
        SETC    INTM                              ;; Disable Interrupt
        LDP     #0
        .IF     TMVER = 1                         ;; TMP Chip Version
        LACL    #00018H                           ;; Clear the RINT and XINT Flag
        .ENDIF
        .IF     TMVER = 0                         ;; TMX Chip Version
        LACC    IFR                               ;; Clear the RINT and XINT Flag
        AND     #0FFE7H                           ;;
        .ENDIF
        SACL    IFR                               ;;

        CLRC    INTM                              ;; Enable Interrupt
        RET

;; Process

PROCESS:
        LDP     #PORT
        LACC    r_port                            ;; Let Transmit = Receive Port
        SACL    x_port                            ;;
        RET

;; End of Echo.Asm
```

Example 2. AIC_Table.Inc File

```
;;-----------------------------------------------------------------------------------------------------------
;; Header File:   AIC_Table.Inc
;; Description    TLC320AC01/02 AIC Initialization Data
;;                1. Master Clock (MCLK) = 10.0 MHz
;;                2. Frame Sync Clock (FCLK) = MCLK / (2 * A) = 142.857 kHz.
;;                3. Sample Rate = FCLK / B = MCLK / (2 * A * B) = 9.524 kHz.
;;-----------------------------------------------------------------------------------------------------------

                .data
AICTBL .word    0000h                             ;; Silent Padding
                .word   0000h                     ;; Silent Padding
                .word   0000000000000011b         ;; Control Request
                .word   0000000100100011b         ;; A = 0x23 = 35
                .word   0000000000000011b         ;; Control Request
                .word   0000001000001111b         ;; B = 0x0F = 15
                .word   0000h                     ;; Silent Padding
                .word   0000h                     ;; Silent Padding
                .word   0000h                     ;; Silent Padding

AICTBL_LENGTH           .equ    $ - AICTBL
AIC_END

;; End of AIC_Table.Inc
```

UART Program

Two types of signals are used in asynchronous serial port operations:

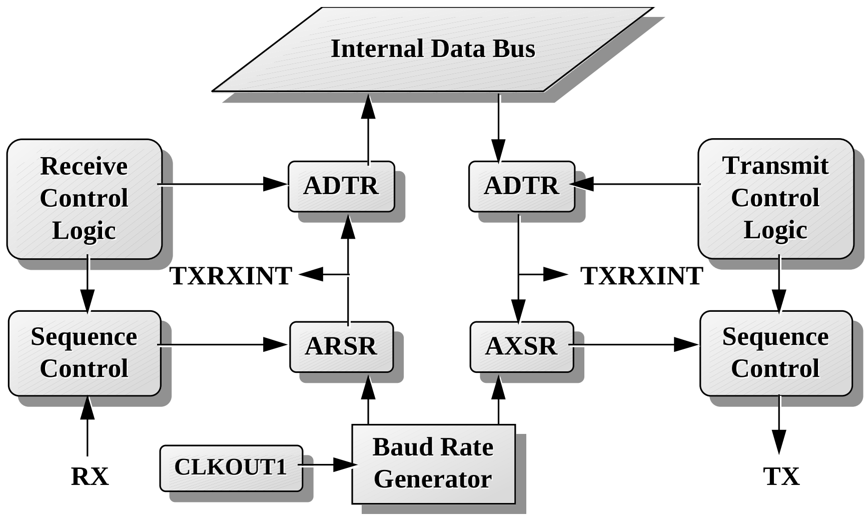
- ❑ Data signal
- ❑ Handshake signals

The data signal carries data from the transmitter to the receiver. One-way serial port transmission requires one data signal; two-way transmission requires two data signals.

In the TMS320C203 DSP, no handshake pins are supported; hence, IO0-IO3 are used as handshake signals. The signals allow the transmitter and receiver to control the time to transfer data.

Figure 16 shows the block diagram for the asynchronous serial port.

Figure 16. Asynchronous Serial Port Block Diagram



Configuring the asynchronous serial port requires two steps.

Step 1: You must select the configuration you want to use.

Step 2: The information must be written to the serial port control register.

In addition, you must set the baud rate by writing to the baud rate divisor register. Apply the equation

baud rate = (CLKOUT1 frequency) / (16 x BRD register)

to calculate the value for the baud rate divisor. (This item is a correction to the *TMS320C2xx User's Guide*.)

The UART.Asm program shown in Example 3 was written by Jeffrey Lai, a summer student from Stanford University, and modified by Art Chen. The function of the UART.Asm is to transmit the characters (from A to Z) to the PC host.

As is required for the Echo.ASM file, the interrupt vector table should be defined in program address from 0000h to 0040h, and the I/O wait-state and asynchronous serial port should be initialized. Then the data can be prepared for transmitting.

Example 3. UART.Asm

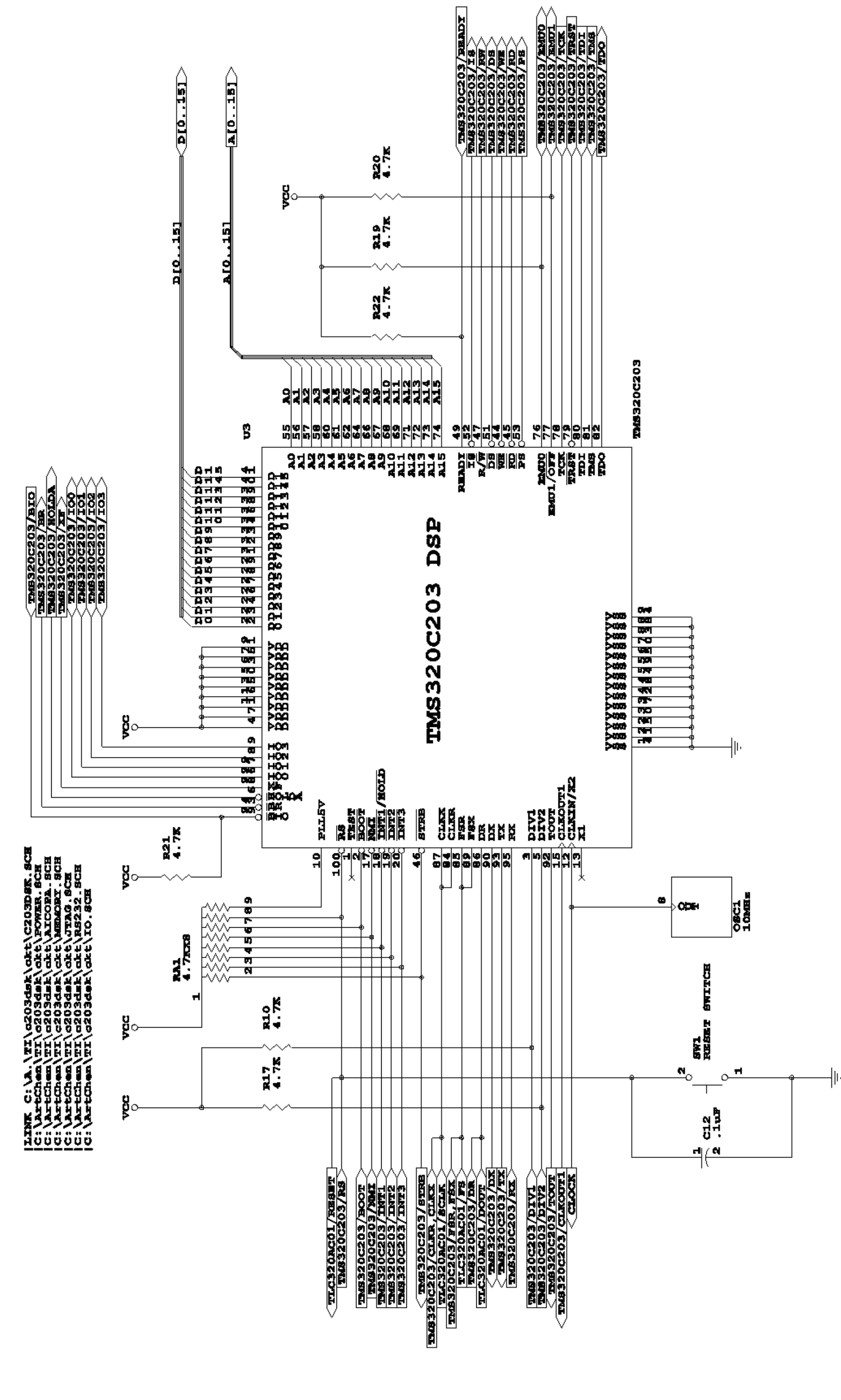
```
;;--------------------------------------------------------------------------------------------
;; Program Name: UART.Asm
;; Description:  C203 UART Program without Auto Baud Rate Detecting
;; Author:       Jeffrey Lai, a Summer Student from Stanford University
;; Modifier:     Art Chen in TI-Asia/DSP
;;--------------------------------------------------------------------------------------------

        .title  "UART Program"

TMVER   .equ    0                       ;; 0->TMX, 1->TMS

        .mmregs                         ;; Include Memory Mapped Registers

ADTR    .set    0FFF4H                  ;; Asynchronous Data Transmit/Receiver
ASPCR   .set    0FFF5H                  ;; Asynchronous SerialPort Control Reg.
IOSR    .set    0FFF6H                  ;; I/O Status Reg.
BRD     .set    0FFF7H                  ;; Baud Rate Divisor
WSGR    .set    0FFFCH                  ;; Wait-State Generator
IO0     .set    15 - 0                  ;; General IO Port 1st in IOSR
IO1     .set    15 - 1                  ;; General IO Port 2nd in IOSR
IO2     .set    15 - 2                  ;; General IO Port 3rd in IOSR
IO3     .set    15 - 3                  ;; General IO Port 4th in IOSR
DR      .set    15 - 8                  ;; Data Ready Indicator in IOSR
TEMT    .set    15 - 12                 ;; Transmit Empty Indicator in IOSR
DTR     .set    IO0                     ;; Data Terminal Ready
DSR     .set    IO1                     ;; Data Set Ready
RTS     .set    IO2                     ;; Request To Send
CTS     .set    IO3                     ;; Clear To Send


        .sect   "vectors"
        B       BEGIN                   ;; Reset Interrupt Vector
        .space  2 * 16                  ;; HOLD/INT1
        .space  2 * 16                  ;; INT2/INT3
        .space  2 * 16                  ;; Timer Interrupt ISR
        .space  2 * 16                  ;; SyncSerPort Receive ISR
        .space  2 * 16                  ;; SyncSerPort Transmit ISR
        B       TXRX_ISR                ;; AsyncSerPort Transmit/Receive ISR
        .space  2 * 16                  ;; ...>

        .bss    TEMP_DATA, 4
temp    .set    TEMP_DATA
test    .set    temp + 1
counter .set    test + 2

        .text
BEGIN:
        SETC    INTM                    ;; Disable Interrupt
        LDP     #TEMP_DATA

;; Set Wait State

        SPLK    #0E00h, temp            ;; 7-Wait State Set
        OUT     temp, WSGR              ;;

;; Set Baud Rate Divisor

        SPLK    #130, temp              ;; BRD = 130 = CLKOUT1(20Mhz) / 16 *
        OUT     temp, BRD               ;;  BaudRate(9600bps)
```

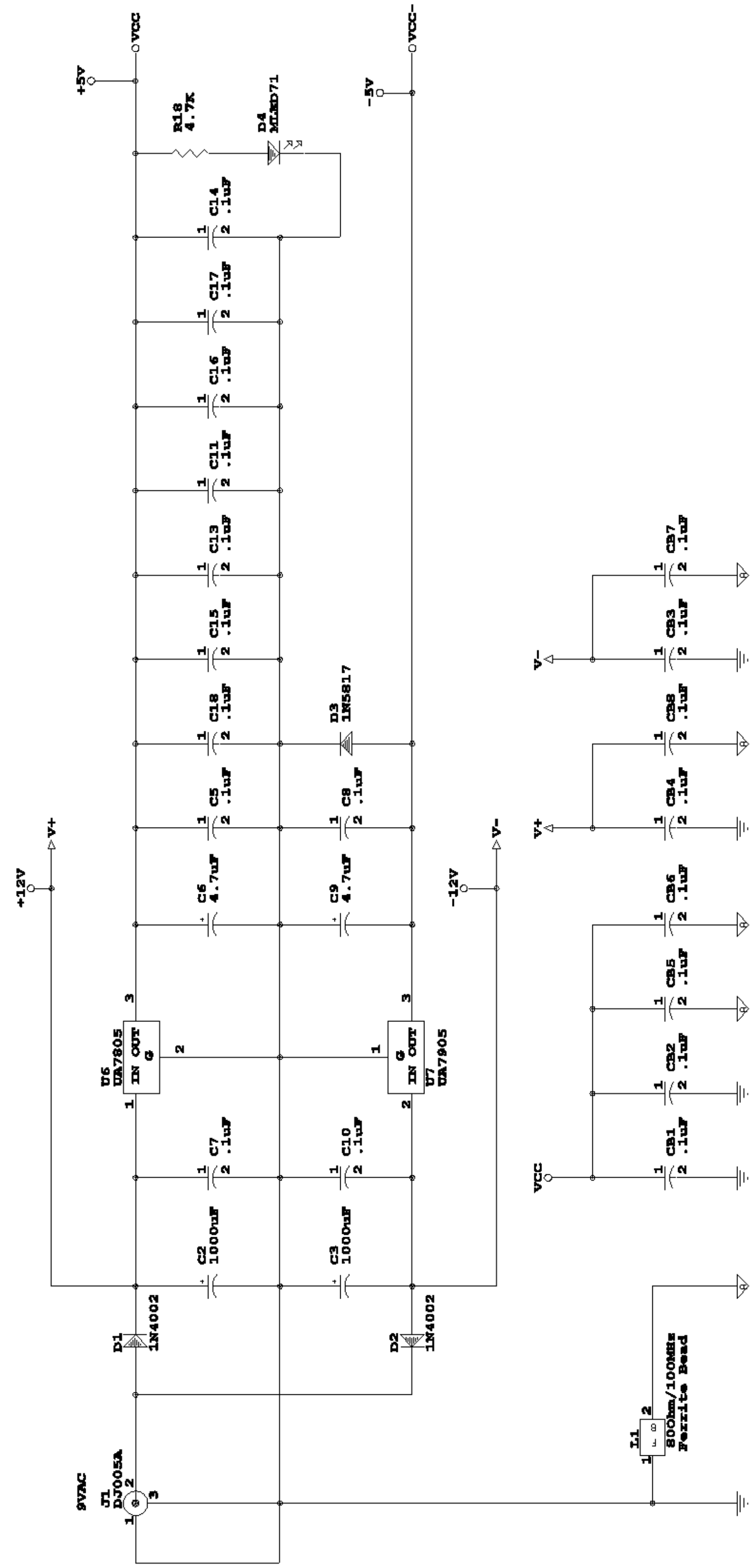
```
        SPLK    #0010000110001010b, temp        ;; Config IO0,1,2,3 and Disable
        OUT     temp, ASPCR                     ;; TX RX Mask, Reset, 1 StopBit

        IN      temp, IOSR                      ;; Set CTS, DSR to 0
        LACC    temp                            ;; and Clear ADC Bit
        AND     #1011111111110101b              ;;
        SACL    temp                            ;;
        OUT     temp, IOSR                      ;;

        SPLK    #64, counter                    ;; Initialize Data, counter = 64

WAIT_FOR_PC_READY:                              ;;          <----------------------\
        IN      test, IOSR                      ;; Wait for PC Ready               |
        BIT     test, DTR                       ;;                                 |
        BCND    WAIT_FOR_PC_READY, TC           ;;                                 |
        BIT     test, RTS                       ;;                                 |
        BCND    WAIT_FOR_PC_READY, TC           ;;          -----------------------/

        LDP     #0
        LACC    #IMR                            ;; Enable the TXRXINT in IMR
        OR      #20H                            ;;
        SACL    IMR                             ;;
        CLRC    INTM

LOOP:                                           ;; Waiting for Interrupt      <----------\
        NOP                                     ;;                                        |
        B       LOOP                            ;;          ------------------------------/

TXRX_ISR:
        SETC    INTM                            ;; Disable Interrupt
        LDP     #0
        .IF     TMVER = 1                       ;; If TMP Chip is used
        LACL    #00020H                         ;; Clear the TXRXINT Flag
        .ENDIF
        .IF     TMVER = 0                       ;; If TMX Chip is used
        LACC    #IFR                            ;; Clear the TXRXINT Flag
        AND     #0FFDFH                         ;;
        .ENDIF
        SACL    IFR                             ;;

        LDP     #TEMP_DATA
        IN      test, IOSR
        BIT     test, DR                        ;; If Data is Ready?
        BCND    RX_DATA, TC                     ;; To Receive Data
        BIT     test, TEMT                      ;; If Transmit is Empty?
        BCND    TX_DATA, TC                     ;; To Transmit Data
        CLRC    INTM                            ;; Enable Interrupt
        RET                                     ;; Nothing Held

TX_DATA:
        LACC    counter                         ;; counter ++ as the Data
        ADD     # 1                             ;;
        SACL    counter                         ;;
        OUT     counter, ADTR                   ;; Transmit Data
        SUB     #90                             ;; If(counter > 90)
        BCND    TX_RET, LT                      ;;  count = 64
        SPLK    #64, counter                    ;;
TX_RET:
        CLRC    INTM                            ;; Enable Interrupt
        RET

RX_DATA:
```

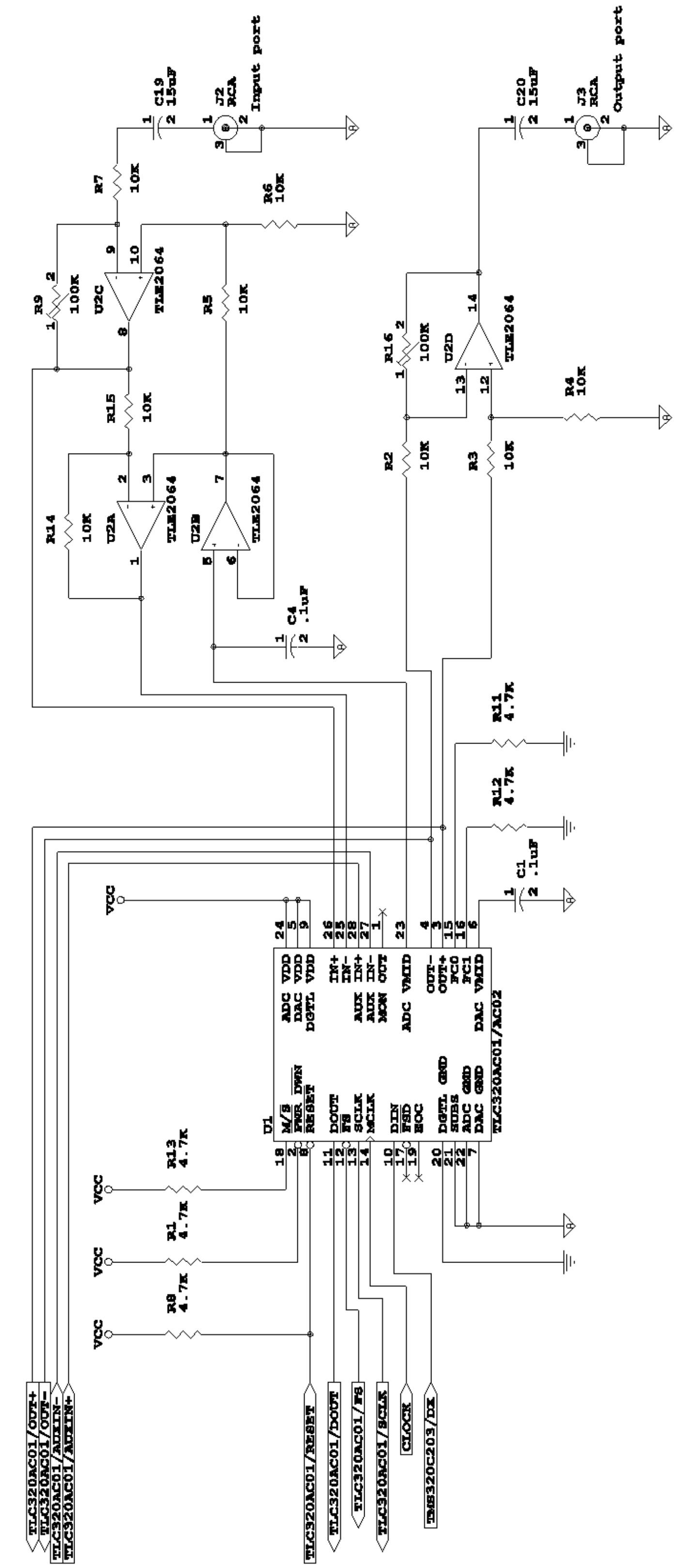
```
        CLRC    INTM                ;; Enable Interrupt
        RET

;; End of UART.Asm
```

Appendix A. TMS320C203 Development Board Schematics

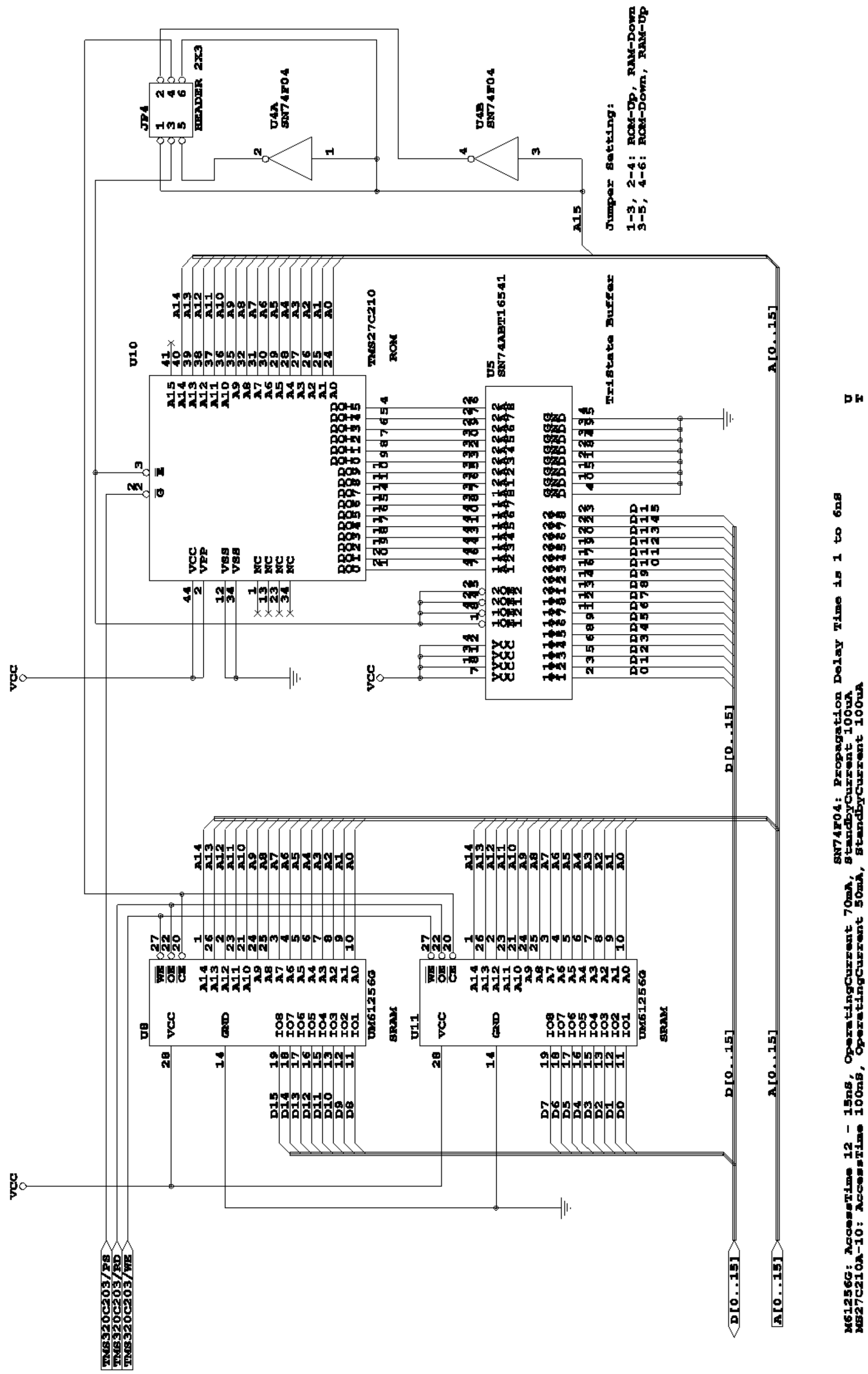
TMS320C203 DSP and Clock



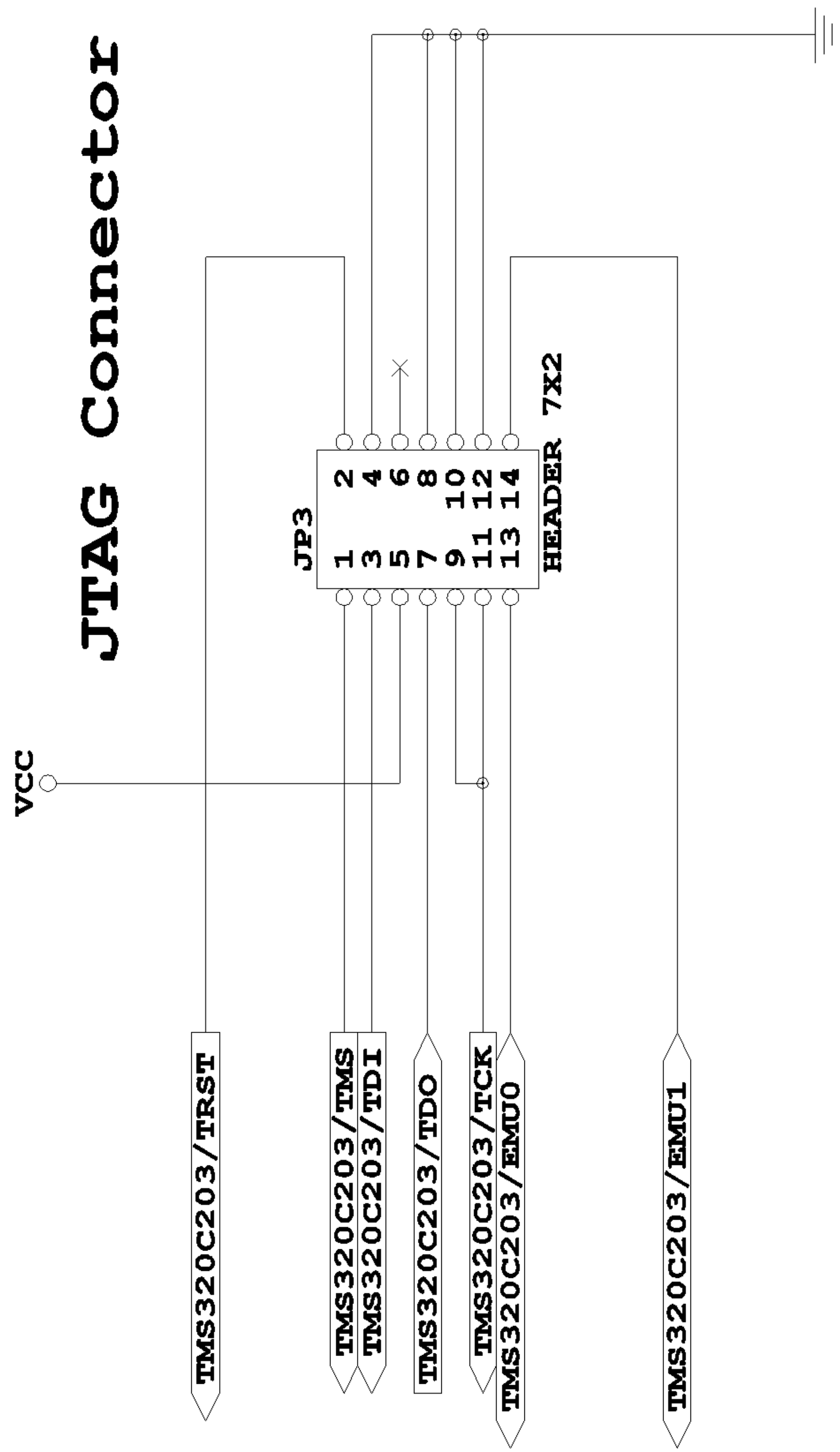

VCC
VCC-
+5V
-5V
R18 4.7K
D4 MLED71
C14 .1uF
C17 .1uF
C16 .1uF
C11 .1uF
C13 .1uF
C15 .1uF
C18 .1uF
D3 1N5817
C5 .1uF
C8 .1uF
C6 4.7uF
C9 4.7uF
V+
V-
+12V
-12V
U6 UA7805
IN OUT G
U7 UA7905
G IN OUT
C7 .1uF
C10 .1uF
C2 1000uF
C3 1000uF
D1 1N4002
D2 1N4002
J1 DJ005A
9VAC
CB7 .1uF
CB3 .1uF
CB8 .1uF
CB4 .1uF
CB6 .1uF
CB5 .1uF
CB2 .1uF
CB1 .1uF
L1
80Ohm/100MHz Ferrite Bead


Input port
Output port
J2
RCA
J3
RCA
C19
15uF
C20
15uF
C4
.1uF
C1
.1uF
R1
R2
R3
R4
R5
R6
R7
R8
R9
R11
R12
R13
R14
R15
R16
10K
100K
4.7K
U2A
U2B
U2C
U2D
TLE2064
U1
TLC320AC01/AC02
VCC
TLC320AC01/OUT+
TLC320AC01/OUT-
TLC320AC01/AUXIN-
TLC320AC01/AUXIN+
TLC320AC01/RESET
TLC320AC01/DOUT
TLC320AC01/FS
TLC320AC01/SCLK
CLOCK
TMS320C203/DX

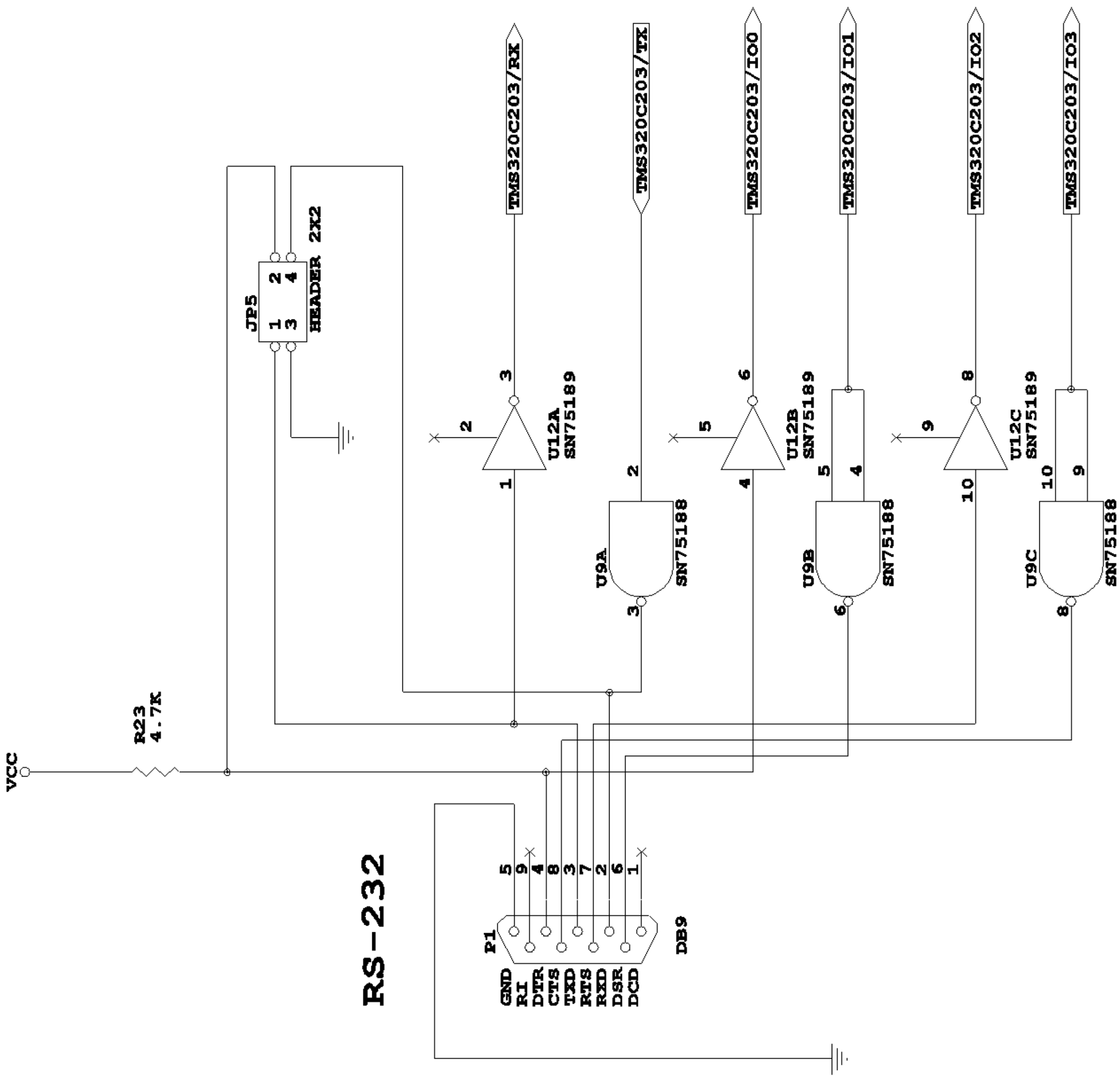
TMS320C203 Development Board External Memory System



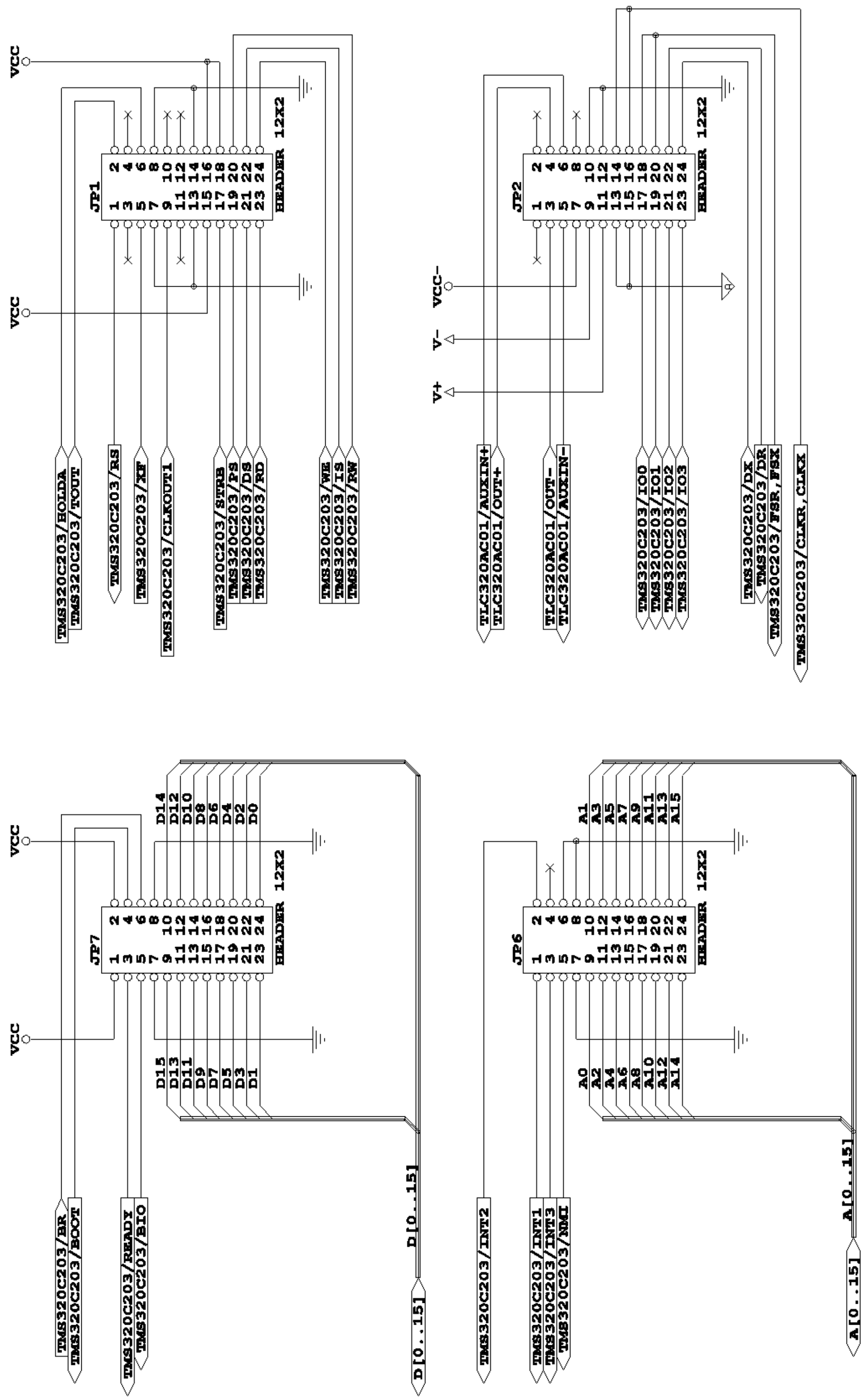
TMS320C203 Development Board JATG Connector



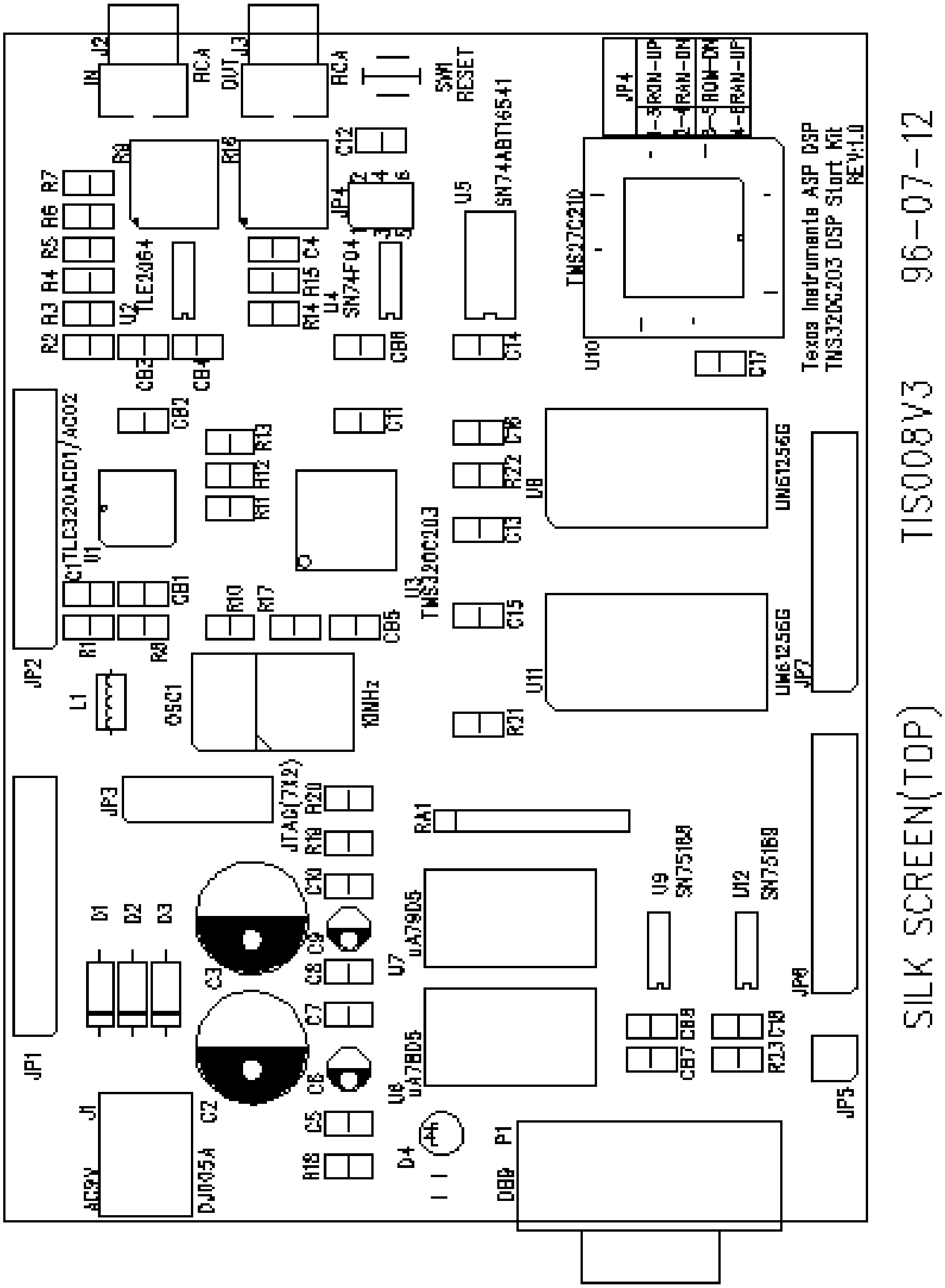
TMS320C203 Development Board RS-232 Connector



TMS320C203 Development Board I/O Connectors



TMS320C203 Development Board Placement



Appendix B. Bill of Materials

Item	Quantity	Reference	Part
1	22	CB1, C1, CB2, CB3, CB4, C4, CB5, C5, CB6, CB7, C7, CB8, C8, C10, C11, C12, C13, C14, C15, C16, C17, C18	0.1 µF
2	2	C2, C3	1000 µF
3	2	C9, C6, C19, C20	4.7 µF
4	2	D1, D2	1N4002
5	1	D3	1N5817
6	1	D4	MLED71
7	4	JP1,JP2,JP6,JP7	HEADER 12x2
8	1	JP3	HEADER 7x2
9	1	JP4	HEADER 2x3
10	1	JP5	HEADER 2x2
11	1	J1	DJ005A
12	2	J2, J3	RCA
13	1	L1	80 Ohm/100 MHz
14	1	OSC1	10 MHz
15	1	P1	DB9
16	1	RA1	4.7 Kx8
17	13	R1, R8, R10, R11, R12, R13, R17, R18, R19, R20, R21, R22, R23	4.7 K
18	8	R2, R3, R4, R5, R6, R7, R14, R15	10 K
19	2	R9, R16	100 K
20	1	SW1	Reset Switch
21	1	U1	TLC320AC01/AC02
22	1	U2	TLE2064
23	1	U3	TMS320C203
24	1	U4	SN74F04
25	1	U5	SN74ABT16541
26	1	U6	UA7805
27	1	U7	UA7905
28	2	U11, U8	UM61256G
29	1	U9	SN75188
30	1	U10	TMS27C210
31	1	U12	SN75189