

LM12454,LM12458,LM12H458,LM12L458

*AN-906 Interfacing the LM12454/8 Data Acquisition System Chips to
Microprocessors and Microcontrollers*



Literature Number: JAJA265

データ・アキュイジション・システム・チップ LM12454/8 とマイクロプロセッサ/マイクロコントローラとのインタフェース

National Semiconductor
Application Note 906
Farid Saleh
August 1993



目次

1.0 はじめに

2.0 概要

- 2.1 DAS プログラミング・モデル
- 2.2 プログラミング手順
- 2.3 一般的なプログラム・フローチャートとその他の手法
- 2.4 DAS/ プロセッサ・インタフェース

3.0 DASとHPC™マイクロコントローラとのインタフェース

- 3.1 全アドレス・デコーディング
- 3.2 最小アドレス・デコーディング
- 3.3 タイミング分析
 - 3.3.1 全アドレス・デコーディング回路
 - 3.3.2 最小アドレス・デコーディング回路

4.0 システム例: 半導体炉

- 4.1 システムの要件と前提
- 4.2 DASの結線とレジスタのプログラミング
- 4.3 マイクロコントローラのプログラミング
 - 4.3.1 半導体炉用 HPC アセンブリ・ルーチンの例

1.0 はじめに

データ・アキュイジション・システム(DAS)チップ LM12454/8 ファミリーは、全差動、セルフキャリブレーションの 12 ビット (+サイン・ビット) A/D コンバータです。差動入力基準電圧、4 または 8 入力のアナログ・マルチプレクサ、そして、きわめてフレキシブルにプログラム可能なロジックを備えています。このロジックには、特定の機能を実現のための以下のユニット等が集積されています:

- アキュイジション時間、入力選択、8 ビットまたは 12 ビット変換モードなどをホストによってプログラム可能なスタンバイ・アロン動作の命令 RAM

- "ウォッチドッグ" モードにおいて、入力と上限と下限リミットの比較を行うためのリミット・レジスタ
- 変換結果を格納するための 32 ワード FIFO レジスタ
- 8 種類の条件のいずれかが発生すると割り込みを生成する割り込み制御ロジック
- 16 ビットのタイマ・レジスタ
- 信号の収集を外部信号に同期させる回路
- データ・アクセスが 8 ビットまたは 16 ビットのマイクロプロセッサとの並列インタフェース

多くの機能と設計の自由度があるために、LM12454/8 ファミリーを使用する事は、一見して非常に厄介なものに思われるかもしれませんが。しかし、デバイスの機能ユニットとその動作概念に関して基本的な理解が得られれば、難しいことではありません。このアプリケーション・ノートでは、LM12454/8 ファミリーの利用が非常に容易であることを示し、デバイスのインタフェースとプログラミング手順を具体的に解説します。

LM12454/8 ファミリーは 6 製品から成ります。各製品とその差異を Table 1 に示しました。本アプリケーション・ノートの以下の部分では、便宜上、LM12454/8 ファミリー製品の総称として DAS という表現を使用します。同様に、図には、ファミリーの 8 入力バージョンのみを示しました。なお、本アプリケーション・ノートは、デバイスのデータシートと合わせて利用してください。読者はある程度このデバイスについて、知っているものとして、書かれています。本ノートは、DAS の簡単な概要とアプリケーションに必要な関連情報は含めています。

2.0 概要

2.1 DAS プログラミング・モデル

Fig. 1 に DAS の機能ブロック・ダイアグラムとユーザ・プログラミング・モデルを示します。(この図は DAS 内部の実際の構成ブロックを示すものではありません) DAS モデルは以下のブロックから成ります:

- デバイスのフロント・エンドに位置する差動出力型フレキシブル・アナログ・マルチプレクサ

TABLE 1: Members of the LM12454/8 Family

Device Number	Clock Frequency (Max, MHz)	Operating Supply Voltage (V)	Number of MUX Inputs	Internal Reference	Low Voltage Flag
LM12454	5	5.0 ± 10%	4	Yes	Yes
LM12458	5	5.0 ± 10%	8	Yes	Yes
LM12H454	8	5.0 ± 10%	4	Yes	Yes
LM12H458	8	5.0 ± 10%	8	Yes	Yes
LM12L454	6	3.3 ± 10%	4	No	No
LM12L458	6	3.3 ± 10%	8	No	No

HPC™ は、ナショナル セミコンダクター社の商標です。

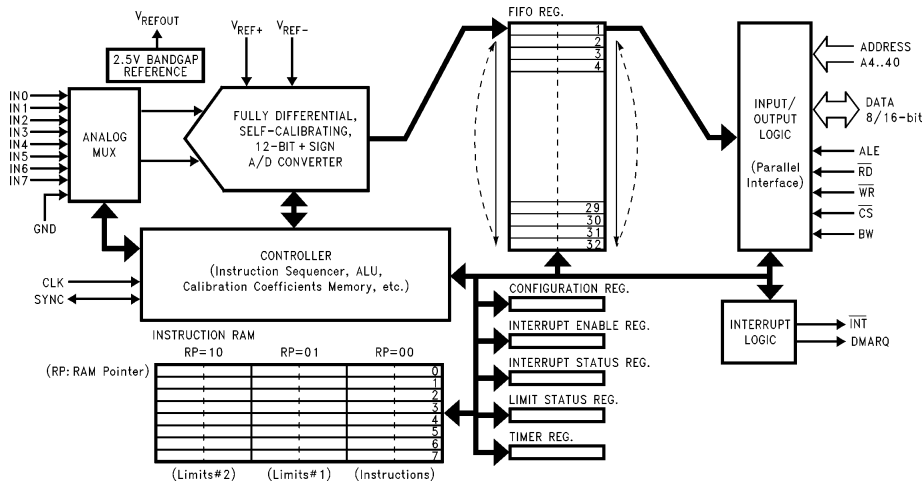


FIGURE 1. DAS Functional Block Diagram, Programming Model

TL/H/11908-1

- 全差動・セルフキャリブレーション機能付12ビット+サインA/Dコンバータ
- 出力データ・バッファとして使用される32ワードFIFOレジスタ
- 選択した入力チャネルでの変換、比較の反復実行をプログラム可能な命令RAM
- DAS動作の全制御とコンフィギュレーション、および内部動作状態の表示のための一連のレジスタ
- 特定の条件で割り込みサービスをプロセッサに要求する割り込み生成ロジック
- DASとプロセッサ間における入出力動作のための並列インタフェース・ロジック。図に示したレジスタはすべて、入出力ブロックを介して外部に読出し可能。同様に、ほとんどのレジスタが書込み可能。
- DAS内部のブロック間動作を制御し、一連の変換、比較、キャリブレーション処理を実行するコントローラ・ユニット。

DASには3種類の動作モードがあります。12ビット+サインの変換、8ビット+サインの変換、そして、8ビット+サインの比較“ウォッチドッグ”モードとも呼ばれます。ウォッチドッグ・モードでは、変換は行なわれず、DASは、入力をサンプリングしてそれを命令RAMに格納されている上下限値と比較します。入力電圧が上下の(ユーザによって設定された)リミット値を越えた場合、割り込み信号を生成して障害状態を示すことができます。

命令RAMは、それぞれ48(3×16)ビット長の8ワードに分かれています。各ワードは、さらに、それぞれ16ビットの3つの部分に分かれています。各ワードは独自のアドレスを持ち、命令の各部分は、構成レジスタ内の2ビットのRAMポインタ(RP)で選択されます。Fig. 1に示したように、この命令RAMの各部分は、命令部分、リミット#1部分、リミット#2部分のラベルが付いています。命令部分は、選択する入力チャネル、各命令の動作モード、アキュジション時間といった動作情報が保持されます。他の二つの部分は、ウォッチドッグ・モードで使用される部分で、ユーザ指定のリミット値が格納されます。一つのウォッチドッグ命令は、二つのリミット値を使用します。(通常は上限値と下限値ですが、両方を上限値または下限値にプログラムする

ことも可能)。DASは、命令0から実行を開始してユーザ指定の命令まで実行を続けた後、命令0に“ループ・バック”することができます。この場合、命令RAM内の8個の命令すべてがループ内で実行される必要はありません。この反復実行は、外部から停止されるまで続けられます。命令RAMへの外部からのアクセスは、この命令シーケンスが停止された後に行なわれなければなりません。

FIFOレジスタは、変換結果の格納に使用されます。このレジスタは“リード・オンリ”で、すべてのロケーションは、それぞれ単一のアドレスでアクセスされます。変換が実行されるたびに結果がFIFOに格納され、FIFOの内部書込みポインタが次のロケーションを指示します。ロケーション32への書込みが終わると、ポインタはロケーション1を指示します。この順番は、FIFOを読出す場合も同様です。なお、FIFOの書込みポインタと読出しポインタはそれぞれ独立しており、書込みや読出しによって他方のポインタが影響を受けることはありません。

構成レジスタは、DASのメイン“コントロール・パネル”です。この構成レジスタの然るべきビットに1や0を書込むことにより、シーケンスのスタートやストップ、ポインタやフラグのリセット、低消費電力スタンバイ・モードへの移行、オフセットや直線性のキャリブレーション、RAMセクションの選択といった各種の動作の実行をDASに命じます。

割り込みイネーブル・レジスタには、イネーブルする最大8個の割り込み生成源を指定します。また、この割り込みイネーブル・レジスタは、二つのユーザ・プログラマブル値も保持しますが、一つは、データ・レディ割り込みを生成する前にFIFOレジスタに格納すべき変換結果の数であり、もう一つは、シーケンスがその命令に達した時に割り込みを生成する命令の番号です。

割り込みステータスおよびリミット・ステータス・レジスタは、“リード・オンリ”で、どの条件によって割り込みが生成されたか、また、どのリミット値が逸脱されたかを示すベクトルとして使用されます。なお、これらのビットは、対応する割り込み条件が発生すると、その条件が外部割り込み生成条件としてイネーブルされているかどうかにかかわらず、常にセットされます。

タイマ・レジスタをプログラムして、各命令の実行前に遅延を挿入できます。命令レジスタのタイマ・ビットによって、命令実行前の遅延がイネーブルまたはディセーブルされます。

DAS のアクセス可能な全てのレジスタとそのビット配置が付録 A に示されています。ビット配置はデータシートに詳説され、本アプリケーション・ノートでも再度言及します。プログラミング・ツールとして利用できる空白のレジスタ・モデルを同じページに用意しました。設計者は、システム要件にしたがって“1”や“0”をこのレジスタ・モデルの空欄に埋めることによってプログラミングの容易化を図ることができます。これらのシートは、設計のドキュメンテーションに利用することもできます。

2.2 プログラミング手順

DAS は、プロセッサによって制御される様に設計されています。しかし、DAS の機能により、プロセッサの負担は大幅に軽減され、ソフトウェアのオーバーヘッドが低減します。プロセッサはまず、一連の動作命令をDASのRAMとレジスタにダウンロードし、その後スタート・コマンドをDAS に与えます。DAS は、命令にしたがって変換または比較を連続実行し、変換結果をFIFOに格納します。これ以降のプロセッサの動作は2通りが可能です。一つは、DAS が生成する割込みを待つ場合です。DAS は、あらかじめ定められた数の変換結果がFIFOに格納されたとき、あるいは、他の割込み条件が発生したときに、プロセッサに対して割込みを生成することができます。プロセッサは、割込みの性質に応じて、FIFOの読み出し、もしくは、対策処理を実行します。もう一つは、割込みとは無関係に動作する場合です。プロセッサは、いつでもデータを读出したり、新しいコマンドをDASへ送ることができます。

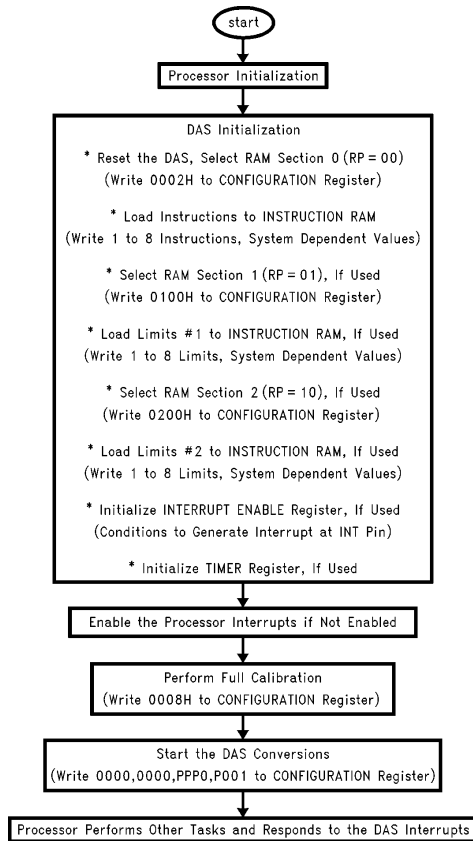
DAS の設計の自由度とアプリケーションの多様性から、すべてのプログラミング手順をここに示すことは実際的ではありませんが、以下に示した代表的な手順によって、DAS起動ルーチンの基本的な概念を理解することができます。

- RESET ビットをセットすることによってDAS をリセットした後、構成・レジスタを介してRAM セクション“00”を選択する。
- (1 ~ 8 番の)命令を命令RAMにロードする。
- (必要ならば)構成レジスタを介してRAMセクション“01”を選択する。
- (必要ならば)1 ~ 8 番の値をリミット#1にロードする。
- (必要ならば)構成レジスタを介してRAMセクション“10”を選択する。
- (必要ならば)1 ~ 8 番の値をリミット#2にロードする。
- (必要ならば)INT ピンから出す割込みの生成条件を選択することにより割込みイネーブル・レジスタを初期化する。
- (必要ならば)タイマ・レジスタに必要な遅延をプログラムする。
- 構成レジスタのSTART ビットをセットすることによって、シーケンサの動作を起動する。同時に、必要に応じて構成レジスタ内の他のビットをセットする。

DASが動作を開始した後のプロセッサの動作は、2通りが可能です。DAS の割込みを待つことも、割込みとは無関係にデータをDASに要求することもできます。

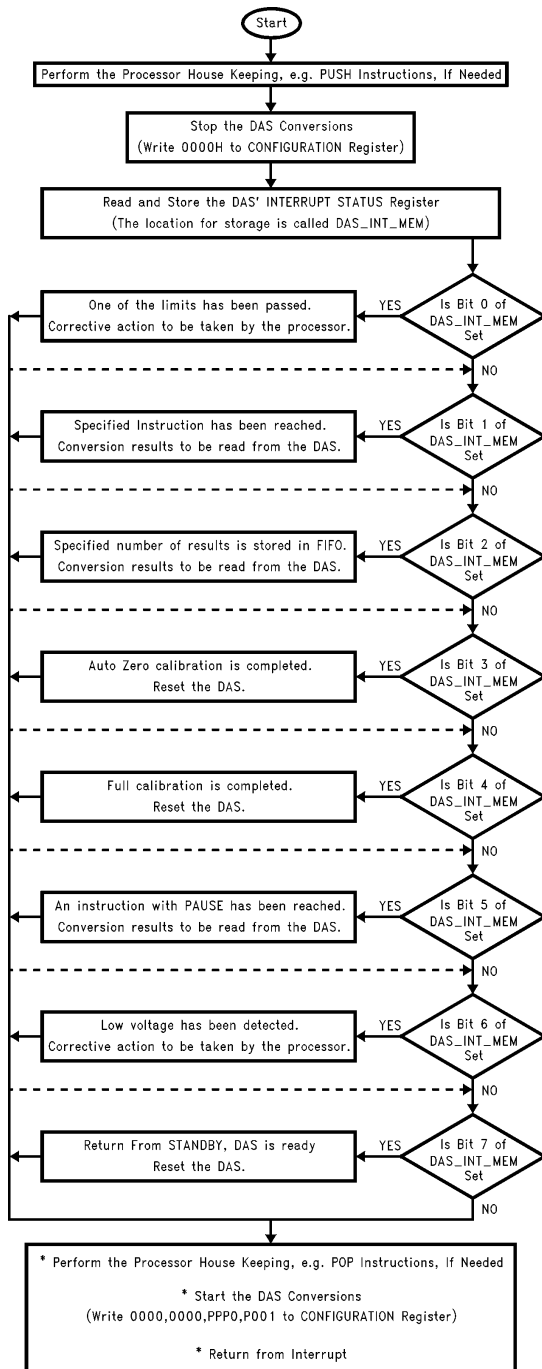
2.3 一般的なプログラム・フローチャートとその他の手法

Fig. 2に代表的なDASプログラム・フローチャートを示します。Fig. 2aは、DAS の初期化と変換の起動です。Fig. 2bは、DAS 割込みサービス・ルーチンの一般形式です。DAS は割込みラインを介してプロセッサとやり取りすることが想定されています。ホスト・プロセッサは他のタスク処理に忙しいのが普通であるため、DASとプロセッサのやり取りは割込みラインを介して行い、DAS への対応は割込みサービス・ルーチンによって行われます。



TL/H/11908-2

FIGURE 2a. A Typical Program Flowchart for the DAS Initialization and Start of Conversions



TL/H/11908-3

FIGURE 2b. A Typical Program Flowchart for the DAS Interrupt Service Routine

フローチャートの最初は、プロセッサの初期化ステップです。プロセッサによっては、DAS との相互作用開始に際して初期化が必要で、このステップも、このフローチャートは含んでいます。

DAS の初期化ステップは、DAS レジスタへの一連の書き込み動作です。これは、2.2 節で説明した通りです。

DAS レジスタを設定した後は一般に、フル・キャリブレーション・サイクルを実行させます。これは、12 ビットの精度を確保するために必要です。ユーザは、フル・キャリブレーション・サイクルを電源投入時に実行するか、一定周期で定期的に行うか、あるいは、一定の温度変化があったときなどの条件により実行するかを選ぶ。キャリブレーションは、そのための制御コードを構成レジスタに書き込むと実行されます。フル・キャリブレーション・サイクルは、5MHz クロックで約 1ms (989 μ s)、8MHz クロックでは約 0.6ms (618 μ s) を要します。キャリブレーション・サイクル起動後、遅延を挿入するか、割込みによってキャリブレーション終了を検出できます。あるいは、割込みステータス・レジスタの対応するフラグ・ビットを読み出すことも可能です。フローチャートでは、キャリブレーション終了の検出は、割込みサービス・ルーチンで処理しています。フル・キャリブレーション・サイクルは、DAS 内部のいくつかのフラグとポインタを変更するため、その後の最初の命令の実行に影響を及ぼします。間違えた命令実行を回避するには、キャリブレーション後に DAS をリセットします。図示した割込みサービス・ルーチンのフローチャートでもリセットが行われています。

キャリブレーション後、DAS は変換が開始可能となります。構成レジスタへの書き込みによって START ビットを "1" にセットすると、変換が開始されます。構成レジスタに書き込むべきデータは、2進形式でフローチャートに示されています。"P" (プログラム) と表記されているビットは、変換中の動作モードを規定する制御ビットです。他のビットはすべて図示された値をプログラムすべきです。ホスト・プロセッサは、DAS を起動した後、前述したようにデータ処理や他の制御タスクを実行することができ、必要に応じて DAS の割込みに対応します。

割込みサービス・ルーチン (Fig. 2b) 起動時、ゼロを構成レジスタの START ビットへ書き込んで、変換を停止させます。ただし、精度やタイミングのために必要な場合を除き、必ずしも変換を停止させる必要はありません。一般に、変換中に DAS の読出しや書き込みを行うと、変換結果に余分なノイズが乗って精度が低下します。精度低下の度合いは、システム設計の多様な要因に依存し、数量化は困難です。電源とグ

ランドの経路、電源のバイパス処理、バス上のロジックの遷移速度、使用しているロジック・ファミリ、DAS が駆動するデータ・バス上の負荷 (抵抗性、容量性) がいずれも、変換ノイズに影響を及ぼします。ただし、大抵のシステムにおいては、変換中の読出しによって精度上に大きな支障を生じた例はほとんどありません。

変換中の読出しについては、注意すべきタイミング上の問題が 2 点あります。

DAS の読出しあるいは書き込み中は、CS を LOW の間、DAS の内部クロックがストップします。これは、外部バスと内部バスの動作を同期させて、内部のバス衝突を防止するためです。書き込みと読出しが内部バスの動作に同期していないことに注意してください。内部クロック・サイクルの停止は、各命令に要するアキュジションと変換の時間を増加させることになります。この増加時間は一定でなく、プロセッサと DAS が非同期で動作するため、予測することは困難です。したがって、固定した信号サンプリング周期がシステム性能に大きな影響を及ぼすような場合は、変換中の読出しは行うべきではありません。

もう一つタイミングに関して注意すべき点は、変換速度と FIFO からの読出し速度の関係です。後続の変換結果で古いデータが消されないよう、FIFO の読出しは十分早く行う必要があります。

フローチャートに戻ります。割込みサービス・ルーチンの主たる仕事は、DAS の割込みステータス・レジスタを読み出して、そのビットを調べ、割込み要因を特定することです。このフローチャートの割込みサービス・ルーチンは、DAS の割込みビットを残らずチェックします。ただし、実際のシステムでは 2, 3 の割込みだけを利用することが多く、その場合は、余分なビット検査はルーチンから省くべきです。また、ビット検査は、システムにおける割込みの優先レベルの順位で行うべきです。各々の割込みに対してどのような処置を行うかは、システムによって大きく異なるため、フローチャートに詳述することは避けました。割込みサービスは最初に変換を停止させますが、変換の再開は、割込みサービス・ルーチン内で行う方法とシステム・ルーチンに復帰した後で行う方法の二通りが可能です。

2.4 DAS/ プロセッサ・インタフェース

プロセッサと DAS 間のインタフェースは、メモリや I/O のインタフェースに似ています。可能な DAS/ マイクロコントローラ・インタフェースの例を Fig. 3, 4, 5 に示しました。

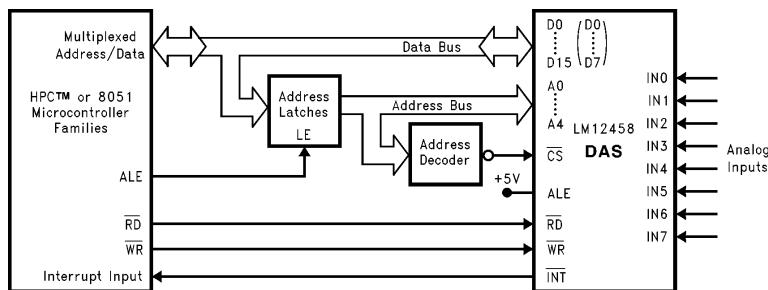


FIGURE 3. LM12458 to HPC or 8051 Microcontroller Interface

TL/H/11908-4

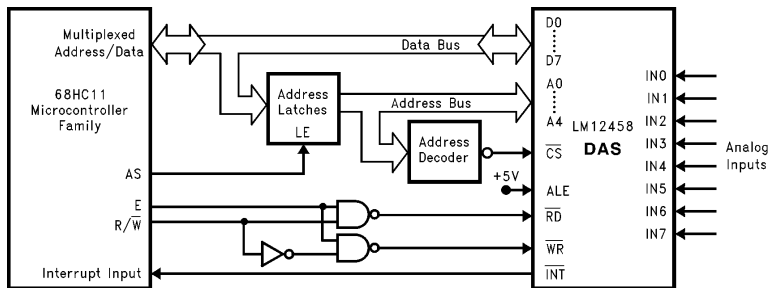


FIGURE 4. LM12458 to 68HC11 Microcontroller Interface

TL/H/11908-5

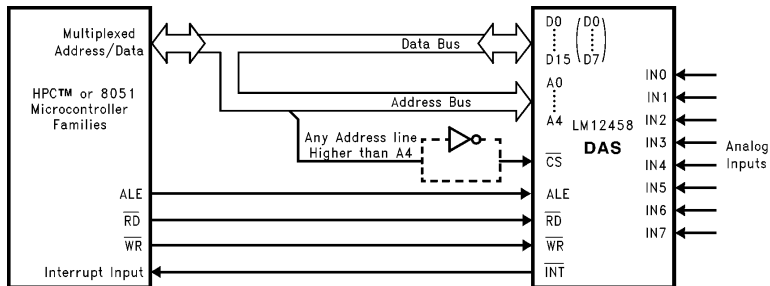


FIGURE 5. LM12458 to HPC or 8051 Microcontroller Interface (Minimum System)

TL/H/11908-6

プロセッサ側から見ると、DASは特定のアドレスを持ったI/Oレジスタの1グループとみなせます。Fig. 6にDASの各レジスタのアドレス、ならびにインタフェース・バス、制御信号との関係を示しました。プロセッサとの並列インタフェースを想定し、DASは標準的なアドレス/データ/制御バス・アーキテクチャを採用しています。DASは、アドレス/データの多重化バス、非多重化バスのいずれのアーキテクチャにも対応可能です。外部のアドレス・ラッチを必要としないシステムでは、ALE入力と内部ラッチにより、DASはアドレス/データ多重化バスに直接インタフェースすることができます。DASのデータ・アクセ

スは、8ビット単位と16ビット単位で行うことができます。BW(バス幅)入力ピンによって、8ビットまたは16ビットのアクセス・モードを選択します。8ビット・アクセス・モードでは、各16ビットI/Oレジスタが2サイクルでアクセスされます。アドレス・ラインA0によって、16ビット・レジスタの上位または下位部分が選択されます。16ビット・アクセス・モードでは、アドレス・ラインA0はdon't careです。Fig. 6aと6bに示されているように、プロセッサから見るとDASは、14個の16ビットI/Oロケーションまたは28個の8ビットI/Oロケーションとみなされます。

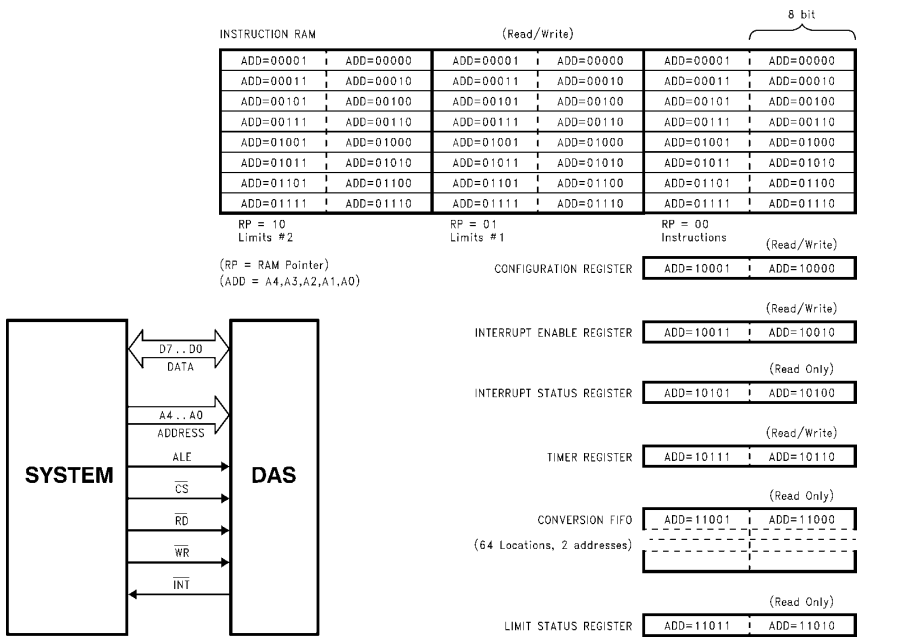


FIGURE 6a. DAS Registers, Address Assignments, Interface Buses and Control Signals for 8-Bit Bus Width

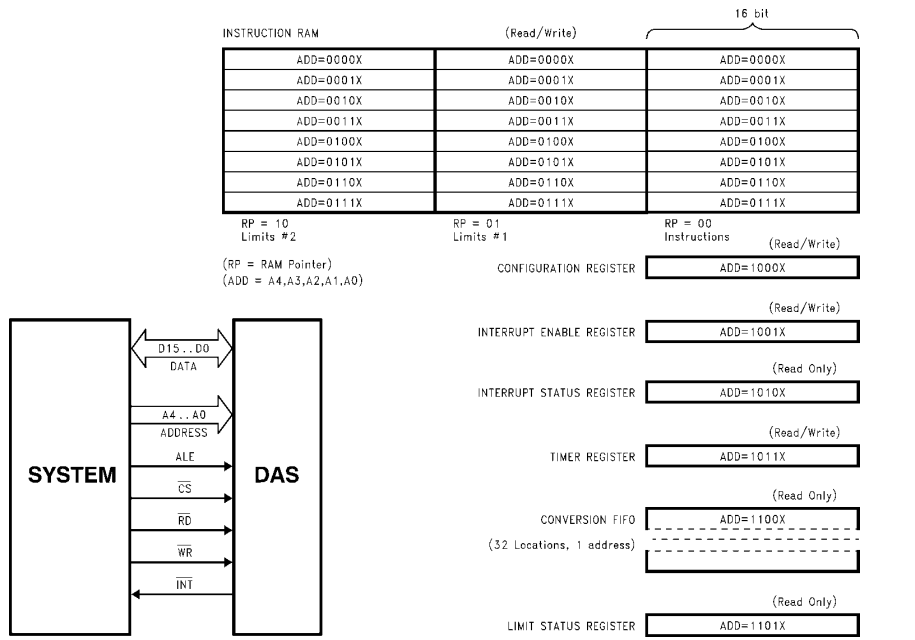


FIGURE 6b. DAS Registers, Address Assignments, Interface Buses and Control Signals for 16-Bit Bus Width

アドレス/データ/制御信号をDASに供給するインタフェースは、以下の要件を満足する必要があります:

- 所定のアドレス範囲内でDASのチップ・セレクトを生成するアドレス・デコーダがあること。
- ALE, CS, RD, WR, アドレス・バス、データ・バス間のタイミング関係がDASのタイミング条件を満足していること。(タイミング条件についてはデータシートを参照してください。)
- DASを割込み生成I/Oとして動作させる場合は、DASとシステム間に適当なサービス要求リンクを確保すること。これは、単にDASのINT出力をプロセッサの割込み入力へ接続すれば実現できる場合もあれば、多数のI/Oデバイスを有するシステムでは、割込み調停ロジック(割込みコントローラ)などの高度な回路が必要になる場合もあります。

Fig. 3は、ナショナルセミコンダクター社の16ビット・マイクロコントローラHPCファミリや8ビット・マイクロコントローラ8051ファミリを使用した場合の一般的なインタフェース回路です。Fig. 4は、マイクロコントローラ68H11ファミリもしくはそれと同等の制御バス・アーキテクチャを備えたプロセッサを使用した場合のインタフェース回路です。Fig. 3と4の回路は最大システム構成であり、マイクロコントローラはDASの他の周辺デバイスにもアクセスし、そのため外部アドレス・ラッチとデコーダが必要であることを想定しています。ただし、アドレス・デコーダの規模と内容は、個々のシステムによって異なります。最小システム構成の場合は、DASはアドレス・ラッチとアドレス・デコーダのための最小の外部ロジックでマイクロコントローラに接続できます。これは、Fig. 5に示されています。ただし、この場合もCS信号がDAS内部のALEでラッチされるため、任意の上位アドレス・ビットでCS入力をドライブできることに注意してください。この構成では、DASのアクセスに使用するアドレス範囲が広がります(多くのビットはdon't care)が、そのアドレス範囲を他のメモリやI/Oロケーションに使用しないよう注意が必要です。例えば、ビットA15をCSのドライブに使用したとします。DASを選択するには、ビットA15は(インバータを経由するため)1でなければなりません。したがって、上位32kのアドレス範囲すべてがDASに使用されます。しかし、アドレス・ビットA5 ~ A14はdon't careであるため、DASは、上位32kのアドレス範囲の任意の場所へマッピングすることができません。

3.0 DASとHPCマイクロコントローラとのインタフェース

この章では、HPC46083マイクロコントローラとDAS間のインタフェース回路について具体的に述べます。HPC46083は、高性能16ビット・マイクロコントローラのHPCファミリを構成する製品です。HPCファミリには、種々のアプリケーション向けに多様なバージョンが用意されています。詳細情報、入手可能バージョン、仕様については、HPCファミリのデータシートをご覧ください。HPC46083は、16ビットのデータ/アドレス多重化バスを備えた16ビット・マイクロコントローラです。HPCファミリの下位製品ですが、必要なシステム・タイミング生成機能、内部ロジック、ROM、RAM、I/Oをすべて備え、各種のアプリケーションに応じた最適の制御機能を実現することができます。アドレス空間は、連続した64kバイトで、その中にすべてのメモリ、レジスタ、I/Oアドレス(メモリ・マップされたI/O)が含まれています。アドレス空間の最初の512バイト(0000H ~ 01FFH)には、256バイトの内蔵ユーザRAMと内部レジスタが配置されています。(アドレスの値は16進で、インジケータとしてサフィックス“H”を付けています。)アドレス空間の最後の8kバイト(E000H ~ FFFFH)には、主としてプログラムの格納に使用される内蔵ROMが配置されています。以下で述べるアプリケーションでは、HPC46083は、(単一チップ・モードではなく)外部アドレス範囲(0200H ~ DFFFH)にアクセス可能な拡張モードに設定されています。HPCファミリの外部データ・バスは、8ビット、16ビットのいずれにも構成でき、各種周辺デバイスとの効率的なインタフェースが可能です。

割込み処理は、HPC46083のベクトル割込み方式で行われます。HPC46083は、8種類の割込みソースに対応できます。そのうち4種類は、マスク可能な外部割込み入力です。割込み検知条件としては、LOWレベル、HIGHレベル、立上がりエッジ、立下がりエッジのうちいずれかを選択することが可能です。DASとのインタフェースにはは外部割込み入力のいずれか一つを使用します。なお、以下の説明ではHPC46083に言及する場合、“HPC”と表現することにします。

Fig. 7、8に2種類のインタフェース回路を示しました。Fig. 7の最初の回路は、外部アドレス・ラッチを包含し、全アドレス・デコーディングを行います。この構成では、HPCが他のアドレス範囲を使用してアクセスするデバイスが他にも存在することを想定しています。Fig. 8は、HPCにインタフェースされている周辺デバイスがDASのみ(または2、3個)であることを前提とした回路で、部分アドレス・デコーディングを採用してインタフェース・ロジックを簡略化しています。なお、図にそれぞれ示したアドレス・デコーディング方式は一般的なものであって、可能な種々の方式のうちのひとつだけではありません。これらの回路は、インタフェースに関する注意点を示すための手段として採用しました。他のロジック・ファミリやPALデバイスを使用して図とは異なるインタフェース回路を構成することも可能です。

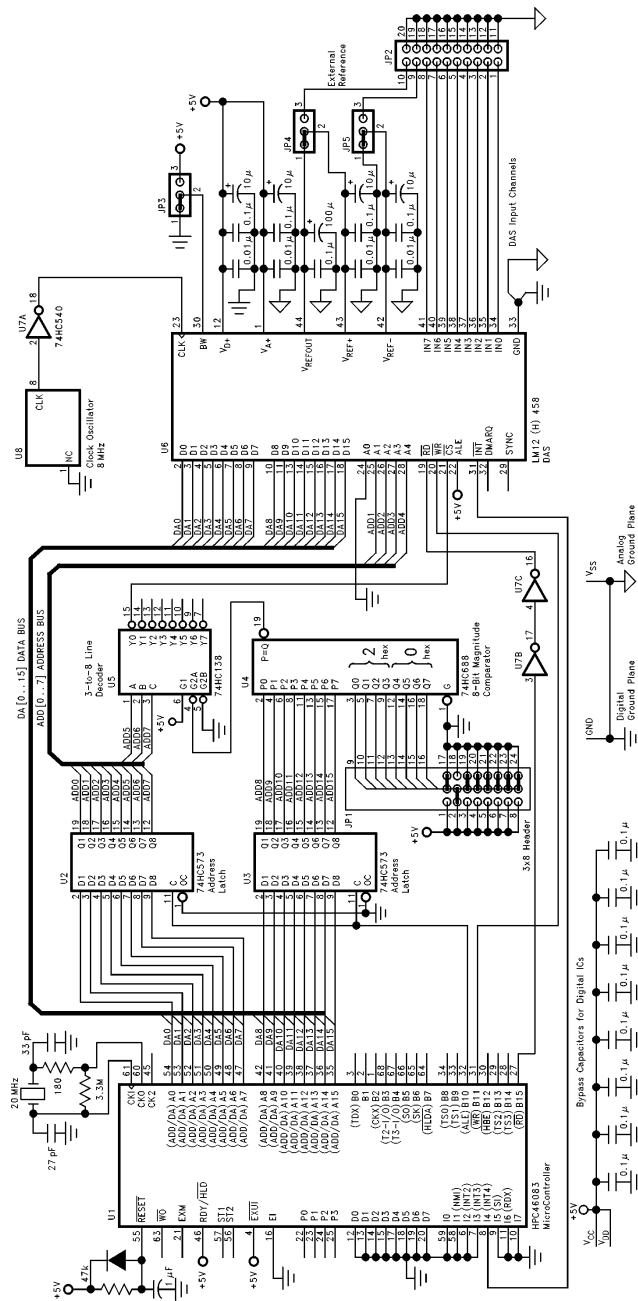


FIGURE 7. The DAS/HPC Microcontroller Interface (Complete Address Decoding)

3.1 全アドレス・デコーディング

Fig. 7に示した回路は、全アドレス・デコーディングによってDASのCS信号を生成します。DASは、外部アドレス範囲(0200H ~ 021BH)の先頭にメモリ・マップされたI/Oとしてアクセスされ、16ビット・データ・アクセスが選択されています。外部アドレス・ラッチU2とU3(74HC573)が、HPCの16ビットの多重化データ/アドレス・ラインに使用されているため、DASのALE入力はHIGHに結線しています。8ビットのマグニチュード・コンパレータU4(74HC688)は、ヘッダJP1上のアドレス範囲セレクト・ジャンパからの論理入力と上位アドレス・バイト(A15...A8)を比較することにより、上位アドレス・バイトのデコーディングを行います。Fig. 7の場合、同ジャンパは02Hに設定されています。したがって、マグニチュード・コンパレータの出力は、3入力8出力ライン・デコーダ・チップU5(74HC138)をイネーブルします。3入力8出力ライン・デコーダの出力はアドレス・ラインA5 ~ A7で、0200Hから021FHまでの32ロケーションのアドレス空間に対応する出力Y0がアクティブになります。この出力(Y0)は、DASのCS入力に使用されます。アドレス・ラインA1 ~ A4は、DASのアドレス入口に直接接続されています。DASのA0入力は、16ビット・データ・アクセスが選択され、A0はdon't careになるため、グラウンドに結線されています。DASは、アドレスが0200H ~ 021BHの28バイトのアドレス・ロケーションを使用します。

DASの信号タイミングでは、RDやWRの少なくとも20ns前にCSをアクティブにする必要があります。この条件は、HPCをクロック周波数20MHzで動作させた場合は満足することができません。図の回路では、アドレス・ラッチとデコーダにおける伝搬遅延時間を補償するため、2個の反転バッファ(U7)がRD制御ラインに置かれています。U7の必要性については、“タイミング分析”の項で詳述します。WRラインには、こうした遅延補償は不要です。

DASのINT出力は、HPCのINT4入力をドライブします。DASはこれにより、収集データが転送可能になった時点、あるいは、プロセッサの関与が必要な何らかの条件が生じた時に、HPCに処理を依頼することができます。なお、図ではINT4に接続していますが、他の外部割込み入力を利用してもかまいません。実際のシステムでは、プロセッサの割込み入力は、割込みを生成するI/Oデバイスの数とそれらデバイスの優先順位に基づいて選択します。

DASのクロックは、8MHzの水晶クロック・モジュールから供給されます。クロック・モジュールの出力は、DAS専用バッファリングされているため、DASのクロックがクリーンに保たれ、同じクロック・ラインを利用している他のデバイスによって生成、誘導される干渉は最小限に抑えられます。

3.2 最小アドレス・デコーディング

Fig. 8の回路では、外部アドレス・ラッチ(U2、U3)、8ビット・マグニチュード・コンパレータ(U4)、アドレス設定ジャンパ(JP1)は使用されていません。3入力8出力ライン・デコーダ(U5)は使用されており、アドレス/データ・ラインDA9とDA15でイネーブルされます。DA9は“1”にしてU5をイネーブルします。これは、DASとHPC内部のRAMとレジスタの間でアドレスの衝突が生じるのを防止するためです。DA9が“0”の場合、HPC内部のROMとレジスタはアドレス範囲0000H ~ 01FFFHを使用します。DA15もU5をイネーブルするため“0”にします。これも、DASとHPC内部のRAMとレジスタの間でアドレスの衝突が生じるのを防止するためです。DA15が“1”の場合、HPC内部のRAMとレジスタはアドレス範囲E000H ~ FFFFHを使用します。DASのCS入力は、この回路でもU5の出力Y0でドライブされます。DASのALE入力は、HPCのALE出力で直接ドライブされます。ALEは、DASのデータ転送サイクルが開始されるたびに、アドレス・ラインとCSラインをDAS内のラッチ回路にラッチします。

DASのアドレス範囲は、Fig. 7と同様に0200H ~ 021BHですが、多くのアドレス・ビットはdon't careになります。Fig. 8の場合、DASレジスタのアドレスを2進形式で表記すると、0XXX、XXIX、000P、PPP0のように表現されます。Pは、レジスタを特定するためのプログラム・ビットで、Xビットはdon't careになります。他のビットはこの通りにプログラムします。

3入力8出力ライン・デコーダ(U5)の出力Y1 ~ Y7も、Fig. 7と同様に他の周辺デバイスのアクセスに使用されます。これらの周辺デバイスも、アドレスとチップ・セレクトを取り込むための内蔵ラッチを備えている必要があります。たとえば、Fig. 8の回路では、最大8個のDASチップをHPCにインタフェースして、64のアナログ入力チャネルのモニタやデータ・ロギングなどを行うことができます。1個のDASだけをインタフェースする場合は、DASのCS入力は、点線の中间のようにU5の代わりに2個のゲートで生成してかまいません。

Fig. 8の回路の場合、重要となるDASのタイミング条件は、ALEの負エッジに対するアドレスとチップ・セレクトのセットアップ時間です。HPCを20MHzで動作させた場合は、この条件を満足することはできず、11.09MHzまで下げなければなりません。これについても、“タイミング分析”の項で詳しく述べます。

3.3 タイミング分析

ユーザは、インタフェースのハードウェア設計と並行して、プロセッサとDAS間において情報のやり取りが確実になされるかどうかタイミングの分析を行う必要があります。例えば、Fig. 7の場合、DASのRD入口に置かれたバッファは、タイミング条件を満足させるためのバッファです。同様に、Fig. 8におけるHPCのクロックは、適正なタイミングを確保するために低い周波数に設定されています。回路の新規設計に際しては、読出しサイクル、書込みサイクルに関するDASのタイミング仕様をHPC(または使用する他のプロセッサ)のタイミング仕様と比較することをお忘れなくください。両者のタイミング特性が整合していないと、ハードウェアの設計変更やソフトウェア手法による対応が必要になります。

3.3.1 全アドレス・デコーディング回路

DASとHPC(動作周波数20MHz)のスイッチング特性を調べると、読出しサイクルのタイミングが書込みサイクルよりも厳しいことがわかります。タイミング図(Fig. 9)に、読出しサイクルに関連する信号のタイミング関係を示しました。最初の3つの信号はHPCによって生成される信号(ALE、ADD/DATA、RD)で、図では、最も厳しいタイミング関係が示されています。他の3信号は、DASが受信するCSとRD、そして、HPCに送られるDATA(DAS)です。アドレスが有効になってから、アドレス・ラッチ(U3)、マグニチュード・コンパレータ(U4)、3入力8出力ライン・デコーダ(U5)を通ることによって生じるCSの最長伝搬遅延時間は、標準で54ns、最悪で116nsです。

Note: 74HCシリーズのロジック・デバイスの最大伝搬遅延時間は、電源電圧4.5V、負荷50pF、温度範囲-40 ~ 85における値です。標準遅延時間は、同様の条件で25°Cの時の値です。

RD(HPC)の立下がりエッジを基準とすると、この遅延時間は、16nsから78nsとなります。DASのRDは、CSの20ns後にアサートされる必要があります。したがって、DASに受信される前にHPCのRDラインに遅延回路を挿入しなければなりません。HPCがRD信号を立上げてから、36ns ~ 98ns(16 + 20ns ~ 78 + 20ns)後にその立下がりエッジがDASに到達するようにする必要があります。

反転バッファ U7B と U7C (75HC540) は、HPC と DAS の間において RD 信号に遅延時間を付与するためのものです。2 個のバッファの伝搬遅延時間の仕様は、標準 24ns、最大 50ns です。これは、上記の分析で割り出した所要遅延時間 36ns ~ 98ns に比べて短いものの、温度範囲 0 ~ 50 °C では十分です。この矛盾は、ロジック・デバイスの仕様シート上の動作要件（負荷容量電源電圧）が実際の回路よりも厳しいことを想定しているためです。ただし、最悪の条件でも回路を確実に動作させるには、RD ラインに遅延時間をさらに付与する必要があります。

2 番目のタイミング要件は、HPC の RD ラインの立下がりエッジを基準とするデータのセットアップ時間です。DAS のデータ出力は、RD ラインの立下がりエッジから標準で 10ns、最大で 80ns 遅れて有効となります。HPC が必要とするセットアップ時間は 45ns で、RD パルス幅（1 ウェイト状態）は 140ns ですから、RD バッファと DAS のデータ待ち時間による合計遅延時間は 95ns になります。この場合も、最悪条件では 45ns のセットアップ時間が確保されないことがあるため、HPC の書込みサイクルに余分の 1 ウェイト状態（100ns）を付加する必要があります。なお、ここで示した設計例は、検討を要する項目を示すための 1 例であり、個々のアプリケーションによって最適な設計は異なります。

Fig. 7 の回路は、開発システム“HPC デザイナーズ・キット”を利用して製作と試験を行っています。この開発システムは、HPC のリアルタイム・エミュレーションを実行し、HPC のすべての機能をアプリケーションに提供することができます。HPC デザイナーズ・キットは、プロセッサの実際のスイッチング特性をかなり忠実に実現します。

Fig. 10 は、HPC 開発システムを利用して構成した Fig. 7 の回路における CS、RD、WD 信号のスクープ写真です。Fig. 10a は、インバータを RD ラインに付加しなかった場合の読出しサイクルと書込みサイクルです。WR に対する CS のセットアップ時間は十分ですが、RD に対する CS のセットアップ時間は不十分です。Fig. 10b は、Fig. 10a の RD に対する CS のセットアップ時間を拡大した写真です。そのセットアップ時間は（室温において）18ns であり、20ns に極めて近いものの、回路と温度の変化に対する余裕は不足しています。Fig. 10c は、RD ラインにインバータを付加した場合の RD に対する CS のセットアップ時間を拡大した写真です。この時のセットアップ時間は 30ns に増加し、回路のパラツキや温度変化をカバーできる 10ns のマージンが確保されています。

3.3.2 最小アドレス・デコーディング回路

Fig. 8 の回路のスイッチング特性を調べると、HPC を 20MHz で動作させた場合、ALE の LOW に対する DAS のアドレスおよび CS のセットアップ時間が不足することがわかります。20MHz では、HPC は、ALE を LOW にすると、18ns（最小）後にはもう有効なアドレスを生成します（Fig. 9 参照）。DAS は、40ns のセットアップ時間を必要とします。これは、HPC のクロック周波数を下げることによって解決可能です。HPC を 10MHz で動作させると、セットアップ時間は最小で 43ns となります。DAS の CS に関しては、U5 で生じる遅延時間も考慮に入れる必要があります。これは U5 の入力から出力までの伝搬遅延時間で、アドレス・ラインが有効になってから、DAS の CS (U5 の Y0 出力) が LOW になるまでの時間です。

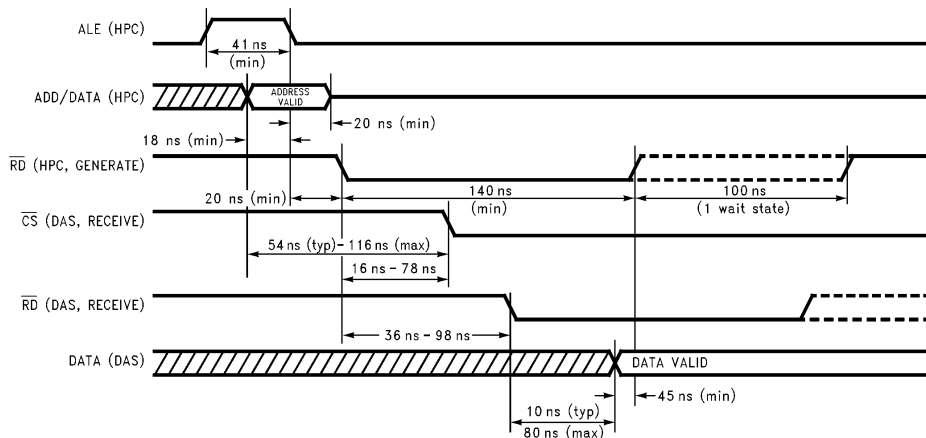
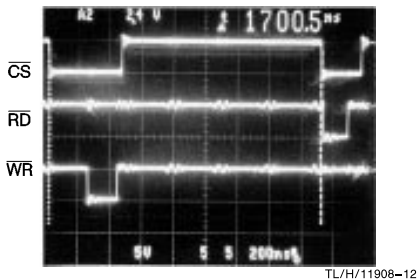
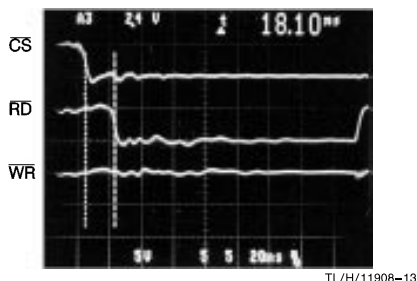


FIGURE 9. DAS/HPC Interface Timing Diagram (Complete Address Decoding)

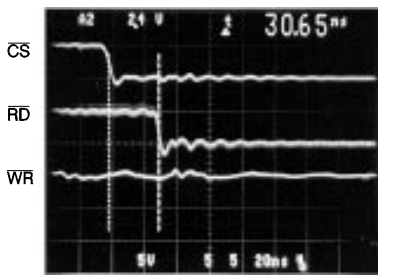
TL/H/11908-11



a) A Write and a Read Cycle,
No Inverter in RD Line



b) A Close Look at the CS to RD
Setup Time, No Inverter in RD Line

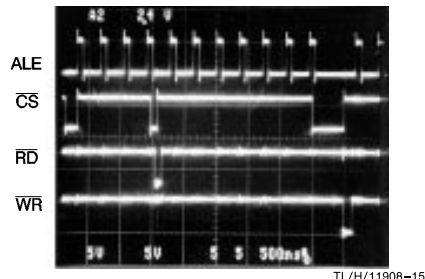


c) A Close Look at the CS to RD
Setup Time, 2 Inverters in RD Line

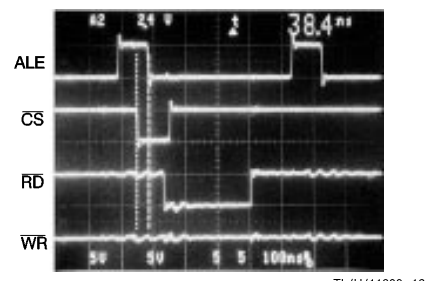
FIGURE 10. Scope Photos of CS, RD and
WR Signals at the DAS, Figure 7 Circuit

実際の測定では、室温の場合、HPCのクロック周波数を11.09MHZにすると、DASとHPCで確実なデータ転送が行われることがわかりました。

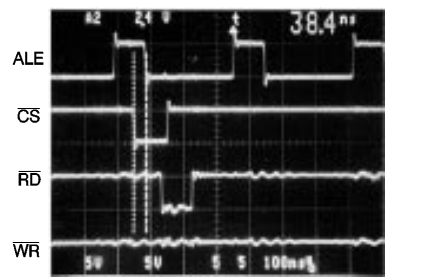
HPCのクロック周波数を下げた場合には、1ウェイト状態を付加する必要があったFig. 7の回路と異なり、ウェイト状態を追加しなくても外部データ転送を行うことができます。これは、外部読出しサイクルおよび書込みサイクルを早めるため、特にFIFOからの読出しを連続的に行う場合に有利です。



a) A Read and a Write Cycle with Zero Wait State



b) A Read Cycle with One Wait State



c) A Read Cycle with Zero Wait State

FIGURE 11. Scope Photos of ALE, CS, RD
and WR Signals at the DAS, Figure 8 Circuit

Fig. 8の回路も、HPC開発システムを利用して製作と試験を行っています。Fig. 11は、DAS入力におけるALE、CS、RD、WR信号のスクリーン写真です。Fig. 11aは、ウェイト状態を付加しない場合の読出しサイクルと書込みサイクルです。読出しパルスと書込みパルスの立ち上がりエッジがCS信号よりも遅れていることに注目してください。これは、CSが内部ラッチされるために問題ありません。Fig. 11bと11cは、1ウェイト状態を付加した場合とウェイト状態を付加しない場合の読出しサイクルを詳細に観察した様子です。ウェイト状態を付加しない方が、読出しサイクルが約180ns速いことがわかります。ALEのLOWに対するCSのセットアップ時間は約38nsです。

これは、DASの公表仕様値に比べまだ2ns足りません。実際の試験では確実なデータ転送がなされましたが、HPCのクロック周波数は、回路パラメータや温度の変化の最悪条件を考慮して10MHz以下に設定すべきです。

4.0 システム設計例: 半導体炉

このアプリケーション例の場合、DASは、半導体炉内の5個のセンサから送られる入力を測定します。炉データのデータ収集/制御システムとしては、Fig. 7または8の回路を使用するものとします。DASレジスタに設定する値は、以降に示すシステムの要件に基づいて決めます。DASの初期化とデータ収集を行うためのHPCの標準的なアセンブリ・ルーチンも示しています。Fig. 12に、半導体炉の測定に採用される標準的な構成を示します。流量センサは、半導体チャンバのダクトにおけるガス流量を測定します。圧力センサはチャンバ内の圧力を測定します。3個の温度センサは、炉内の中央部、および両端部の温度を測定します。

4.1 システムの要件と前提

炉の動作を制御するために、以下の5種類の測定を行う必要があります:

- T1における絶対温度、分解能12ビット。
 - T1に対するT2の相対温度、分解能12ビット+サイン・ビット
 - T1に対するT3の相対温度、分解能12ビット+サイン・ビット
 - チャンバを通過するガス流量F、分解能8ビット
 - チャンバ内の圧力P、分解能8ビット
- 以下の3種類のアラーム条件もモニタします:
- ガス流量F、上限値より大
 - ガス流量F、下限値より小
 - 圧力P、上限値より大

また、システムとして、以下の前提がなされています:

- センサからの信号はすべて、DAS入力へ入る電圧レベルが0V ~ 2.5Vの範囲に納まるように調整される(ゲインとオフセットの調整)。
- すべての信号調整回路の出力は、アナログ・グランドを基準としたシングル・エンド型。
- 流量センサ信号の調整回路出力の信号のソース・インピーダンスは600Ω。
- DASの基準電圧は2.5V、すなわち、 $V_{REF+} = 2.5V$ 、 $V_{REF-} = AGND$ 。
- 炉の測定・監視システムには、Fig. 7または8のいずれかの回路を使用する。以下の説明とプログラム・コードはいずれの回路にも有効。
- 測定結果のスループット・レートは50Hz程度が望ましい(20msごとに1組のデータを取り込み)。ただし、入力信号は変化が遅いため、スループット・レートが多少ずれてもシステム性能に影響することはない。

4.2 DASの結線とレジスタのプログラミング

システム要件に基づいて、DASの結線とレジスタの設定に進みます。

5個のセンサ出力は、DASの最初の5入力へ割り当てます:

- IN0:T1
- IN1:T2
- IN2:T3
- IN3:F
- IN4:P
- IN5:未使用 - GNDへ結線
- IN6:未使用 - GNDへ結線
- IN7:未使用 - GNDへ結線

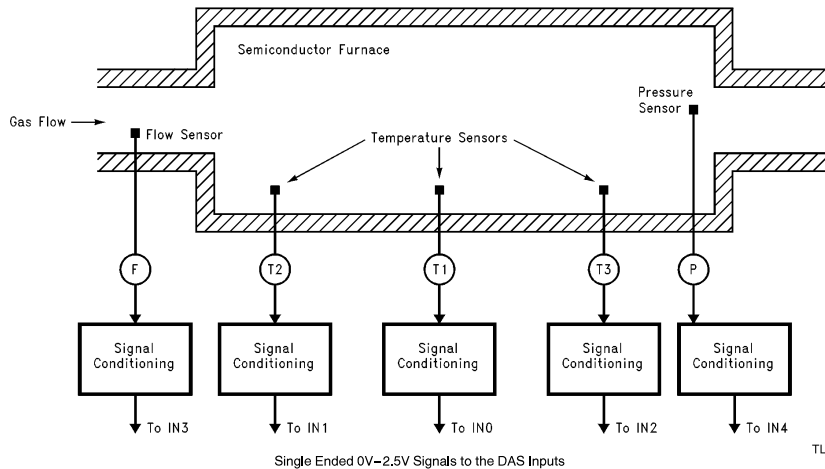


FIGURE 12. Diagram of a Typical Measurement Arrangement in a Semiconductor Furnace

測定とリミットの監視に7個のDAS命令が必要です。5個は変換中で、2個は“F”と“P”をプログラムされたリミットと比較する“ウォッチドッグ”機能を実行します。なお、変数“F”は、1個の命令だけで上下限値のモニタが可能です。

システム動作としては、以下の手順を想定しています：

- 7個の命令は、0から6の順番で遅延時間ゼロで実行する。
- 命令#6の実行後、DASは、命令#0に戻って実行を続ける。各ループは、命令ループと呼ばれます。
- スループット・レートが50Hzになるように、命令#0を実行する前にタイマ・レジスタを利用して遅延時間を付加する。
- 1回の命令ループで、5個の変換結果を生成する。FIFOが30個(6X5)の変換結果で満たされると、マイクロコントローラが読み出す。これは、指定された数の変換結果がFIFOに格納された時点で(DASからHPCへ)割込みを出すことによって実現する。

命令レジスタ：

- Sync および Pause ビットは使用しません。

命令レジスタの定義：

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Acquisition Time				W-dog	8/12	Timer	Sync	V _{IN} ⁻				V _{IN} ⁺		Paus.	Loop

命令# 0: T1 の測定、シングル・エンド、12 ビット、タイマをイネーブル

$V_{IN}^+ = IN0 (T1)$, $V_{IN}^- = AGND$

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

命令# 1: T1 - T2 の測定、差分モード、12 ビット+サイン

$V_{IN}^+ = IN0 (T1)$, $V_{IN}^- = IN1 (T2)$

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

命令# 2: T1 - T3 の測定、差分モード、12 ビット+サイン

$V_{IN}^+ = IN0 (T1)$, $V_{IN}^- = IN2 (T3)$

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

命令# 3: F の測定、シングル・エンド、8 ビット、D = 2

$V_{IN}^+ = IN3 (F)$, $V_{IN}^- = AGND$

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	0	0	1	0	0	0	0	0	1	1	0	0

命令# 4: ウォッチドッグ・モード、F、シングル・エンド、D = 2

$V_{IN}^+ = IN3 (F)$, $V_{IN}^- = AGND$

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	0	1	0	0	0	0	0	0	0	1	1	0	0

命令# 5: P の測定、シングル・エンド、8 ビット

$V_{IN}^+ = IN4 (P)$, $V_{IN}^- = AGND$

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0

命令# 6: ウォッチドッグ・モード、P、シングル・エンド、ループ・ビットをイネーブル

$V_{IN}^+ = IN4 (P)$, $V_{IN}^- = AGND$

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1

命令# 7: 不使用

- FIFOの読み出し中に変換を停止させることはしない。読み出しは、最後の比較命令(#6)の期間中と命令#0の前に設定された遅延期間中に行う。読み出しのため、6命令から成る各ループの終わりに余分な遅延時間が付加されますが、この遅延時間は20msのループ期間と比べれば無視できます。(変換中の読み出しと読み出し/書き込み中の内部クロックの停止については、2.3節を参照のこと。)

流量センサからの入力には600Ωのソース・インピーダンスを有するため、アキュイジション時間が他に比べて長くなります。8ビットおよび“ウォッチドッグ”モードの場合、命令RAMのビットD12 ~ D15にプログラムするアキュイジション時間値(D)は、DASデータシートの式にしたがい、 $2(D - 0.36 \times R_s(k\Omega) \times f_{clk}(MHz)) = 0.36 \times 0.6 \times 8 = 1.73$ としてください。

ここでDASレジスタの内容は、以上に基づいて指定できます。

命令#4と#6はリミット値も指定します。命令#4は2個のリミット値、命令#6は1個のリミット値を指定します。リミット値は、F_MIN、F_MAX、P_MAX と表記されている場所に設定します。

命令RAM、リミットの定義:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Don't Care							>/<	Sign	Limit						

命令#4、リミット#1

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	1	0	F_MAX							

命令#4、リミット#2

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	F_MIN							

命令#6、リミット#1

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	1	0	P_MAX							

命令#6、リミット#2、間違った割込みが生じないように負の最大値をリミットとして指定

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

割込みイネーブル・レジスタ:

- INT0:比較リミット: イネーブル
- INT1:命令番号: ディセーブル
- INT2: FIFO フル: イネーブル
- INT3:自動ゼロ調整完了: ディセーブル
- INT4:キャリブレーション完了: イネーブル
- INT5:停止: ディセーブル
- INT6:電源異常: ディセーブル
- INT7:スタンバイ・リターン: ディセーブル
- 命令番号の指定: 0、未使用
- FIFO 内の結果格納数指定: 30(11110:2進)

割込みイネーブルレジスタ

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Number of results in FIFO				Instruction Number				INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0
1	1	1	1	0	0	0	0	0	0	0	1	0	1	0	1

構成・レジスタ:

- 各変換前に自動ゼロ調整とキャリブレーションは行いません。
- 変換結果のビットD13 ~ D15 に命令番号。
- Sync ビットは使用せず、入力、出力いずれにも設定可。

構成・レジスタ、変換開始コマンド

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Don't Care				Diag.	Test	RAM Pointer		Sync I/O	A/Z Each	Chan Mask	Stand-by	Full Cal	Auto Zero	Reset	Start
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

構成・レジスタ、リセット・コマンド

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

構成・レジスタ、フル・キャリブレーション・コマンド

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

構成・レジスタ、RAMバンク1選択コマンド(変換はストップされます)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

構成・レジスタ、変換停止コマンド

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

タイマ・レジスタ:

タイマの事前設定値を計算する場合、まず、合計命令実行時間を求める必要があります。下表に、各命令に使用されるクロック・サイクル数を示しました。状態とその継続時間については、データシートの4.0章(シーケンサ)をご覧ください。

Instruction #	State 0	State 1	State 7	State 6	State 4	State 5	Number of Clock Cycles
0	1	1	9			44	55
1	1	1	9			44	55
2	1	1	9			44	55
3	1	1	2			21	25
4	1	1	2	5	1	5	15
5	1	1	6			21	29
6	1	1	6	5	1	5	19
Total:							253

命令の実行には、合計253クロック・サイクルを要します。タイマ遅延は、2クロック・サイクル以上に設定されるため、これを含めると、

$$253 + 2 = 255 \text{ クロック・サイクル}$$

となります。255クロック・サイクルは、

$$255 \times 1/8\text{MHz} = 31.875\mu\text{s}$$

に相当します。したがって、20msからこの31.875μsを引いた時間

$$20\text{ms} - 31.875\mu\text{s} = 19.968\text{ms}$$

をタイマ遅延としなければなりません。

タイマの1ステップは32クロック・サイクルですから、8MHzクロック(125ns周期)では、

タイマカウントは

$$32 \times 125\text{ns} = 4\mu\text{s}$$

に相当します。

したがって、タイマに事前設定する値は、

$$19.968\text{ms} \div 4\mu\text{s} = 4992$$

となります(16進では1380H)。

以上で必要なDASレジスタの設定は終わりました。次は、DASとの相相互作用をプロセッサにプログラムします。

4.3 マイクロコントローラのプログラミング

マイクロコントローラとDAS間のやり取りは、基本的に読出しおよび書込み動作で実行されます。マイクロコントローラは、DASとの相相互作用を外部データ転送命令を使用することによって行います。

DASへの書込み動作には2個の変数が必要で、これは、DASレジスタのアドレスと書込むデータです。DASからの読出し動作に必要な変数は、レジスタのアドレスだけです。以下の例では、DASレジスタのアドレスとデータの名称はそれぞれ、DAS_REG_ADD、DAS_DATAとしています。

下に、HPC、8051および68HC11マイクロコントローラ・ファミリ用のアセンブリ・モニタの例を示しました：

HPCファミリは、16ビットのデータをメモリ・マップされたI/Oに直接書込むことができます。その基本的なデータ転送命令は“LD”(ロード)です。読出し、書込みは一つの命令で実行されます。

書込み

```
LD    DAS_REG_ADD, #DAS_DATA
```

読出し

```
LD    [destination], DAS_REG_ADD
```

8051ファミリでは、メモリ・マップされた外部I/Oへのアクセスは、DPTR(データ・ポインタ)を介して間接的に行います。書込みの場合は、データを前もってアキュムレータヘロードしておく必要があります。読出しでも、このアキュムレータに一旦データが格納されます。転送は8ビットずつ行われるため、16ビットのデータは2サイクルを必要とします。基本命令は“MOV”(move)と“MOVX”(move external)、“INC”(increment)です。

書込み

```
MOV    DPTR, #DAS_REG_ADD
MOV    A, DAS_DATA(low byte)
MOVX   @DPTR, A
INC    DPTR
MOV    A, DAS_DATA(high byte)
MOVX   @DPTR, A
```

読出し

```
MOV    DPTR, #DAS_REG_ADD
MOVX   A, @DPTR
MOV    [destination], A
INC    DPTR
MOVX   A, @DPTR
```

上記の例では、16ビットのアドレス指定を採用していますが、8051のR0またはR1をDPTRの代わりに使用して8ビットのアドレス指定も可能です。

68HC11ファミリは、メモリ・マップされた外部I/Oに直接アクセスすることができます。書込みの場合は、データを前もってアキュムレータヘロードしておく必要があります。読出しでも、このアキュムレータに一旦データが格納されます。68HC11は8ビット・プロセッサですが、ダブル・アキュムレータ(A+B、Dと呼ぶ)を利用する16ビット・データ転送命令を備えており、一つの命令で2転送サイクルを実行します。基本となる命令は“LDD”(load double accumulator)と“STD”(store double accumulator)です。

書込み

```
LDD    #DAS_DATA
STD    DAS_REG_ADD
```

読出し

```
LDD    DAS_REG_ADD
```

4.3.1 半導体炉用HPCアセンブリ・ルーチンの例

Fig. 13に示したプログラムは、半導体炉に適用した場合のHPCアセンブリ・ルーチンの1例です。図では、DASの初期化とDASの割込みサービス・ルーチンだけが示されています。プログラム全体としては、データ操作や制御機能もすべて記述しなければなりません、ここでは説明を省略します。

ルーチンは、Fig. 2のフローチャートの手順に忠実にしたがっており、理解しやすいようにコメントも豊富に入れています。ルーチンはまた、フローチャートの各セクションにしたがって区分がなされています。ルーチンとフローチャートの主たる差異は、割込みサービス・ルーチンの開始でDASが停止されないことです。

割込みサービス・ルーチンは、IFBIT(if bit is true)命令を使用して割込みステータス・ビットの状態を調べています。これは、制御アプリケーションにおいて非常に便利な手法として利用できます。

READ_FIFOルーチンは、FIFOの内容を読み出し、それをDAS_RESULTという名前のロケーションから始まる指定のメモリ・ブロックに格納します。ブロックのサイズ、すなわち、読出すFIFOロケーションの数はプログラムすることができます(図では30ロケーション)。このルーチンはわずか5行のアセンブリです。複合機能命令“XS”(exchange and skip)によって、データの転送、アドレスのインクリメントまたはデクリメント、そして、比較判定を行わせています。READ_FIFOルーチンの最初のLD命令は、HPCのBおよびKレジスタに、メモリ・ブロックの先頭アドレスと最終アドレスをロードする命令です。2番目の命令は、FIFOからワード(16ビット)を読み出して“A(アキュムレータ)にロードする命令です。XS命令は、Bによって指示されているメモリ・ロケーションにAを格納してBを2だけ(2バイト分)インクリメントし、BとKを比較してメモリ・ブロックの最終アドレスかどうかを調べます。最終アドレスにまだ達していなければ、プログラムは前に戻ってFIFOの次のワードをロードします。達していれば、プログラムは“JP”(jump)命令をスキップしてサービス・ルーチンから出ます。

```

1
2      #NOHEADING
3      #PWIDTH= 112
4
5      ;*****
6      ;*
7      ;* AN HPC ASSEMBLY ROUTINE FOR THE SEMICONDUCTOR APPLICATION EXAMPLE FOR THE *
8      ;* APPLICATION NOTE "INTERFACING THE LM12454/8 DATA ACQUISITION SYSTEM *
9      ;* CHIPS TO MICROPROCESSORS AND MICROCONTROLLERS" *
10     ;*
11     ;* BY: FARID SALEH *
12     ;* DATE: 7/27/93 *
13     ;*****
14
15
16     ;***** HPC REGISTERS SYMBOLIC DEFINITIONS *****
17
18 00C0      PSW      = 0C0H      ;PROCESSOR STATUS REGISTER
19      ;      SP      = 0C4H      ;STACK POINTER
20      ;      PC      = 0C6H      ;PROGRAM COUNTER
21      ;      A       = 0C8H      ;ACCUMULATOR
22      ;      K       = 0CAH      ;K REGISTER
23      ;      B       = 0CCH      ;B REGISTER
24      ;      X       = 0CEH      ;X REGISTER
25 00E2      PORTB    = 0E2H      ;PORT B DATA REGISTER
26 00F2      DIRB     = 0F2H      ;PORT B DIRECTION REGISTER
27 00F4      BFUN     = 0F4H      ;PORT B ALTERNATE FUNCTION REGISTER
28 00D0      ENIR     = 0D0H      ;INTERRUPT ENABLE REGISTER
29 00D4      IRCD     = 0D4H      ;INTERRUPT / INPUT CAPTURE CONDITION REGISTER
30 00D2      IRPD     = 0D2H      ;INTRUPT PENDING REGISTER
31
32     ;***** DAS REGISTERS / VARIABLES / CONSTANTS SYMBOLIC DEFINITIONS *****
33
34 0200      INSTR0    = 0200H      ;DAS INSTRUCTION REGISTER ADDRESSES
35 0202      INSTR1    = 0202H      ;READ/WRITE REGISTERS
36 0204      INSTR2    = 0204H      ; "
37 0206      INSTR3    = 0206H      ; "
38 0208      INSTR4    = 0208H      ; "
39 020A      INSTR5    = 020AH      ; "
40 020C      INSTR6    = 020CH      ; "
41 020E      INSTR7    = 020EH      ; "
42
43 0210      CONFIG    = 0210H      ;DAS CONFIGURATION REGISTER ADDRESS, R/W
44 0212      INTEN     = 0212H      ;DAS INTERRUPT ENABLE REG. ADDRESS, R/W
45 0214      INTSTAT    = 0214H      ;DAS INTERRUPT STATUS REG. ADD. READ ONLY
46 0216      TIMER     = 0216H      ;DAS TIMER REG. ADDRESS, R/W
47 0218      FIFO      = 0218H      ;DAS FIFO ADDRESS, READ ONLY
48 021A      LMTSTAT    = 021AH      ;DAS LIMIT STATUS REG. ADD. READ ONLY
49
50 01C0      DAS_RESULT = 01C0H      ;START ADDRESS OF TOP 64 LOCATION OF HPC
51      ;ON-CHIP RAM TO STORE CONVERSION RESULTS
52 00BC      DAS_INT_MEM = 00BCH      ;HPC MEMORY LOCATION TO STORE INTERRUPT
53      ;STATUS REGISTER
54 00BE      DAS_LIM_MEM = 00BEH      ;HPC MEMORY LOCATION TO STORE LIMIT STATUS
55      ;REGISTER
56 001E      FIFO_CNT   = 30          ;NUMBER OF RESULTS IN FIFO, A DECIMAL VALUE
57 1380      TIMER_SET  = 01380H      ;TIMER PRESET VALUE
58
59 00FF      F_MAX      = 0FFH        ;HIGH LIMIT FOR GAS FLOW
60 0000      F_MIN      = 000H        ;LOW LIMIT FOR GAS FLOW
61 00FF      P_MAX      = 0FFH        ;HIGH LIMIT FOR PRESSURE
62
63 00BA      FLAGS      = 00BAH        ;GENERAL SOFTWARE FLAGS BYTE
64 0000      CAL_FLG    = 0           ;BIT 0 OF FLAGS BYTE FOR CALLIBRATION
65
66 0000      .SECT PDASTST,ROM16
67

```

FIGURE 13. HPC Assembly Program Listing

TL/H/11908-18

```

68 ;***** HPC INITIALIZATION *****
69
70 0000 START:
71 0000 9718C0 LD PSW.B,#018H ;PROCESSOR STATUS WORD,EXPANDED MODE
72 ;1 WAIT STATE, CAN BE ZERO WAIT STATE
73 ;FOR CIRCUIT IN FIGURE 8, #01CH
74 0003 9700D0 LD ENIR.B,#00H ;DISABLE ALL INTERRUPTS
75 0006 9700D2 LD IRPD.B,#00H ;CLEAR ANY INTERRUPT PENDING BIT
76 0009 9700D4 LD IRCD.B,#00H ;SET I4 FOR HIGH TO LOW EDGE DETECT
77 000C B70000E2 LD PORTB.W,#00H ;PORT B ALL ZERO
78 0010 B7FFFFFF2 LD DIRB.W,#0FFFFFFH ;PORT B ALL OUTPUTS, B10,11,12 AND 15
79 ;ARE PREDEFINED DUE TO EXPANDED MODE
80 ;NOTE: PORT A IS ALSO PREDEFINED
81 0014 B70000F4 LD BFUN.W,#00H ;PORT B NO ALTERNATE FUNCTION
82
83 ;***** DAS REGISTERS INITIALIZATION *****
84
85 0018 83020210AB LD CONFIG.W,#0002H ;RESET, SELECT RAM SECTION 0, RP=00
86
87 001D 8702000200AB LD INSTR0.W,#0200H ;INSTRUCTIONS INITIALIZATION, VALUES
88 0023 83200202AB LD INSTR1.W,#0020H ;ARE AS SPECIFIED ON THE SYSTEM
89 0028 83400204AB LD INSTR2.W,#0040H ;DESIGN
90 002D 87240C0206AB LD INSTR3.W,#0240CH ; "
91 0033 87280C0208AB LD INSTR4.W,#0280CH ; "
92 0039 870410020AAB LD INSTR5.W,#0410H ; "
93 003F 870811020CAB LD INSTR6.W,#0811H ; "
94
95 0045 8701000210AB LD CONFIG.W,#0100H ;SELECT RAM SECTION 1, RP=01
96
97 004B 8702FF0208AB LD INSTR4.W,#(0200H+F_MAX) ;HIGH LIMIT FOR INSTRUCTION 4
98 0051 8702FF020CAB LD INSTR6.W,#(0200H+P_MAX) ;HIGH LIMIT FOR INSTRUCTION 6
99
100 0057 8702000210AB LD CONFIG.W,#0200H ;SELECT RAM SECTION 2, RP=10
101
102 005D 83000208AB LD INSTR4.W,#(0000H+F_MIN) ;LOW LIMIT FOR INSTRUCTION 4
103 0062 870100020CAB LD INSTR6.W,#0100H ;LOW LIMIT FOR INSTRUCTION 6
104 ;NEGATIVE FULL-SCALE, NOT USED
105
106 0068 87F0150212AB LD INTEN.W,#((FIFO_CNT*2048)+015H) ;SHIFT FIFO_CNT TO MSBS OF HIGH
107 ;BYTE, THEN ADD LOW BYTE (015H),
108 ;DAS INT # 0, 2 AND 4 ARE ENABLED
109 ;TIMER INITIALIZED WITH PRESET VALUE
110 006E 8713800216AB LD TIMER.W,#TIMER_SET
111
112 ;***** ENABLING HPC INTERRUPT #4 *****
113
114 0074 9711D0 LD ENIR.B,#011H ;ENABLE HPC GLOBAL AND INTERRUPT #4
115
116 ;***** DAS FULL CALIBRATION *****
117
118 0077 96BA08 SBIT CAL_FLG,FLAGS.B ;SET CALIBRATION FLAG FOR PROGRAM
119 ;CONTROL
120 007A 83080210AB LD CONFIG.W,#0008H ;DAS CALIBRATION IS STARTED
121
122 007F 96BA10 WAIT1: IFSBIT CAL_FLG,FLAGS.B ;CHECK FOR CAL_FLG, IF 1 WAIT,
123 ;IDEL LOOP UNTIL CALIBRATION IS DONE
124 0082 63 JP WAIT1 ;AND INTRRUPT FROM DAS IS RECEIVED,
125 ;IN A COMPLETE PROGRAM, PROCESSOR
126 ;CAN DO OTHER TASKS
127
128 ;***** STARTING THE CONVERSIONS *****
129
130 0083 83010210AB LD CONFIG.W,#0001H ;START BIT = 1, DAS STARTS
131
132 0088 60 WAIT2: JP WAIT2 ;IDEL LOOP FOR HPC TO WAIT FOR DAS
133 ;INTERRUPT
134 ;THIS IS MAINLY A TEST STATEMENT HERE
135 ;AND IN A COMPLETE PROGRAM PROCESSOR
136 ;IS DOING OTHER TASKS
137
138

```

FIGURE 13. HPC Assembly Program Listing(つづき)

TL/H/11908-19

```

139          ;***** THE DAS INTERRUPT SERVICE ROUTINE *****
140
141 FFF6 8900      R .IPT 4, DAS_INTSERV          ;ASSEMBLER INTERRUPT ADDRESS DIRECTIVE
142
143 0089          DAS_INTSERV:
144
145 0089 AFC8      PUSH    A          ;PUSH INSTRUCTIONS TO SAVE REGISTER
146 008B AFCC      PUSH    B          ;CONTENTS ON STACK, A, B, K, X AND
147 008D AFCA      PUSH    K          ;FSW ARE SHOWN AS GENERAL PURPOSE
148 008F AFCE      PUSH    X          ;REGISTERS, IF REGISTERS ARE NOT USED
149 0091 AFC0      PUSH    PSW.W      ;IN INTERRUPT SERVICE ROUTINES, THEY
150                                     ;CAN BE DELETED FROM THE LIST
151
152 0093 A40214BCAB LD      DAS_INT_MEM.W,INTSTAT.W ;STORE CONTENTS OF DAS INTERRUPT REG.
153                                     ;IN HPC MEMORY FOR BIT TESTING
154
155          ;***** INDIVIDUAL BITS IN THE DAS_INT_MEM ARE TESTED AND DIFFERENT *****
156          ;***** ROUTINES WILL SERVE INDIVIDUAL CASES
157
158 0098 96BC12     IFBIT    2,DAS_INT_MEM.B      ;IF INTERRUPT IS FROM FIFO FULL
159 009B 4E         JP      READ_FIFO            ;JUMP TO ROUTINE READ_FIFO
160                                     ;OTHERWISE TEST THE NEXT BIT
161 009C 96BC10     IFBIT    0,DAS_INT_MEM.B      ;IF ANY LIMITS IS PASSED JUMP TO
162 009F 49         JP      DAS_LIMIT            ;ROUTINE DAS_LIMIT FOR ACTION
163                                     ;OTHERWISE MOVE ON
164 00A0 83020210AB LD      CONFIG.W,#0002H      ;IF NON OF THE ABOVE BITS MUST BE
165 00A5 96BA18     RBIT    CAL_FLG,FLAGS.B      ;CALIBRATION COMPLETE, RESET THE DAS
166 00A8 4C         JP      DONE                ;AND CAL_FLG, THEN RETURN
167
168          ;***** SERVICE ROUTINE DAS_LIMIT *****
169
170 00A9          DAS_LIMIT:
171
172          ;BODY OF THE SERVICE ROUTINE
173          ;THIS ROUTINE SHOULD READ THE DAS LIMIT STATUS REGISTER AND TEST THE
174          ;NECESSARY BITS, BASED ON WHAT BIT IS SET THE PROPER ACTION IS TAKEN
175          ;
176 00A9 4B         JP      DONE
177
178          ;***** SERVICE ROUTINE READ_FIFO *****
179
180          READ_FIFO:
181 00AA          BK.W,#DAS_RESULT,#(DAS_RESULT*2*FIFO_CNT-2)
182                                     ;LOAD B FOR STARTING ADDRESS OF THE
183 00AA A701C001FA LD      BK.W,#DAS_RESULT,#(DAS_RESULT*2*FIFO_CNT-2)
184                                     ;BLOCK TO BE FILLED WITH FIFO,
185                                     ;SET K FOR UPPER LIMIT OF THE BLOCK
186
187 00AF B60218A8     LPFIFO: LD      A,FIFO.W      ;LOAD ACC WITH FIFO CONTENTS, FIFO
188                                     ;POINTER IS INCREMENTED ON EACH READ
189 00B3 E1          XS      A,[B+].W              ;STORE ACC TO THE HPC's RAM WITH B
190                                     ;AUTO-INCREMENT AND SKIP IF GREATER
191                                     ;THAN K
192 00B4 65          JP      LPFIFO
193
194 00B5 3FC0         DONE: POP      PSW.W          ;RELOAD THE SAVED REGISTERS BACK
195 00B7 3FCE        POP      X                  ;FROM STACK
196 00B9 3FCA        POP      K
197 00BB 3FCC        POP      B
198 00BD 3FC8        POP      A
199
200 00BF 3E         RETI                          ;RETURN FROM INTERRUPT ROUTINE
201
202 00C0          .END START

```

**** Errors: 0, Warnings: 0

TL/H/11908-20

FIGURE 13. HPC Assembly Program Listing(つづき)

付録A: レジスタビット配置とプログラマーズ・ノート

構成レジスタ (読み出し/書き込み)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Don't Care				Diag.	Test	RAM Pointer		Sync I/O	A/Z Each	Chan Mask	Stand-by	Full Cal	Auto Zero	Reset	Start

- D0: - Start:0 = 命令実行をストップする。1 = 命令実行をスタートする。
- D1: - Reset:1 にセットされると、Start ビットとステータス・レジスタ内の全ビット、そして命令ポインタをゼロにリセットし、2クロック・パルス後には本ビットも自動的にゼロにリセットされる。
- D2: - Auto - Zero:1 にセットされると、長い自動ゼロ・キャリブレーション・サイクルが実行される。
- D3: - Full Calibration:1 にセットされると、フル・キャリブレーション・サイクルが実行される。
- D4: - Standby:1 にセットされると、チップは低消費電力のスタンバイ・モードに移行し、0 にリセットされると、チップはパワー・アップ時間を経てアクティブ・モードに復帰する。
- D5: - Channel Mask:0 = 変換結果のビット 13 ~ 15 に、それに対応する命令の番号が示される。1 = 変換結果のビット 13 ~ 15 には、拡張サイン・ビットが示される。
- D6: - A/Z Each:1 にセットすると、変換ごとに短い自動ゼロ・サイクルが前もって実行される。
- D7: - Sync I/O:0 = Sync ピンは入力、1 = Sync ピンは出力。
- D9 - D8: - RAM Pointer:命令 RAM のセクションを選択する。00 = 命令、01 = リミット # 1、10 = リミット # 2。
- D10: - 出荷試験用のビット。通常の操作ではゼロに固定すること。
- D11: - Diagnostic:1 にセットすると、選択された通常の命令と並行して診断用の変換を実行する。
- D15 - 12: - Don't care.

プログラマーズ・ノート

構成レジスタ: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

構成レジスタ: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

構成レジスタ: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

付録A: レジスタビット配置とプログラマーズ・ノート

構成レジスタ (読出し / 書込み) (つづき)

命令:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Acquisition Time				W-dog	8/12	Timer	Sync	V _{IN} -				V _{IN} +		Paus.	Loop

- D0: - Loop:0 = 次の命令に進む。1 = 命令 # 0 に戻る。
- D1: - Pause:0 = 停止しない。1 = 停止する (命令を実行しない)。Pause = 1 を検出すると、構成レジスタの Start ビットが 0 にリセットされ、Start ビットに 1 が書込まれると命令の実行が再び開始される。
- D4 - D2: - V_{IN}⁺: A/D の非反転入力に接続する入力チャネルを選択する。
- D7 - D5: - V_{IN}⁻: A/D の反転入力に接続する入力チャネルを選択する。
- D8: - Sync:0 = 正常動作。内部タイミング。1 = S/H と変換 (比較) タイミングが SYNC 入力ピンで制御される。
- D9: - Timer:0 = タイマ動作なし。1 = 命令実行はタイマがゼロにカウントダウンしてから開始。
- D10: - 8/12:0 = 分解能 12 ビット + サイン。1 = 分解能 8 ビット + サイン。
- D11: - Watchdog:0 = ウォッチドッグ比較なし。1 = ウォッチドッグ比較を実行する。
- D15 - D12: - Acquisition Time: S/H アクイジション時間を決定する。12 ビット + サインの場合は (9 + 2D) クロック・サイクル、8 ビット + サインの場合は (2 + 2D) クロック・サイクル。
D = D15 - D12 の内容、R_S = 入力ソース・インピーダンス。
12 ビット + サインの場合は、 $D = 0.45 \times R_S [\text{k}\Omega] \times f_{\text{CLK}} [\text{MHz}]$
8 ビット + サインの場合は、 $D = 0.36 \times R_S [\text{k}\Omega] \times f_{\text{CLK}} [\text{MHz}]$

プログラマーズ・ノート

命令 # 0: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

命令 # 1: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

命令 # 2: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

命令 # 3: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

付録A:レジスタのビット配置とプログラマーズ・ノート

命令 RAM(読出し/書込み)(つづき)

プログラマーズ・ノート

命令 # 4: アドレス:

シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

命令 # 5: アドレス:

シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

命令 # 6: アドレス:

シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

命令 # 7: アドレス:

シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

付録A:レジスタのビット配置とプログラマーズ・ノート

命令 RAM(読出し/書込み):(つづき)

リミット #1

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Don't Care						>/<	Sign	Limit							

D7 - D0: - Limit:8 ビットのリミット値。

D8: - Sign:リミット値のサイン・ビット。0 = 正、1 = 負。

D9: - >/<:上限値であるか下限値であるかを指定。0 = 入力のリミットを下回ると割込みが生成される。1 = 入力のリミットを上回ると割込みが生成される。

D15 - D10 - Don't care。

プログラマーズ・ノート

命令 # 0:リミット # 1:アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

命令 # 1:リミット # 1:アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

命令 # 2:リミット # 1:アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

命令 # 3:リミット # 1:アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

命令 # 4:リミット # 1:アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

命令 # 5:リミット # 1:アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

命令 # 6:リミット # 1:アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

命令 # 7:リミット # 1:アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16進値

付録 A: レジスタのビット配置とプログラマーズ・ノート

命令 RAM (読出し / 書込み) (つづき)

リミット #2

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Don't Care						>/<	Sign	Limit							

D7 - D0: - Limit: 8 ビットのリミット値。

D8: - Sign: リミット値のサイン・ビット。0 = 正、1 = 負。

D9: - > / <: 上限値であるか下限値であるかを指定。0 = 入力のリミットを下回ると割込みが生成される。1 = 入力のリミットを上回ると割込みが生成される。

D15 - D10 - Don't care。

プログラマーズ・ノート

命令 # 0: リミット # 2: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

命令 # 1: リミット # 2: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

命令 # 2: リミット # 2: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

命令 # 3: リミット # 2: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

命令 # 4: リミット # 2: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

命令 # 5: リミット # 2: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

命令 # 6: リミット # 2: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

命令 # 7: リミット # 2: アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

付録A:レジスタのビット配置とプログラマーズ・ノート

割込みイネーブル・レジスタ(読出し/書込み)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Number of results in FIFO to Generate Interrupt (INT2)					Instruction Number to Generate Interrupt (INT1)			INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0

ビット # 0 - 7 は、1 にセットされると、それぞれ以下に示す割込み生成条件をイネーブルします。

- D0: - INT0:ウォッチドッグ・モードでリミットを越えると割込みを生成する。
- D1: - INT1:D10 - D8 に指定した命令が実行段階になった時点で割込みを生成する。
- D2: - INT2:FIFO 内の変換結果の数が、D15 - D11 に指定した数に等しくなると割込みを生成する。
- D3: - INT3:自動ゼロ・サイクルが完了すると割込みを生成する。
- D4: - INT4:フル・キャリブレーション・サイクルが完了すると割込みを生成する。
- D5: - INT5:停止条件が検出されると割込みを生成する。
- D6: - INT6:電源電圧低下が検出されると割込みを生成する。
- D7: - INT7:チップがスタンバイ状態からアクティブ状態に復帰すると、割込みを生成する。
- D10 - D8: - 実行段階になった時点で割込みを生成する命令の番号を指定する。
- D15 - D11: - FIFO 内の変換結果の数を指定する。これと同じ数の変換結果が格納されると割込みを生成する。

プログラマーズ・ノート

割込イネーブルレジスタ:アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

割込イネーブルレジスタ:アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

タイマ・レジスタ(読込み/書込み)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
N = Timer Preset Value															

命令の Timer ビットがセットされていると、その命令の実行を遅らせる。

遅延時間(クロック・サイクル数)は次式で与えられる:

遅延 = $32 \times N + 2$ [クロック・サイクル]

プログラマーズ・ノート

タイマレジスタ:アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

タイマレジスタ:アドレス: シンボル名

Note:

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

16 進値

付録A:レジスタのビット配置とプログラマーズ・ノート

FIFO レジスタ (リード・オンリ)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Instruction Number or Extended Sign				Sign	Conversion Result										

D11 - D0: - 変換結果。12 ビット + サイン・ビットの場合は、12 ビット値。8 ビット + サイン・ビットの場合は、D11 - D4 が変換結果、D3 - D0 が 1110。

D12: - Sign: 変換結果のサイン・ビット。0 = 正、1 = 負。

D15 - D13: - 変換結果に対応する命令の番号、または、2 の補数形式の拡張サイン・ビット。どちらを意味するかは、構成・レジスタのビット D5 (Chan Mask) で選択される。

付録A:レジスタのビット配置とプログラマーズ・ノート

FIFO レジスタ: アドレス:

シンボル名

Note:

割込ステータス・レジスタ (リード・オンリ)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Number of results in FIFO					Instruction Number Being executed			INST7	INST6	INST5	INST4	INST3	INST2	INST1	INST0

ビット # 0 ~ 7 は割込みフラグ (ベクトル) で、対応する条件が発生すると 1 にセットされます。このセットは、その条件が割込みイネーブル・レジスタによってイネーブルされているいかにかわらず、常に行われます。レジスタが読出されるか、コンフィギュレーション・レジスタを介したデバイス・リセットがなされると、ビットは 0 にリセットされます。

D0: - INT0: ウォッチドッグ・モードでリミットを越えると 1 にセットされます。

D1: - INT1: D10 - D8 に指定した命令が実行段階になった時点で 1 にセットされます。

D2: - INT2: FIFO 内の変換結果の数が、D15 - D11 に指定した数に等しくなると 1 にセットされます。

D3: - INT3: 自動ゼロ・サイクルが完了すると 1 にセットされます。

D4: - INT4: フル・キャリブレーション・サイクルが完了すると 1 にセットされます。

D5: - INT5: ポーズ条件が検出されると 1 にセットされます。

D6: - INT6: 電源電圧低下が検出されると 1 にセットされます。

D7: - INT7: チップがスタンバイ状態からアクティブ状態に復帰すると、1 にセットされます。

D10 - D8: - 実行中の命令番号またはポーズもしくはタイム遅延中に実行される命令番号を保持。

D15 - D11: - デバイスの動作中、FIFO に現在格納されている変換結果の数を保持。

プログラマーズ・ノート:

割込みステータス・レジスタ: アドレス:

シンボル名

Note:

付録 A: レジスタのビット配置とプログラマーズ・ノート

リミット・ステータス・レジスタ (リード・オンリ)
(表)

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Limits #2: Status								Limits #1: Status							

以下のビットで一連のリミット・フラグ (ベクトル) を形成しており、各ビットはそれに対応するリミットが逸脱されると 1 にセットされます。各ビットは、命令のリミットと以下のように対応しています:

D0:	- 命令 # 0 のリミット # 1 を逸脱
D1:	- 命令 # 1 のリミット # 1 を逸脱
D2:	- 命令 # 2 のリミット # 1 を逸脱
D3:	- 命令 # 3 のリミット # 1 を逸脱
D4:	- 命令 # 4 のリミット # 1 を逸脱
D5:	- 命令 # 5 のリミット # 1 を逸脱
D6:	- 命令 # 6 のリミット # 1 を逸脱
D7:	- 命令 # 7 のリミット # 1 を逸脱
D8:	- 命令 # 0 のリミット # 2 を逸脱
D9:	- 命令 # 1 のリミット # 2 を逸脱
D10:	- 命令 # 2 のリミット # 2 を逸脱
D11:	- 命令 # 3 のリミット # 2 を逸脱
D12:	- 命令 # 4 のリミット # 2 を逸脱
D13:	- 命令 # 5 のリミット # 2 を逸脱
D14:	- 命令 # 6 のリミット # 2 を逸脱
D15:	- 命令 # 7 のリミット # 2 を逸脱

プログラマーズ・ノート:

リミット・ステータス・レジスタ: アドレス: シンボル名
Note:

生命維持装置への使用について

弊社の製品はナショナル セミコンダクター社の書面による許可なくしては、生命維持用の装置またはシステム内の重要な部品として使用することはできません。

1. 生命維持用の装置またはシステムとは (a) 体内に外科的に使用されることを意図されたもの、または (b) 生命を維持あるいは支持するものをいい、ラベルにより表示される使用法に従って適切に使用された場合に、これの不具合が使用者に身体的障害を与えると予想されるものをいいます。
2. 重要な部品とは、生命維持にかかわる装置またはシステム内のすべての部品をいい、これの不具合が生命維持用の装置またはシステムの不具合の原因となりそれらの安全性や機能に影響を及ぼすことが予想されるものをいいます。

ナショナル セミコンダクター ジャパン株式会社

本 社 / 〒135-0042 東京都江東区木場 2-17-16 TEL: (03) 5639-7300 <http://www.nsjk.co.jp/>

製品に関するお問い合わせはカスタマ・レスポンス・センタのフリーダイヤルまでご連絡ください。



0120-666-116



この紙は再生紙を使用しています

ご注意

日本テキサス・インスツルメンツ株式会社（以下TIJといいます）及びTexas Instruments Incorporated（TIJの親会社、以下TIJないしTexas Instruments Incorporatedを総称してTIといいます）は、その製品及びサービスを任意に修正し、改善、改良、その他の変更をし、もしくは製品の製造中止またはサービスの提供を中止する権利を留保します。従いまして、お客様は、発注される前に、関連する最新の情報を取得して頂き、その情報が現在有効かつ完全なものであるかどうかご確認下さい。全ての製品は、お客様とTIJとの間に取り引契約が締結されている場合は、当該契約条件に基づき、また当該取引契約が締結されていない場合は、ご注文の受諾の際に提示されるTIJの標準販売契約約款に従って販売されます。

TIは、そのハードウェア製品が、TIの標準保証条件に従い販売時の仕様に対応した性能を有していること、またはお客様とTIJとの間で合意された保証条件に従い合意された仕様に対応した性能を有していることを保証します。検査およびその他の品質管理技法は、TIが当該保証を支援するのに必要とみなす範囲で行なわれております。各デバイスの全てのパラメーターに関する固有の検査は、政府がそれ等の実行を義務づけている場合を除き、必ずしも行なわれておりません。

TIは、製品のアプリケーションに関する支援もしくはお客様の製品の設計について責任を負うことはありません。TI製部品を使用しているお客様の製品及びそのアプリケーションについての責任はお客様にあります。TI製部品を使用したお客様の製品及びアプリケーションについて想定されうる危険を最小のものとするため、適切な設計上および操作上の安全対策は、必ずお客様にてお取り下さい。

TIは、TIの製品もしくはサービスが使用されている組み合わせ、機械装置、もしくは方法に関連しているTIの特許権、著作権、回路配置利用権、その他のTIの知的財産権に基づいて何らかのライセンスを許諾するということは明示的にも黙示的にも保証も表明もしておりません。TIが第三者の製品もしくはサービスについて情報を提供することは、TIが当該製品もしくはサービスを使用することについてライセンスを与えるとか、保証もしくは是認するということを意味しません。そのような情報を使用するには第三者の特許その他の知的財産権に基づき当該第三者からライセンスを得なければならない場合もあり、またTIの特許その他の知的財産権に基づきTIからライセンスを得て頂かなければならない場合もあります。

TIのデータ・ブックもしくはデータ・シートの中にある情報を複製することは、その情報に一切の変更を加えること無く、かつその情報と結び付けられた全ての保証、条件、制限及び通知と共に複製がなされる限りにおいて許されるものとします。当該情報に変更を加えて複製することは不正で誤認を生じさせる行為です。TIは、そのような変更された情報や複製については何の義務も責任も負いません。

TIの製品もしくはサービスについてTIにより示された数値、特性、条件その他のパラメーターと異なる、あるいは、それを超えてなされた説明で当該TI製品もしくはサービスを再販売することは、当該TI製品もしくはサービスに対する全ての明示的保証、及び何らかの黙示的保証を無効にし、かつ不正で誤認を生じさせる行為です。TIは、そのような説明については何の義務も責任もありません。

TIは、TIの製品が、安全でないことが致命的となる用途ないしアプリケーション（例えば、生命維持装置のように、TI製品に不良があった場合に、その不良により相当な確率で死傷等の重篤な事故が発生するようなもの）に使用されることを認めておりません。但し、お客様とTIの双方の権限有る役員が書面でそのような使用について明確に合意した場合は除きます。たとえTIがアプリケーションに関連した情報やサポートを提供したとしても、お客様は、そのようなアプリケーションの安全面及び規制面から見た諸問題を解決するために必要とされる専門的知識及び技術を持ち、かつ、お客様の製品について、またTI製品をそのような安全でないことが致命的となる用途に使用することについて、お客様が全ての法的責任、規制を遵守する責任、及び安全に関する要求事項を満足させる責任を負っていることを認め、かつそのことに同意します。さらに、もし万一、TIの製品がそのような安全でないことが致命的となる用途に使用されたことによって損害が発生し、TIないしその代表者がその損害を賠償した場合は、お客様がTIないしその代表者にその全額の補償をするものとします。

TI製品は、軍事的用途もしくは宇宙航空アプリケーションないし軍事的環境、航空宇宙環境にて使用されるようには設計もされていませんし、使用されることを意図されていません。但し、当該TI製品が、軍需対応グレード品、若しくは「強化プラスチック」製品としてTIが特別に指定した製品である場合は除きます。TIが軍需対応グレード品として指定した製品のみが軍需品の仕様書に合致いたします。お客様は、TIが軍需対応グレード品として指定していない製品を、軍事的用途もしくは軍事的環境下で使用することは、もっぱらお客様の危険負担においてなされるということ、及び、お客様がもっぱら責任をもって、そのような使用に関して必要とされる全ての法的要求事項及び規制上の要求事項を満足させなければならないことを認め、かつ同意します。

TI製品は、自動車用アプリケーションないし自動車の環境において使用されるようには設計されていませんし、また使用されることを意図されていません。但し、TIがISO/TS 16949の要求事項を満たしていると特別に指定したTI製品は除きます。お客様は、お客様が当該TI指定品以外のTI製品を自動車用アプリケーションに使用しても、TIは当該要求事項を満たしていなかったことについて、いかなる責任も負わないことを認め、かつ同意します。

Copyright © 2011, Texas Instruments Incorporated
日本語版 日本テキサス・インスツルメンツ株式会社

弊社半導体製品の取り扱い・保管について

半導体製品は、取り扱い、保管・輸送環境、基板実装条件によっては、お客様での実装前後に破壊/劣化、または故障を起こすことがあります。

弊社半導体製品のお取り扱い、ご使用にあたっては下記の点を遵守して下さい。

1. 静電気

- 素手で半導体製品単体を触らないこと。どうしても触る必要がある場合は、リストストラップ等で人体からアースをとり、導電性手袋等をして取り扱うこと。
- 弊社出荷梱包単位（外装から取り出された内装及び個装）又は製品単品で取り扱いを行う場合は、接地された導電性のテーブル上で（導電性マットにアースをとったもの等）、アースをした作業者が行うこと。また、コンテナ等も、導電性のものを使うこと。
- マウンタやはんだ付け設備等、半導体の実装に関わる全ての装置類は、静電気の帯電を防止する措置を施すこと。
- 前記のリストストラップ・導電性手袋・テーブル表面及び実装装置類の接地等の静電気帯電防止措置は、常に管理されその機能が確認されていること。

2. 温・湿度環境

- 温度：0～40℃、相対湿度：40～85%で保管・輸送及び取り扱いを行うこと。（但し、結露しないこと。）

- 直射日光が当たる状態で保管・輸送しないこと。

3. 防湿梱包

- 防湿梱包品は、開封後は個別推奨保管環境及び期間に従い基板実装すること。

4. 機械的衝撃

- 梱包品（外装、内装、個装）及び製品単品を落下させたり、衝撃を与えないこと。

5. 熱衝撃

- はんだ付け時は、最低限260℃以上の高温状態に、10秒以上さらさないこと。（個別推奨条件がある時はそれに従うこと。）

6. 汚染

- はんだ付け性を損なう、又はアルミ配線腐食の原因となるような汚染物質（硫黄、塩素等ハロゲン）のある環境で保管・輸送しないこと。
- はんだ付け後は十分にフラックスの洗浄を行うこと。（不純物含有率が一定以下に保証された無洗浄タイプのフラックスは除く。）

以上