

ADuC8xxファミリーおよびADuC702xファミリーでのDAC出力数の拡張

著者：Aude Richard

概要

アナログ・デバイセズのMicroConverter®ファミリー製品は、1つまたは複数の電圧出力DACを内蔵しています。

アプリケーションによっては、アナログ出力チャンネルの追加が必要になることがあります。このため、MicroConverterはディスクリートの外部マルチチャンネルDACへインターフェースする必要があります。MicroConverterをアナログ・デバイセズの小型フットプリントを持つSPI®互換のマルチチャンネルDACファミリーへインターフェースすると、システム内で使用可能な電圧出力チャンネル数を容易に増やすことができます。

このアプリケーション・ノートでは、シンプルな3線式SPI互換インターフェースを使って、ADuC814またはADuC7020を

AD5304/AD5314/AD5324 (AD53x4) クワッド電圧出力8/10/12ビットDACへインターフェースする方法を説明します。

AD53x4

AD53x4 (図1) は、クワッド8/10/12ビット・バッファ付き電圧出力DACであり、10ピンのmicro SOICパッケージを採用しています。これらのデバイスは、2.5V~5.5Vの単電源で動作し、消費電流は3Vで500μAです。内蔵の出力アンプは、スルーレート0.7V/μsでレールtoレールの出力振幅を実現しています。内蔵の3線式シリアル・インターフェースは、最大クロック・レート30MHzで動作し、標準SPIインターフェース規格と互換性を持っています。詳細については、AD5304データシートを参照してください。

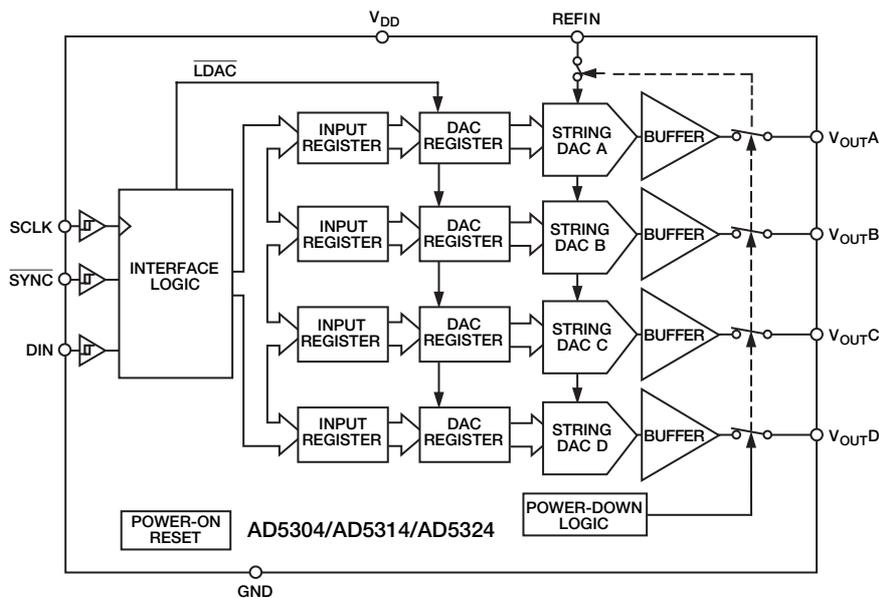


図1. AD53x4の機能ブロック図

REV. 0

アナログ・デバイセズ株式会社

本 社 / 〒105-6891 東京都港区海岸1-16-1 ニューピア竹芝サウスタワービル
 電話03(5402)8200
 大阪営業所 / 〒532-0003 大阪府大阪市淀川区宮原3-5-36 新大阪MTビル2号
 電話06(6350)6868

ハードウェア・インターフェース

SPIインターフェース

図2に、AD53x4とMicroConverterAD5641との間のシリアル・インターフェースを示します。AD53x4はスレーブ・デバイスであるため、MicroConverterはマスターとして設定され、AD53x4のSCLKを駆動します。MOSI（マスター出力スレーブ入力）ピンは、DACのシリアル・データ・ライン（DIN）に接続されます。実質的にチップ・セレクト入力であるAD53x4のSYNC入力は、ADuC814（P3.4）のポートのビット・プログラマブル・ピンまたはADuC7020（P1.7）の汎用I/Oから駆動されます。

AD53x4の入力シフトレジスタは、16ビット幅で、データは16ビット・ワードとしてデバイスにロードされます。SYNCがローレベルのとき、MicroConverterからの2バイトのデータの到着を待ちます。

ADuC814には、スレーブ・セレクトまたはチップ・セレクト出力がないため、GPIOを使う必要があります。ADuC7020では、SPIインターフェースに各バイトの送信後にハイレベルなるスレーブ・セレクトまたはチップ・セレクト・ラインがあるため、外部DACのSYNCラインを制御するGPIOが必要です。データはMSBファーストで転送されます。

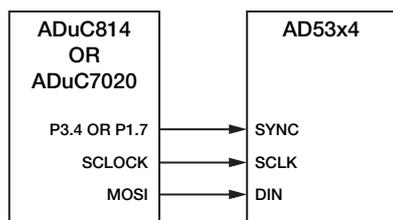


図2. AD53x4とADuC814/ADuC7020とのインターフェース

MICROCONVERTERから供給されるリファレンス電圧

必要に応じて、MicroConverterの内部リファレンス電圧をAD53x4に供給することができます。

- ADuC814は、2.5Vの高精度バンド・キャップ・リファレンス電圧を提供します。内部の2.5Vは、絶対精度 $2.5V \pm 2.5\%$ に出荷時にキャリブレーションされています。この内部リファレンス電圧をAD53x4に対して使用する場合は、CREFピンとAGNDとの間でバッファする必要があります（図3）。5V電源での内部リファレンス電圧のノイズ性能（代表値）は、1kHzで150nV/Hzであり、DCノイズは100mV p-pです。

- ADuC7020は2.5Vのオンチップ・リファレンスを提供し、出荷時に絶対精度 $2.5V \pm 10mV$ にキャリブレーションされています。この内部リファレンスをAD53x4に対して使用する場合は、VREFピンでバッファし、470nFのコンデンサをこのピンとAGNDとの間に接続する必要があります。

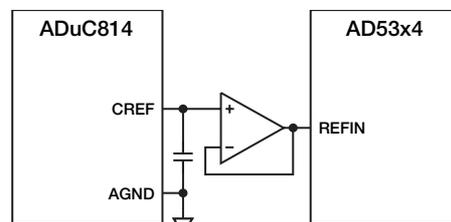


図3. ADuC814リファレンス電圧出力の使用

ADuC814用のソフトウェア・インターフェース

リスト1は、AD53x4のインターフェースに関係する部分をコンパニオン・コードから抜粋したものです。

SPIインターフェースはinit814関数で初期化されて、AD53x4のSPIタイミング条件を満たす種々のモード・パラメータを設定します。

ad53x4outはインターフェース関数であり、まず、AD53x4の入力シフトレジスタの上位バイトにロードされる与えられたパラメータの16ビット・データの上位バイトをフォーマットします（フォーマットの詳細についてはAD5304/AD5314/AD5324のデータシートを参照してください）。

次にこのバイトはspiTx関数を使って送信されます。この後に、下位バイトが送信されて16ビット・データの転送が完了します。AD53x4のSYNCであるad53x4csは、SPIが連続する2バイトを送信する間、アサート状態が維持されることに注意してください。

ad53x4outから呼び出されるspiTx関数は、txDatパラメータに含まれているバイト・データをSPIDATレジスタへ書き込むことにより、このバイト・データを送信します。その後、この関数は、ISPIフラグがセットされて送信が完了したことが表示されるのを待ちます。送信ビジー状態ではISPIビットがハードウェアにより自動的にリセットされないため、SPIDATへの書き込みの前にソフトウェアからセットする必要があります。また、spiTx関数は、送信の完了を確認した後に呼び元を終了させて、データ転送中ではなくデータ転送後に、呼び出し元にSYNCを確実に反転させるようにする必要があります。

```

sbit ad53x4cs = P3^4; // Chip select for ad53x4 (PORT3.4)
void init814(void) // Initialize internal peripherals
{
    /* Initialize other peripherals here */
    /* Initialize SPI to talk to AD53x4 */
    CFG814 = 0x01; // Serial interface enable for P3.5..P3.7 pins
    SPICON = 0x38; // Enable SPI I/F as master, SCLOCK idle H,
                  // advance MSB output, sclock=fcore/2=1.05Mhz
}
void spiTx(unsigned char txDat) // Transmit a byte data over the SPI
{
    ISPI = 0; // Clear ISPI bit
    SPIDAT = txDat;
    while(!ISPI); // Wait until tx complete
}
void ad53x4out(unsigned char adrs, // A1,A0 bit of the contents
               bit pdN, // PD bit of the contents
               bit ldacN, // LDAC bit of the contents
               unsigned short dat) // 12-bit output data
{
    unsigned char txDat;
    ad53x4cs = 0;

    txDat = ((unsigned char) (dat>>8)) & 0x0f;
    txDat |= ldacN ? 0x10 : 0x00;
    txDat |= pdN ? 0x20 : 0x00;
    txDat |= (adrs<<6);
    spiTx(txDat); // Tx the upper byte

    txDat = (unsigned char) dat;
    spiTx(txDat); // Tx the lower byte

    ad53x4cs = 1;
}

```

リスト1. コンパニオン・コード53x4forADuC814.Cからの関連部分の抜粋

ADuC7020用のソフトウェア・インターフェース

リスト2は、AD53x4とADuC7020とのインターフェースに関係した部分をコンパニオン・コードADuC7020toAD53x4.cから抜粋したものです。

SPIインターフェースはinit702x関数で初期化されて、AD53x4のSPIタイミング条件を満たす種々のモード・パラメータを設定します。

AD53x4outはインターフェース関数であり、まず、AD53x4の入力シフトレジスタの上位バイトにロードされる与えられたパラメータの16ビット・データの上位バイトをフォーマットします（フォーマットの詳細についてはAD5304/AD5314/AD5324のデータシートを参照してください）。

次にこのバイトはspiTx関数を使って送信されます。この後に、下位バイトが送信されて16ビット・データの転送が完了します。

AD53x4のSYNCであるP1.7は、SPIが連続する2バイトを送信する間、アサート状態が維持されることに注意してください。

ad53x4outから呼び出されるspiTx関数は、txDatパラメータに含まれているバイト・データをSPITXレジスタへ書き込むことにより、このバイト・データを送信します。その後、この関数は、SPISTAのSPIRXフル・ビット（ビット4）がセットされて送信が完了したことが表示されるのを待ちます。SPIRXのダミー・データを読み出すことにより、ソフトウェアからSPIRXフル・ビットをクリアする必要があることに注意してください。

```

void init702x(void){
    GP1DAT = 0x80800000;           // Configure P1.7 as output
    GP1SET = 0x00800000;           // and pull /SYNC High
    GP1CON = 0x02020000;           // Configure P1.6 and P1.4 in SPI mode
    SPICON = 0x4B;                 // Configure SPI as Master, clock idles high
    SPIDIV = 0x14;                 // Set the DIV to run at 1.05Mhz
}

void spiTx(unsigned char txDat){
    SPITX = txDat;                 // Send txDat to the AD5304
    while(!(SPISTA & 0x08));        // Wait of end of transfer
    txDat = SPIRX;                 // Read in dummy data from SPIRX to prevent overflow
}

void ad53x4out(unsigned short address, unsigned short pdN, unsigned short ldacN, unsigned short dat){
    txDat1 = 0;                    // Reset txDat1 to 0
    txDat0 = 0;                    // Reset txDat0 to 0

    //configure the high byte to be sent
    txDat1 |= (address << 6);      // Store the address in the upper 2 bits of txDat1 (8 bit)
    txDat1 |= pdN ? 0x20 : 0x00;    // If pdN is set, copy a logic 1 into the next space,
    // if not copy a logic 0
    txDat1 |= ldacN ? 0x10 : 0x00;  // If ldacN is set, copy a 1 into the next space, if not copy a 0
    txDat1 |= (dat >> 4);          // Copy the first 4 bits of data into the last 4 bits of txDat1

    //configure the low byte to be sent
    txDat0 |= (dat << 4);          // Copy the last 4 bits of data into the first 4 bits of txDat0
    txDat0 &= 0xf0;                // Insure the first 4 bits of data in txDat0 are empty

    //send the data
    GP1CLR = 0x800000;             // Pull /SYNC High
    spiTx(txDat1);                 // Send txDat1 to the SPI
    spiTx(txDat0);                 // Send txDat0 to the SPI
    GP1SET = 0x800000;             // Pull /SYNC Low
}

```

リスト2. コンパニオン・コードADuC7020to53x4.cからの関連部分の抜粋