

セクション 33. オーディオ D/A コンバータ (DAC)

ハイライト

本セクションには以下の主要項目を記載しています。

33.1 はじめに	33-2
33.2 主な特長	33-3
33.3 DAC レジスタ	33-3
33.4 モジュールの動作	33-7
33.5 割り込みとステータス	33-10
33.6 DMA なしでのオーディオ DAC 動作	33-11
33.7 DMA ありでのオーディオ DAC 動作	33-13
33.8 外付け回路の例	33-16
33.9 省電力モード時の動作	33-17
33.10 レジスタマップ	33-18
33.11 関連アプリケーション ノート	33-19
33.12 改訂履歴	33-20

33.1 はじめに

オーディオ D/A コンバータ (DAC) モジュールは、オーディオ アプリケーション向けに設計された 16 ビット Δ - Σ 型信号コンバータです。2 つの出力チャンネルがステレオ動作をサポートします。データ入力は、DMA モジュールまたは DAC データおよび制御レジスタを介してアプリケーション プログラムから出力された 16 ビットデジタル値の形式です。データ出力はアナログ電圧であり、デジタル入力値に比例します。

各出力チャンネルは以下の 3 つの電圧出力を供給します。

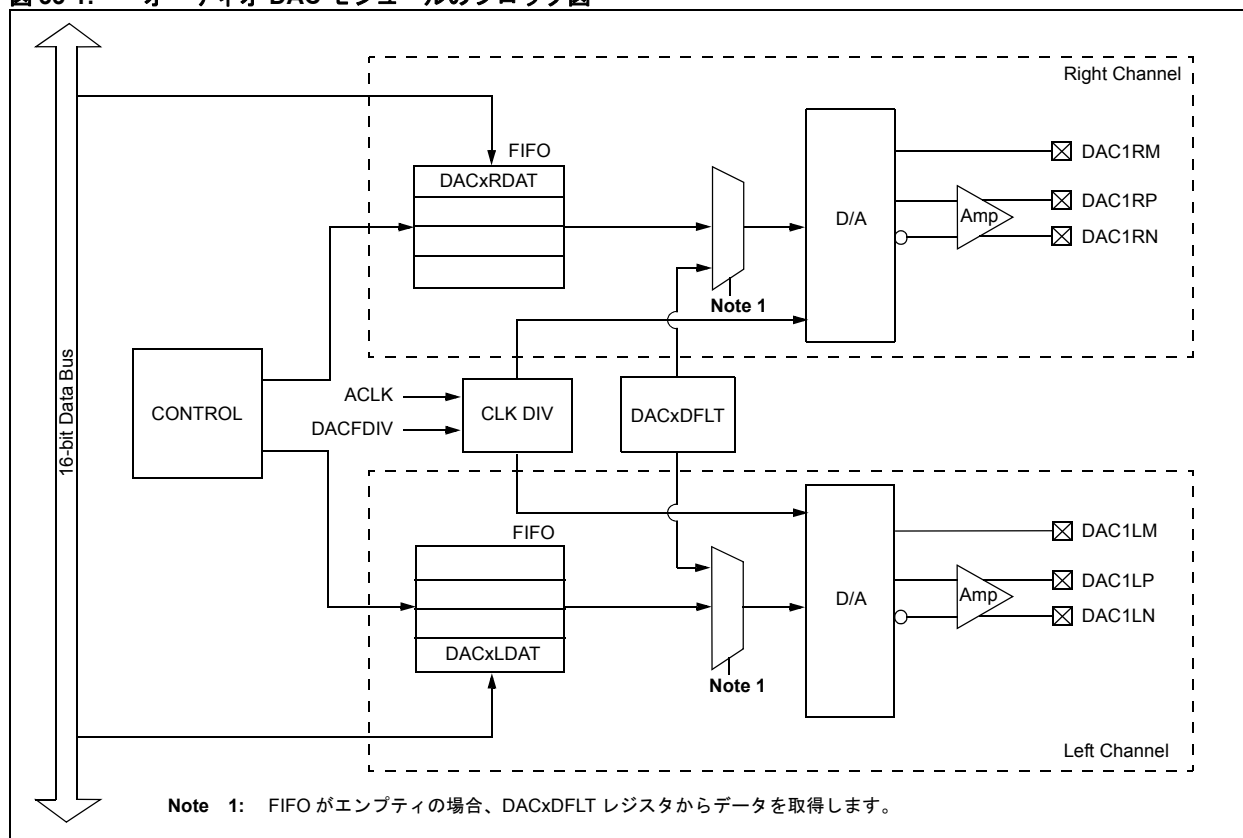
- 正の DAC 出力
- 負の DAC 出力
- 中間電圧出力 (一部のデバイスでは未実装)

中間出力は出力電圧レンジの中間点を表すオフセット電圧レベルです。

図 33-1 に、オーディオ DAC の概略ブロック図を示します。4 ワード段の FIFO はチャンネルごとにデータ入力をバッファします。いつでも FIFO が空 (エンプティ) になると (例えば、DMA モジュールまたはプロセッサがタイミングよくデータを供給できない場合)、DAC は DAC 既定値データレジスタ (DACxDFLT) からの代替データを受け入れます。このレジスタは「安全な」出力電圧 (多くの場合、中間値またはゼロ) を表す既定値の入力値を示します。

DAC のサンプルレートは補助オシレータまたはシステムクロックのレートを分周回路によって整数分周して設定します。分周比は DAC 制御レジスタ (DACxCON) の DAC クロック分周器 (DACFDIV<6:0>) のコンフィグレーション ビットで指定します。

図 33-1: オーディオ DAC モジュールのブロック図



33.2 主な特長

オーディオ DAC の主な特長は以下の通りです。

- 16 ビット分解能 (14 ビット精度)
- 2 次デジタル Δ - Σ 変調器
- 256 倍のオーバーサンプリング比
- 最大 100 ksps のサンプリング レート
- ユーザが制御可能なサンプルクロック
- 45 kHz の最大入力信号周波数
- 差動アナログ出力
- 4 ワード段の入力バッファ
- 16 ビットのプロセッサ I/O インターフェイスと DMA インターフェイス

Note: 本モジュールはオーディオ アプリケーション専用設計されています。このモジュールを制御ループ型のアプリケーションに使用するのには推奨しません。

33.3 DAC レジスタ

本モジュールは 5 つの DAC レジスタで制御します。

- **DACxCON: DAC 制御レジスタ**

このレジスタは、対応する DAC モジュールの有効化 / 無効化、データ形式、DAC フィルタ クロック分周器、アイドル / スリープモードでの動作を指定して、DAC モジュールを設定します。

- **DACxSTAT: DAC ステータスおよび制御レジスタ**

このレジスタは有効にするチャンネルと、そのチャンネルのデータバッファのステータスを指定します。

- **DACxDFLT: DAC 既定値データレジスタ**

このレジスタはFIFOがエンプティの時に入力として使用されるDAC既定値を指定します。

- **DACxLDAT: DAC 左側チャンネル データレジスタ**

このレジスタは左側チャンネルのデータを指定します。

- **DACxRDAT: DAC 右側チャンネル データレジスタ**

このレジスタは右側チャンネルのデータを指定します。

dsPIC33F ファミリ リファレンス マニュアル

レジスタ 33-1: DACxCON: DAC 制御レジスタ

R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0
DACEN	—	DACSIDL	AMPON	—	—	—	FORM
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1	
—	DACFDIV<6:0>							
bit 7								bit 0

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 -n = POR 時の値 「1」= ビットをセット 「0」= ビットをクリア x = ビットは未知

- bit15 **DACEN:** DAC イネーブルビット
 1 = モジュールを有効にする
 0 = モジュールを無効にする
- bit14 **未実装:** 「0」として読み出し
- bit13 **DACSIDL:** Stop-in-Idle モードビット
 1 = デバイスがアイドルモードになるとモジュールの動作を停止する
 0 = アイドルモード時でもモジュールの動作を継続する
- bit12 **AMPON:** スリープ /Stop-in-Idle モード中アナログ出力アンプ イネーブルビット
 1 = スリープモード /Stop-in-Idle モード中にアナログ出力アンプを有効にする
 0 = スリープモード /Stop-in-Idle 中にアナログ出力アンプを無効にする。全てのチャンネルが
 リセットされる
- bit11-9 **未実装:** 「0」として読み出し
- bit8 **FORM:** データ形式選択ビット
 1 = 符号付き整数
 0 = 符号なし整数
- bit7 **未実装:** 「0」として読み出し
- bit6-0 **DACFDIV<6:0>:** DAC クロック分周器ビット
 1111111 = 入力クロックを 128 分周する
 .
 .
 .
 .
 0000101 = 入力クロックを 6 分周する (既定値)
 .
 .
 .
 .
 0000010 = 入力クロックを 3 分周する
 0000001 = 入力クロックを 2 分周する
 0000000 = 入力クロックを 1 分周する (分周しない)

セクション 33. オーディオ D/A コンバータ (DAC)

レジスタ 33-2: DACxSTAT: DAC ステータスおよび制御レジスタ

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R-0	R-0
LOEN	—	LMVOEN	—	—	LITYPE	LFULL	LEMPY
bit 15						bit 8	

R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R-0	R-0
ROEN	—	RMVOEN	—	—	RITYPE	RFULL	REMPY
bit 7						bit 0	

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
 -n = POR 時の値 「1」= ビットをセット 「0」= ビットをクリア x = ビットは未知

- bit15 **LOEN:** 左側チャンネル DAC 出力イネーブルビット
 1 = 正および負の DAC 出力を有効にする
 0 = DAC 出力を無効にする
- bit14 **未実装:** 「0」として読み出し
- bit13 **LMVOEN:** 左側チャンネル中間 DAC 出力電圧イネーブルビット
 1 = 中間 DAC 出力を有効にする
 0 = 中間 DAC 出力を無効にする
- bit 12-11 **未実装:** 「0」として読み出し
- bit10 **LITYPE:** 左側チャンネル割り込みタイプビット
 1 = FIFO がエンプティの場合に割り込み
 0 = FIFO がフルでない場合に割り込み
- bit9 **LFULL:** 左側チャンネルデータ入力 FIFO フル ステータスビット
 1 = FIFO はフルである
 0 = FIFO はフルでない
- bit8 **LEMPY:** 左側チャンネルデータ入力 FIFO エンプティ ステータスビット
 1 = FIFO はエンプティである
 0 = FIFO はエンプティでない
- bit7 **ROEN:** 右側チャンネル DAC 出力イネーブルビット
 1 = 正および負の DAC 出力を有効にする
 0 = DAC 出力を無効にする
- bit6 **未実装:** 「0」として読み出し
- bit5 **RMVOEN:** 右側チャンネル中間 DAC 出力電圧イネーブルビット
 1 = 中間 DAC 出力を有効にする
 0 = 中間 DAC 出力を無効にする
- bit4-3 **未実装:** 「0」として読み出し
- bit2 **RITYPE:** 右側チャンネル割り込みタイプビット
 1 = FIFO がエンプティの場合に割り込み
 0 = FIFO がフルでない場合に割り込み
- bit1 **RFULL:** 右側チャンネルデータ入力 FIFO フル ステータスビット
 1 = FIFO はフルである
 0 = FIFO はフルでない
- bit0 **REMPY:** 右側チャンネルデータ入力 FIFO エンプティ ステータスビット
 1 = FIFO はエンプティである
 0 = FIFO はエンプティでない

dsPIC33F ファミリ リファレンス マニュアル

レジスタ 33-3: DACxDFLT: DAC 既定値データレジスタ

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DACDFLT<15:8>							
bit 15				bit 8			

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DACDFLT<7:0>							
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
-n = POR 時の値 「1」= ビットをセット 「0」= ビットをクリア x = ビットは未知

bit15-0 **DACDFLT:** DAC 既定値ビット

レジスタ 33-4: DACxLDAT: DAC 左側チャンネル データレジスタ

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DACLDAT<15:8>							
bit 15				bit 8			

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DACLDAT<7:0>							
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
-n = POR 時の値 「1」= ビットをセット 「0」= ビットをクリア x = ビットは未知

bit15-0 **DACLDAT:** 左側チャンネル データビット

レジスタ 33-5: DACxRDAT: DAC 右側チャンネル データレジスタ

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DACRDAT<15:8>							
bit 15				bit 8			

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DACRDAT<7:0>							
bit 7				bit 0			

凡例:

R = 読み出し可能ビット W = 書き込み可能ビット U = 未実装ビット、「0」として読み出し
-n = POR 時の値 「1」= ビットをセット 「0」= ビットをクリア x = ビットは未知

bit15-0 **DACRDAT:** 右側チャンネル データビット

33.4 モジュールの動作

図 33-2 に、D/A 変換の処理を示します。デジタル補間フィルタは入力信号をアップ サンプリングして追加の補間データ点を作成します。オーバーサンプリング比は 256:1、つまり入力の入力サンプリング レートの 256 倍です。例えば、100 ksps の入力信号 (最大サンプリング レート) は 1 秒あたり 25.6M のデータ点を生成します。

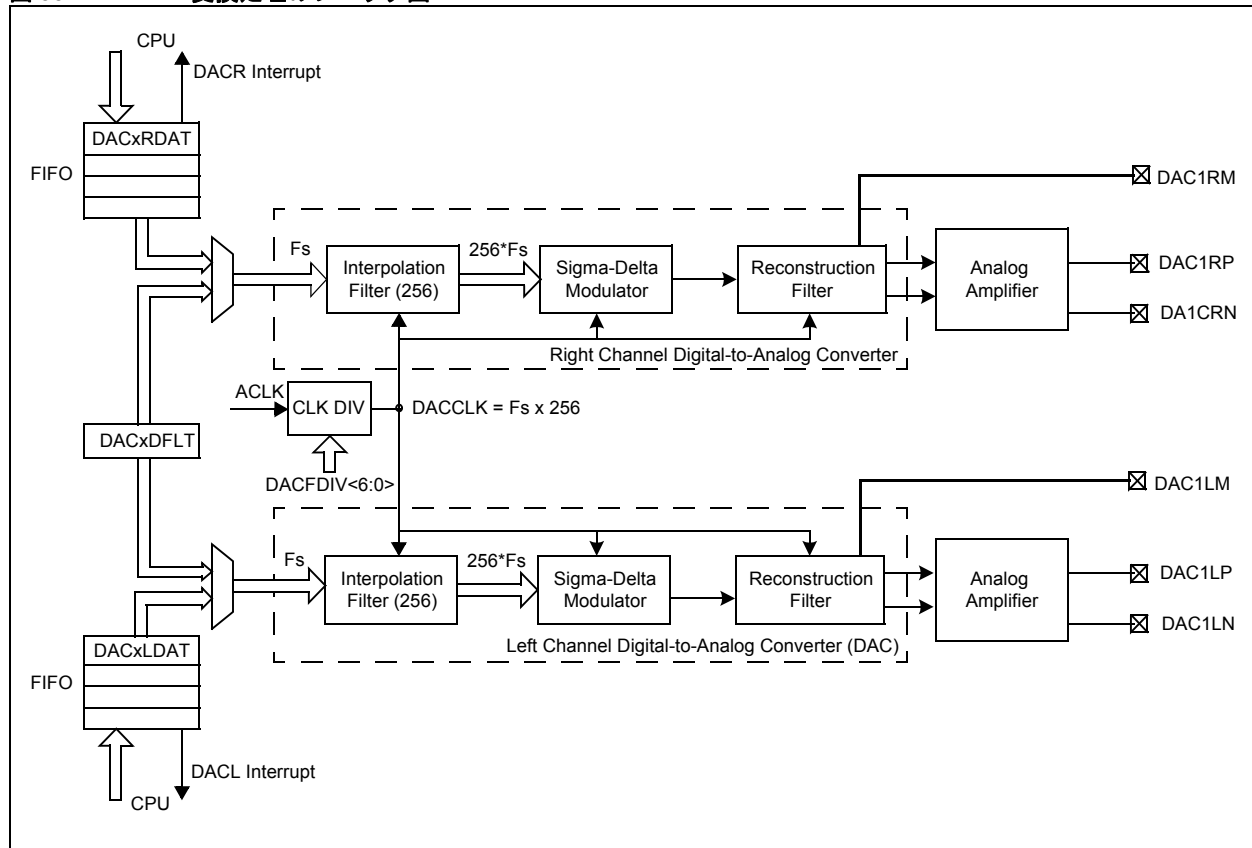
補間フィルタはアップ サンプリング処理によって発生する不要ノイズも除去します。

補間フィルタの出力は Δ - Σ 変調器を駆動し、これが補間フィルタからのワード出力をシリアル ビットストリームに変換します。

変調器からのビットストリームは再構成フィルタによって処理され、ビットストリームはアナログ信号に変換されます。その後、再構成フィルタはローパスフィルタを適用して所定の電圧レベルを出力します。再構成フィルタは以下に示す 2 つの差動電圧出力と中間基準電圧を生成します。

- ・ 正 (出力信号を表す電圧レベル)
- ・ 負 (正の出力信号電圧レベルの補数)
- ・ 中間 (出力電圧レンジの中間を表すオフセット電圧レベル)

図 33-2: D/A 変換処理のブロック図



33.4.1 データ形式

DAC モジュールは 2 つの形式の 16 ビット入力データをサポートしています。データ形式は DAC 制御 (DACxCON<8>) レジスタのデータ形式選択 (FORM) ビットで設定します。サポートされている形式は以下の通りです。

- 1 = 符号付き (2 の補数)
- 0 = 符号なし

FORM ビットを符号なしデータに設定すると (FORM = 0)、ユーザ入力データによって以下の動作が実行されます。

- 0xFFFF = 正の最大出力電圧
- 0x8000 = 中間出力電圧
- 0x7FFF = 中間値をわずかに下回る値
- 0x0000 = 最小出力電圧

FORM ビットを符号付きデータに設定すると (FORM = 1)、ユーザ入力データによって以下の動作が実行されます。

- 0x7FFF = 正の最大出力電圧
- 0x0000 = 中間出力電圧
- 0xFFFF = 中間値をわずかに下回る値
- 0x8000 = 最小出力電圧

33.4.2 クロック

DAC クロック (DACCLK) はサンプリング レートの 256 倍に等しくする必要があります。入力を 100 ksps とすると、DAC クロックレートは 25.6 MHz (100,000 x 256 = 25,600,000) である必要があります。

DACCLK は高速オシレータ (補助クロックまたはシステムクロック) を分周する事によって生成します。分周比は DAC 制御 (DACxCON<6:0>) レジスタのクロック分周器ビット (DACFDIV<6:0>) で指定します。結果として得られる DACCLK は 25.6 MHz を超えてはなりません。

dsPIC® DSC がクロック源としてデバイスの FRC を使用すると仮定してみましょう。デバイスオシレータの前置 PLL 分周器 (N1) と後置 PLL 分周器 (N2) は 2 分周するよう設定します。オシレータの選択の詳細は、**セクション 39. 「オシレータ (パート III)」** (DS70216) を参照してください。DAC モジュールは 8 kHz でサンプリングされた信号を出力するよう設定する事が求められます。従って、DAC クロックレートは 256 x 8 kHz = 2.048 MHz にする必要があります。

この例では dsPIC DSC PLL を設定してシステムクロックを出力できます。システムクロックは、DAC クロックレート (=2.048 MHz) の整数倍です。DAC モジュールへのシステムクロック源は Fvco と呼び、後置 PLL 分周器 (N2) の前にある PLL の出力です。詳細は、**セクション 39. 「オシレータ (パート III)」** (DS70216) の図 39-8 「PLL ブロック図」を参照してください。M の値を 40 とすると、Fvco は以下の式 33-1 で求められます。

式 33-1: Fvco の式

$$F_{vco} = (7.3728 * 106 * 40) / 2 = 147.456 * 106 \text{ MHz}$$

この Fvco 値は、必要な DAC クロック値の整数倍 (2.048 MHz x 72 = 147.456 x 10⁶ MHz) です。SELACLK (ACLKCON<13>) ビットを「0」にセットし、Fvco クロックを 72 分周するように DACFDIV レジスタ値を設定すると、必要な DAC クロックを生成できます。

セクション 33. オーディオ D/A コンバータ (DAC)

表 33-1 に、一般的なオーディオ サンプルレートを得るために必要な DACCLK レートを示します。

表 33-1: クロックレート比

DACCLK レート (MHz) (Fs x 256)	サンプル データレート (kHz) (Fs)
1.8432	7.2
2.048	8.0
2.4576	9.6
2.8224	11.025
4.096	16
5.6448	22.05
6.144	24
8.192	32
11.2896	44.1
12.288	48
22.5792	88.2
24.576	96.0
25.6	100

DACCLK レートは適切な DACFDIV<6:0> コンフィグレーション ビットを使用して補助オシレータを選択する事によって得られます。例えば、25.6 MHz の DACCLK レート (100 ksps の入力レートの場合) は表 33-2 に示すようにして得ることができます。

表 33-2: 補助オシレータ周波数

補助オシレータ周波数	DACFDIV コンフィグレーション ビット
25.6 MHz	DACFDIV = 0b0000000
51.2 MHz	DACFDIV = 0b0000001
76.8 MHz	DACFDIV = 0b0000010

オシレータの選択の詳細は、セクション 39. 「オシレータ (パート III)」 (DS70216) を参照してください。

33

オーディオ
D/A コンバータ
(DAC)

33.5 割り込みとステータス

オーディオ DAC にはチャンネルごとに 1 つずつ、2 つの割り込みがあります。DAC ステータス レジスタ (DACxSTAT) の割り込みコンフィグレーション ビット (左側チャンネルは LITYPE、右側チャンネルは RITYPE) の設定に応じて、DAC 割り込みは「FIFO エンプティ」または「FIFO がフルでない」のいずれかの条件によってトリガされます。「FIFO エンプティ」割り込みをオーディオ DAC で使用すると、スループットを最大限まで引き上げると同時に、CPU に対する割り込みの影響を最小限に抑える事ができます。「FIFO エンプティ」割り込みは、DMA で使用する場合最も簡単で好ましい割り込み方法です。

「FIFO がフルでない」割り込みは、DMA なしのアプリケーションで DAC アンダーランの発生を最小限に抑えるために使用します。この割り込みは DMA でも使用できますが、ソフトウェア サポートの追加が必要です。

DAC 割り込みサービスルーチン (ISR) は、対応する割り込みフラグ (IFS4 レジスタの DAC1LIF と DAC1RIF) をクリアする必要があります。

Note: オーディオ DAC は 256 DAC クロックサイクルごとに FIFO からデータを読み出します。データが読み出された後、FIFO のステータスに応じて DAC 割り込みが発生します。

表 33-3 に示すように、各チャンネルにはその FIFO のステータスを示す 2 つのステータスビットがあります。

表 33-3: 割り込みステータスビット

チャンネル	名前 ⁽¹⁾	説明
左	LFULL (DAC1STAT<9>)	左側チャンネルの FIFO がフル
	LEMPY (DAC1STAT<8>)	左側チャンネルの FIFO がエンプティ
右	RFULL (DAC1STAT<1>)	右側チャンネルの FIFO がフル
	REMPY (DAC1STAT<0>)	右側チャンネルの FIFO がエンプティ

Note 1: これらのビットは FIFO ステータスを確認するためにソフトウェアで読み取る事ができます。

33.6 DMA なしでのオーディオ DAC 動作

例 33-1 に、オーディオ DAC モジュールの代表的な設定を示します。この例では、対応する FIFO がフルでない時は必ず両方のチャンネルの割り込みが発生するように設定しています。DAC 制御 (DACxCON<15>) レジスタの DAC イネーブル (DACEN) ビットをセットすると、両方のチャンネルの DAC 割り込みが発生します。FIFO は最初はエンプティであるため、最初のデータ値は DAC 既定値 (DACxDFLT) レジスタから読み出されます。この例では、既定値は 0x8000 という中間値に設定しています。

例 33-1: DMA なしでの DAC 動作

```
DAC1STATbits.ROEN = 1;          /* Right Channel DAC Output Enabled */
DAC1STATbits.LOEN = 1;          /* Left Channel DAC Output Enabled */

DAC1STATbits.RITYPE = 0;        /* Right Channel Interrupt if FIFO is not Full */
DAC1STATbits.LITYPE = 0;        /* Left Channel Interrupt if FIFO is not Full */

DAC1CONbits.AMPON = 0;          /* Amplifier Disabled During Sleep and Idle Modes */
DAC1CONbits.DACFDIV = 0;        /* Divide Clock by 1 (Assumes Clock is 25.6MHz) */
DAC1CONbits.FORM = 0;           /* Data Format is Unsigned */

DAC1DFLT = 0x8000;              /* Default value set to Midpoint when FORM = 0 */

IFS4bits.DAC1RIF = 0;           /* Clear Right Channel Interrupt Flag */
IFS4bits.DAC1LIF = 0;           /* Clear Left Channel Interrupt Flag */

IEC4bits.DAC1RIE = 1;           /* Right Channel Interrupt Enabled */
IEC4bits.DAC1LIE = 1;           /* Left Channel Interrupt Enabled */

DAC1CONbits.DACEN = 1;          /* DAC1 Module Enabled */

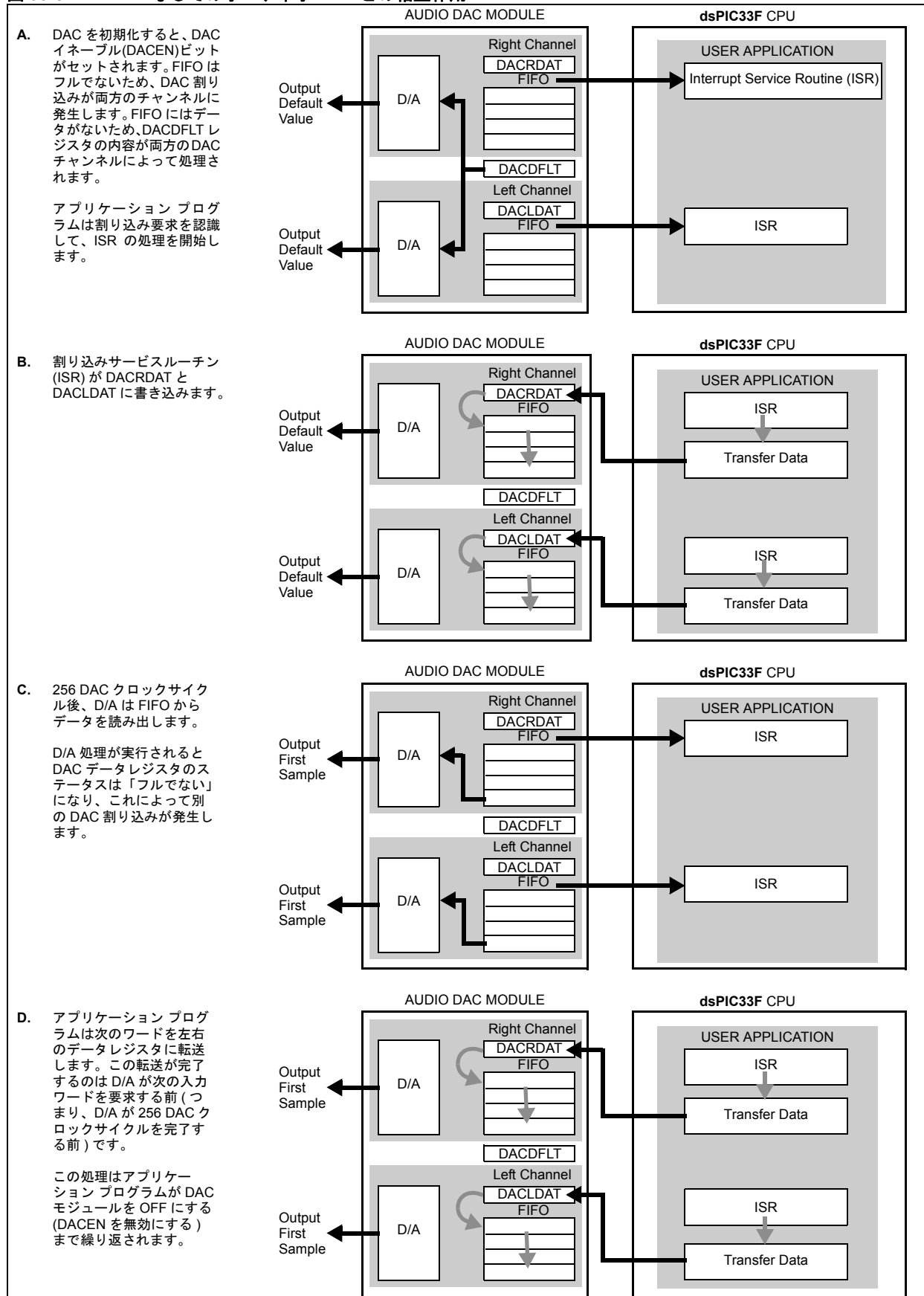
void __attribute__((interrupt, no_auto_psv)) _DAC1RInterrupt(void)
{
    IFS4bits.DAC1RIF = 0;        /* Clear Right Channel Interrupt Flag */
    DAC1RDAT = MyDataR[0];       /* User Code to Write to FIFO Goes Here */
}

void __attribute__((interrupt, no_auto_psv)) _DAC1LInterrupt(void)
{
    IFS4bits.DAC1LIF = 0;        /* Clear Left Channel Interrupt Flag */
    DAC1LDAT = MyDataL[0];       /* User Code to Write to FIFO Goes Here */
}
```

図 33-3 に、オーディオ DAC とアプリケーション プログラムの相互作用で、DAC 割り込みに応答してデータを適時転送する手順を示します。この例では、1 回の割り込みで 1 ワード転送します。アプリケーションによっては、1 回の割り込みで最大 4 ワードの書き込みが可能です。

Note: FIFO がフルである場合または DAC が有効になる前に書き込みが行われた場合、FIFO への書き込みは無視されます。

図 33-3: DMA なしでのオーディオ DAC との相互作用



33.7 DMA ありでのオーディオ DAC 動作

一部の dsPIC33F では、DMA モジュールが CPU 時間を消費せずに CPU からオーディオ DAC へデータを転送できます。特定の dsPIC33F で DMA をサポートしているかどうかについては、各デバイスのデータシートを参照してください。DMA モジュールの詳細は、**セクション 22. 「ダイレクト メモリアクセス (DMA)」** (DS70182) を参照してください。

オーディオ DAC モジュールは DMA を使用する場合、DAC チャンネルごとに 1 つの DMA チャンネルが必要です。DAC チャンネルは使用可能な任意の DMA チャンネルに割り当てる事ができます。DMA がオーディオ DAC から割り込みを受け取ると相互作用が開始します。DAC ステータス レジスタ (DACxSTAT) の割り込みコンフィグレーション ビット (左側チャンネルは LITYPE、右側チャンネルは RITYPE) の設定に応じて、DMA 割り込みは「FIFO エンプティ」または「FIFO がフルでない」のいずれかの条件によってトリガされます (**33.5 「割り込みとステータス」** 参照)。

33.7.1 DMA チャンネルと周辺モジュールとの関連付けの設定

DMA チャンネルは、応答する割り込み要求と、オーディオ DAC が書き込む周辺モジュールの宛先アドレスを認識する必要があります。割り込みは (DMA モジュールの) DMA チャンネル x IRQ 要求 (DMAxREQ) レジスタの割り込み選択 (IRQSEL<6:0>) ビットで識別します。書き込みアドレスは (これも DMA モジュールの) DMA チャンネル x 周辺モジュールアドレス (DMAxPAD) レジスタで識別します。

表 33-4 に、特定の周辺モジュールと任意の DMA チャンネルを関連付けるために書き込む値を示します。

表 33-4: DMA チャンネルと周辺モジュールの関連付け

周辺モジュールと DMA の関連付け	DMAxREQ レジスタ IRQSEL<6:0> ビット	周辺モジュールに書き込む DMAxPAD レジスタの値
DAC1 – 右側のデータ出力	1001110	0x03F6 – DAC1RDAT
DAC1 – 左側のデータ出力	1001111	0x03F8 – DAC1LDAT

33.7.2 DMA コードの設定

例 33-2 に、DMA 動作に対応する代表的 DAC 設定のコードを示します。この例では、FIFO がエンプティの時に割り込みが発生するように設定しています。DAC イネーブル (DACEN) ビットをセットすると、FIFO は最初エンプティであるため割り込みが発生します。FIFO はエンプティであるため、最初のデータ値は既定値のレジスタから読み出されます。この場合、既定値は 0 です。

図 33-4 に、オーディオ DAC と DMA モジュールの相互作用で、DAC 割り込みに応答してデータを適時転送する手順を示します。この例では、1 回の割り込みで 1 ワード転送します。

例 33-2: DMA ありでの DAC 動作

```
/* DMA Code */
unsigned int RightBufferA[32]__attribute__((space(dma)));
unsigned int RightBufferB[32]__attribute__((space(dma)));
unsigned int LeftBufferA[32]__attribute__((space(dma)));
unsigned int LeftBufferB[32]__attribute__((space(dma)));

/* DMA Channel 0 set to DAC1RDAT */
DMA0CONbits.AMODE = 0; /* Register Indirect with Post Increment */
DMA0CONbits.MODE = 2; /* Continuous Mode with Ping-Pong Enabled */
DMA0CONbits.DIR = 1; /* Ram-to-Peripheral Data Transfer */

DMA0PAD = (volatile unsigned int)&DAC1RDAT; /* Point DMA to DAC1RDAT */

DMA0CNT = 31; /* 32 DMA Request */
DMA0REQ = 78; /* Select DAC1RDAT as DMA Request Source */

DMA0STA = __builtin_dmaoffset(RightBufferA);
DMA0STB = __builtin_dmaoffset(RightBufferB);

IFS0bits.DMA0IF = 0; /* Clear DMA Interrupt Flag */
IEC0bits.DMA0IE = 1; /* Set DMA Interrupt Enable Bit */

DMA0CONbits.CHEN = 1; /* Enable DMA Channel 0 */

/* DMA Channel 1 set to DAC1LDAT */
DMA1CONbits.AMODE = 0; /* Register Indirect with Post Increment */
DMA1CONbits.MODE = 2; /* Continuous Mode with Ping-Pong Enabled */
DMA1CONbits.DIR = 1; /* Ram-to-Peripheral Data Transfer */

DMA1PAD = (volatile unsigned int)&DAC1LDAT; /* Point DMA to DAC1LDAT */

DMA1CNT = 31; /* 32 DMA Request */
DMA1REQ = 79; /* Select DAC1LDAT as DMA Request Source */

DMA1STA = __builtin_dmaoffset(LeftBufferA);
DMA1STB = __builtin_dmaoffset(LeftBufferB);

IFS0bits.DMA1IF = 0; /* Clear DMA Interrupt Flag */
IEC0bits.DMA1IE = 1; /* Set DMA Interrupt Enable Bit */

DMA1CONbits.CHEN = 1; /* Enable DMA Channel 1 */

/* DAC1 Code */
DAC1STATbits.ROEN = 1; /* Right Channel DAC Output Enabled */
DAC1STATbits.LOEN = 1; /* Left Channel DAC Output Enabled */

DAC1STATbits.RITYPE = 1; /* Right Channel Interrupt if FIFO is Empty */
DAC1STATbits.LITYPE = 1; /* Left Channel Interrupt if FIFO is Empty */

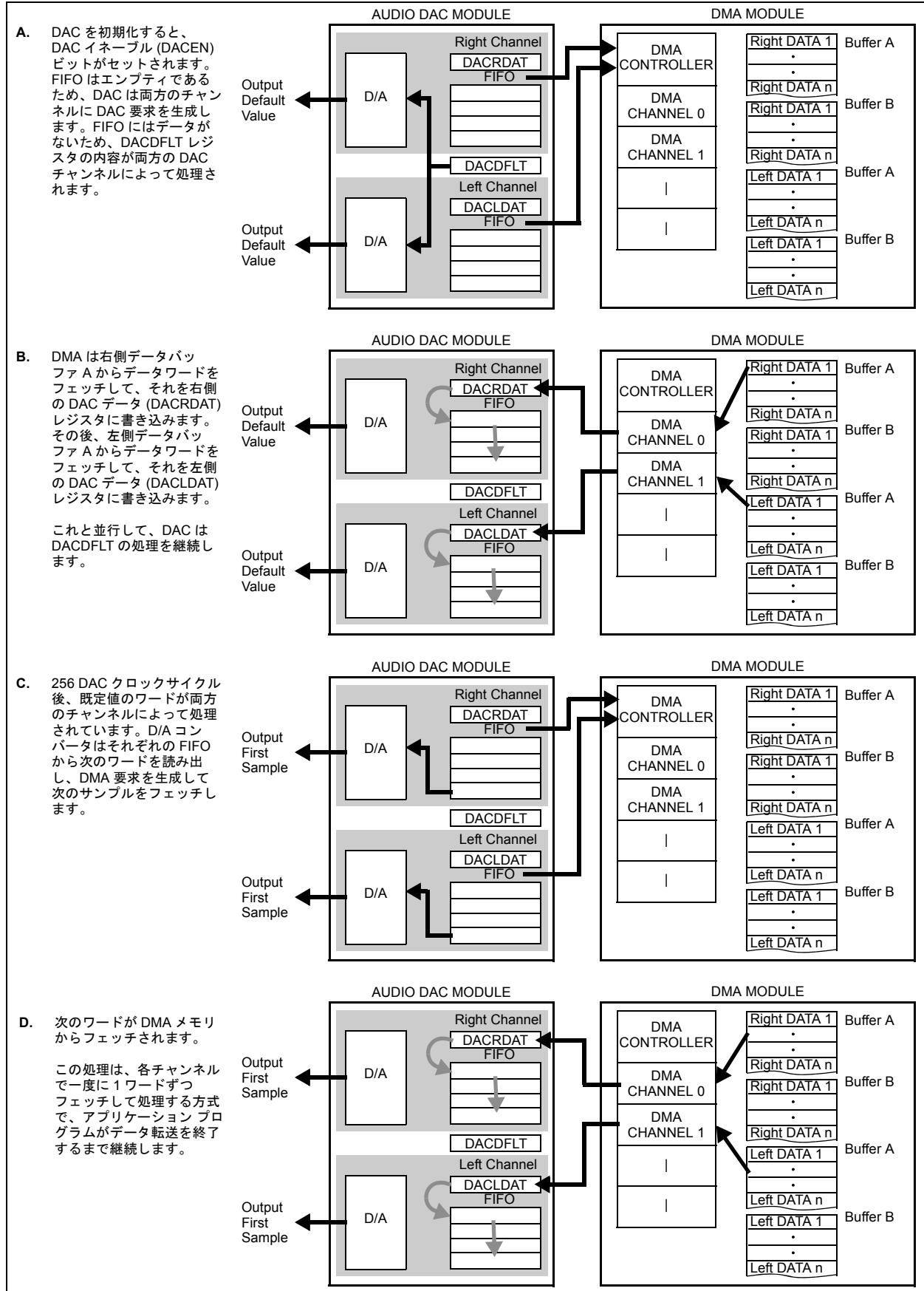
DAC1CONbits.AMPON = 0; /* Amplifier is Disabled During Sleep/Idle Modes */
DAC1CONbits.DACFDIV = 0; /* Divide Clock by 1 (Assumes Clock is 25.6MHz) */
DAC1CONbits.FORM = 0; /* Data Format is Unsigned */
DAC1CONbits.DACEN = 1; /* DAC1 Module Enabled */
/* Rest of User Code Goes Here */

void__attribute__((interrupt, no_auto_psv))_DMA0Interrupt(void)
{
    IFS0bits.DMA0IF = 0; /* Clear DMA Channel 0 Interrupt Flag */
    /* User Code to update Right Buffer in DMA */
}

void__attribute__((interrupt, no_auto_psv))_DMA1Interrupt(void)
{
    IFS0bits.DMA1IF = 0; /* Clear DMA Channel 1 Interrupt Flag */
    /* User Code to update Left Buffer in DMA */
}
```

セクション 33. オーディオ D/A コンバータ (DAC)

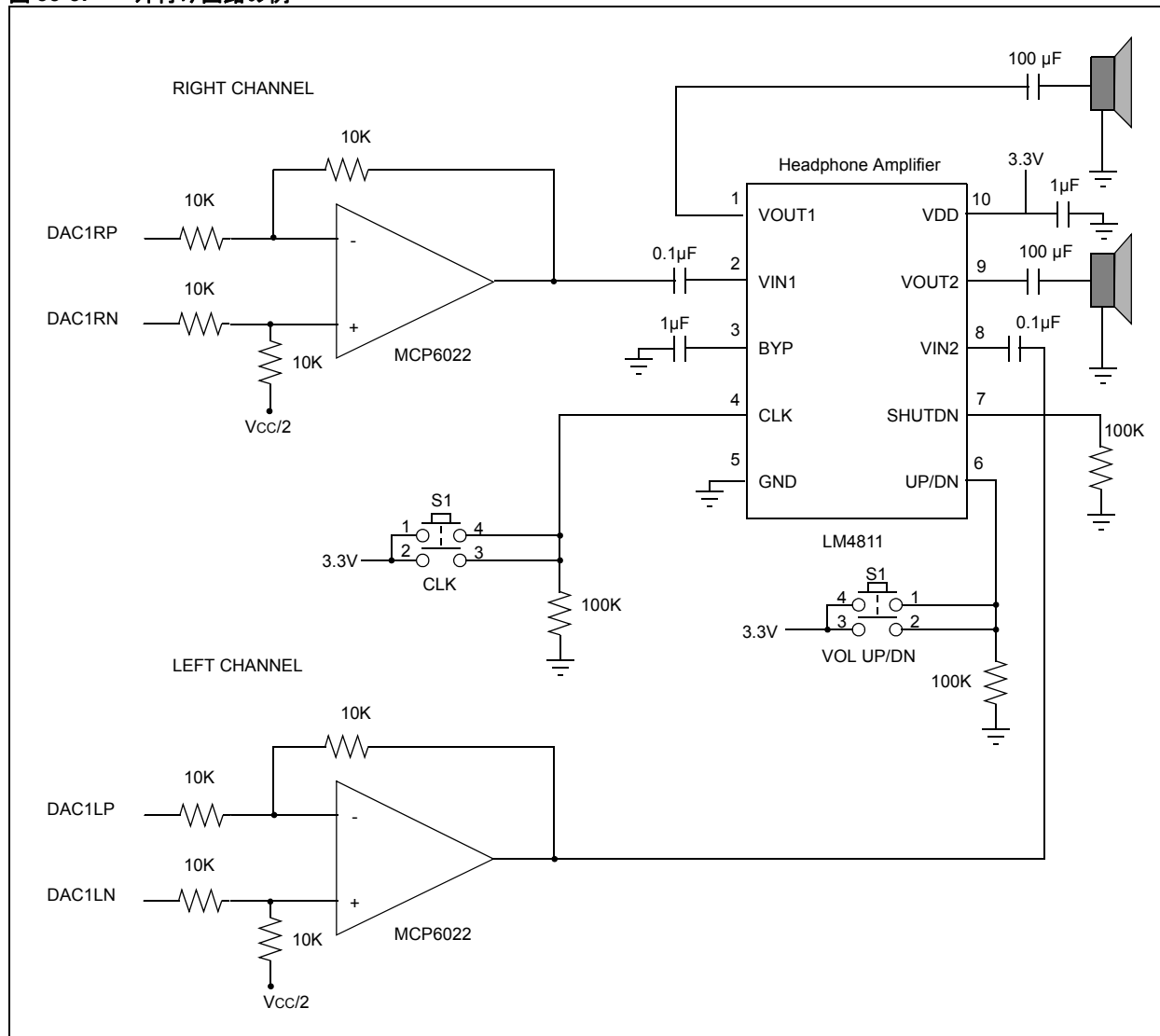
図 33-4: DMA ありでのオーディオ DAC の相互作用



33.8 外付け回路の例

図 33-5 に、オーディオ DAC モジュールの 1 つのチャンネルを使用してスピーカを接続する代表的構成を示します。この例ではオーディオ DAC の差動出力を使用し、差動増幅構成のアンプを使用してシングルエンド出力を生成します。対応する出力は正入力の 2 倍です。

図 33-5: 外付け回路の例



33.9 省電力モード時の動作

dsPIC33F デバイスファミリには、1つの通常 (フルパワー) モードと、PWSAV 命令によって呼び出される3つの省電力モードの4つの電力モードがあります。選択したモードによっては、省電力モードへの移行がモジュールの動作にも影響する場合があります。

DAC 制御 (DACxCON<12>) レジスタのアナログ出力アンプイネーブル (AMPON) ビットをセットすると、スリープモード /Stop-in-Idle モード中でもアナログ回路とアナログ出力アンプの電源を ON にします。この設定により、スリープモード /Stop-in-Idle モード中でもアナログ出力電圧を既知の状態に維持できます。AMPON ビットをクリアしてからプロセッサをスリープモードまたは Stop-in-Idle モードに移行させると、アナログ回路の電源は OFF になってリセット状態に保持され、消費電流を低減できます。出力アンプの電源が OFF になると、アナログ出力はハイ インピーダンス状態になります。

スリープモードでは、AMPON ビットをクリアすると、FIFO を含む DAC の内部状態はリセットされ (SFR を除く)、クロックは停止して、出力アナログアンプと関連回路の電源は OFF になります。AMPON ビットをセットすると、DAC の内部状態と SFR は維持され、クロックは停止して、出力のアナログアンプと関連回路の電源は ON のままになります。

アイドルモードに移行した時に DAC 制御 (DACxCON<13>) レジスタの Stop-in-Idle モード (DACSIDL) ビットがセットされている場合で、AMPON ビットがセットされていない場合、DAC の内部状態はリセットされ (SFR を除く)、クロックは停止して、出力のアナログアンプと関連回路の電源は OFF になります。AMPON ビットをセットすると、DAC の内部状態と SFR は維持され、クロックは停止して、出力のアナログアンプと関連回路の電源は ON のままになります。アイドルモードに移行した時に DACSIDL がセットされていない場合、DAC は通常動作を継続します。

デバイスがスリープモードまたは Stop-in-Idle モードに移行すると、AMPON ビットの状態には関係なく、DACEN ビット (DACxCON<15>) はクリアされます。AMPON ビットをセットしてからデバイスがスリープモードまたは Stop-in-Idle モードに移行した場合、新しいデータはスリープモードまたは Stop-in-Idle モードの終了時に FIFO に格納する事ができます。DAC を再度有効にすると、DAC は無効になった所から動作を再開します。

Note: AMPON = 1 かつ DACEN = 0 の場合、FIFO を読み出すと FIFO はエンプティになります。

33.10 レジスタマップ

表 33-5 に、オーディオ DAC レジスタの要約を示します。

表 33-5: オーディオ DAC のレジスタマップ

レジスタ名	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	全 リセット
DACxCON	DACEN	—	DACSIDL	AMPON	—	—	—	FORM	—	DACFDIV<6:0>							0000
DACxSTAT	LOEN	—	LMVOEN	—	—	LITYPE	LFULL	LEMPY	ROEN	—	RMVOEN	—	—	RITYPE	RFULL	REMPY	0000
DACxDFLT	DACDFLT<15:0>																0000
DACxRDAT	DACRDAT<15:0>																0000
DACxLDAT	DACLDAT<15:0>																0000

凡例: x = リセット時の値は不定、— = 未実装、「0」として読み出し、リセット時の値は 16 進数で表示

33.11 関連アプリケーション ノート

本セクションに関連するアプリケーション ノートの一覧を以下に示します。一部のアプリケーション ノートは dsPIC33F ファミリー向けではありません。ただし概念は共通しており、変更が必要であったり制限事項が存在するものの利用が可能です。オーディオ D/A コンバータ (DAC) モジュールに関連する現在のアプリケーション ノートは以下の通りです。

タイトル

アプリケーション ノート番号

現在、関連するアプリケーション ノートはありません。

N/A

Note: dsPIC33F デバイスファミリー向けのその他のアプリケーション ノートとサンプルコードは、マイクロチップ社のウェブサイト (www.microchip.com) をご覧ください。

33.12 改訂履歴

リビジョン A (2007 年 10 月)

初版発行

リビジョン B (2009 年 9 月)

このリビジョンでの変更内容は以下の通りです。

- 式：
 - FVCO の式を 33.4.2 「クロック」に追加 (式 33-1 参照)
- 図：
 - 図 33-5 の 2 つの差動アンプの + 符号、- 符号、入力信号名を更新
- Notes:
 - 33.2 「主な特長」にオーディオ DAC モジュールに関する注釈を追加
- セクション：
 - 33.4.2 「クロック」に DAC のサンプルコードを追加
 - 33.11 「関連アプリケーション ノート」を追加
 - 33.1 「はじめに」で以下の記述を削除：
正負の出力は約 1.65 V の中間電圧を中心に約 ± 1 V の電圧振幅を持つ差動出力です。
この信号には 1 k Ω 負荷の駆動能力があります。
 - 33.2 「主な特長」で「S/N 比: 90 dB」という記述を削除。
 - 33.4 「モジュールの動作」で以下の記述を削除：
再構成フィルタからの差動出力はアナログアンプによって増幅され、必要な 2 Vp-p の電圧振幅が 1 k Ω 負荷に加えられます。
- 表現と体裁の変更等、本書全体の細部を修正

ISBN: 978-1-60932-998-3