

## ステータス体形

---

- ・ 概要—ソース・メータのステータス体系の、動作から見た概要を説明します。
- ・ レジスタと待ち行列のクリア—レジスタと待ち行列をクリア (リセット) するアクションを説明します。
- ・ レジスタのプログラミングと読み取り—イネーブルレジスタをプログラムする方法とステータス体系の中のレジスタを読み取る方法を説明します。
- ・ ステータスバイトとサービスリクエスト (SRQ)—サービスリクエスト (SRQ) を発生するようにステータスバイトをプログラムする方法を説明します。シリアルポールシーケンスを使って SRQ を検出する方法を示します。
- ・ ステータスレジスタセット—4種類のステータスレジスタセット、すなわち標準イベントステータス、動作イベントステータス、測定イベントステータス、および疑問イベントステータスについて、ビット識別とコマンドに関する情報を用意します。
- ・ 待ち行列—出力待ち行列とエラー待ち行列について、詳細とコマンドに関する情報を用意します。

## 概要

ソース・メータは一連のステータスレジスタと待ち行列を用意しており、操作員は各種の計測器イベントを監視、操作することができます。ステータス体系は図 15-1 に示すとおりです。ステータス体系の中心は、ステータスバイトレジスタです。このレジスタは、ユーザの試験プログラムを使って読み取ることができ、これによってサービスリクエスト (SRQ) が発生したかどうか、そしてどのようなイベントがそのリクエストを発生させたかを明らかにすることができます。

### ステータスバイトと SRQ

ステータスバイトレジスタは、4 個のステータスレジスタセットと 2 個の待ち行列のサマリビットを受信します。これらのレジスタセットと待ち行列は、各種の計測器イベントを監視します。使用可能になったイベントが発生すると、このイベントは、ステータスバイトレジスタの中のサマリビットを 1 に設定します。ステータスバイトのサマリビットが 1 に設定され、対応するイネーブルビットが 1 に設定されると (ユーザがプログラムした形で)、RQS/MSS ビットは SRQ の発生を知らせるように設定されます。

### ステータスレジスタセット

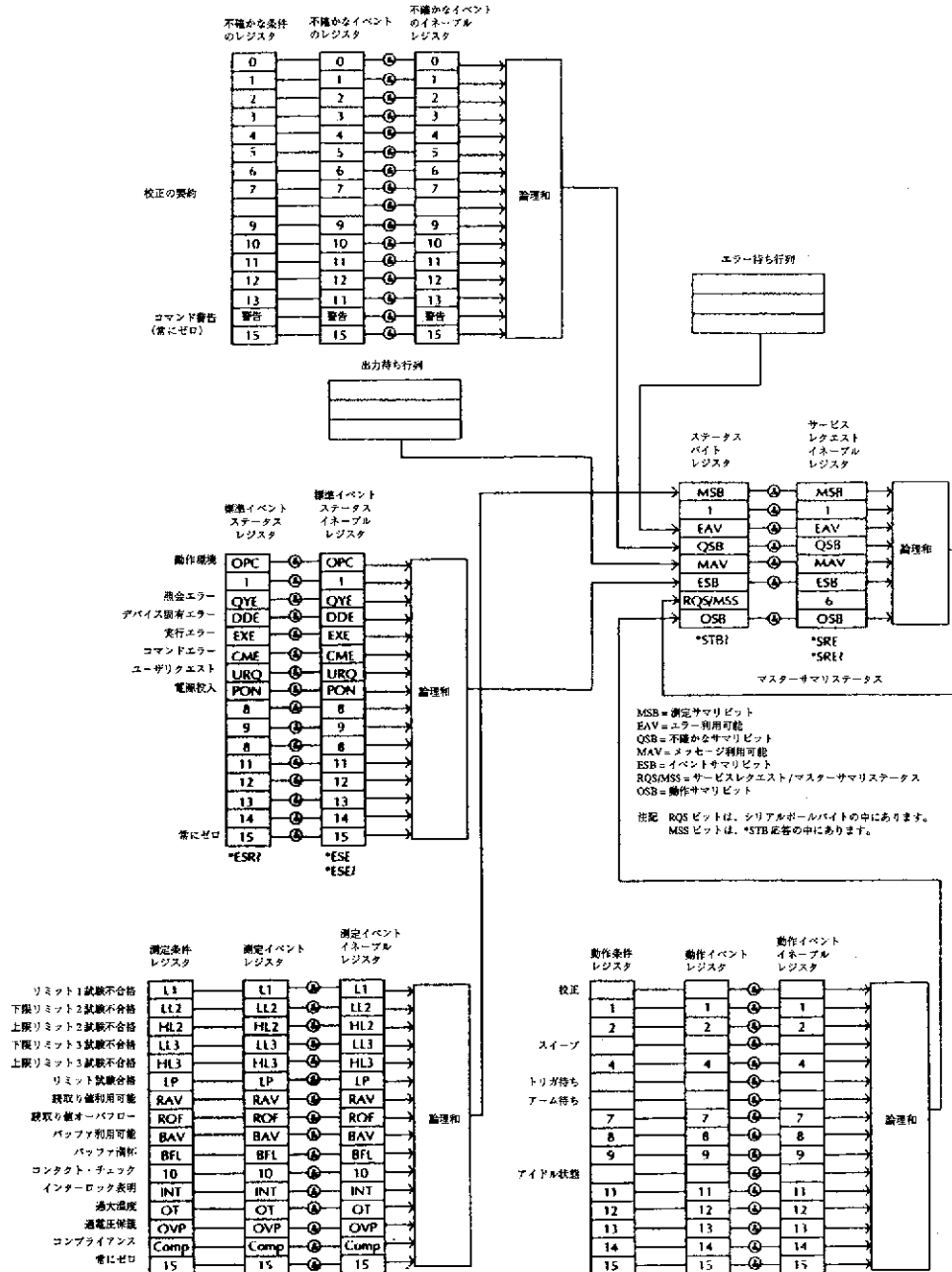
代表的なステータスレジスタセットは、条件レジスタ、イベントレジスタ、イベントイネーブルレジスタで構成されます。条件レジスタは、読み取り専用レジスタであり、常時アップデートを行い、計測器の現在の動作条件を反映します。

1 つのイベントが発生すると、当該のイベントレジスタビットは 1 に設定されます。このビットはレジスタがリセットされるまで、1 にラッチされた状態に留まります。イベントレジスタビットが 1 に設定され、対応するイネーブルビットが 1 に設定されると (ユーザがプログラムした形で)、レジスタの出力 (サマリ) は 1 に設定され、その結果、ステータスバイトレジスタのサマリビットは 1 に設定されます。

### 待ち行列

ソース・メータは 1 つの出力待ち行列と、1 つのエラー待ち行列を使用します。照会コマンドに対する応答メッセージは、出力待ち行列に置かれます。いろいろなプログラミングエラーとステータスメッセージは、発生の都度、エラー待ち行列に置かれます。待ち行列にデータが含まれる場合は、この待ち行列は、ステータスバイトレジスタの関連サマリビットを 1 に設定します。

図 15-1  
ソース・メータの  
ステータスレジスタ体系



## レジスタと待ち行列をクリアする

ソース・メータをオン状態にすると、ステータス体系中のすべてのレジスタのビットがクリア(リセット)され、2個の待ち行列は空になります。イベントレジスタとイベントイネーブルレジスタをリセットするコマンドと、エラー待ち行列を表 15-1 の一覧表に示します。これらのコマンドだけではなく、どのイネーブルレジスタでも、レジスタをプログラムする個別のコマンドとともに 0 パラメータ値を送ることにより、リセットすることができます。

**注記**     *SYSTEM:PRESet* と *\*RTS* は、ステータス体系レジスタと待ち行列には影響を与えません。

表 15-1

レジスタをリセットし待ち行列をクリアするための一般コマンドと SCPI コマンド

コマンド	内容	参照
レジスタをリセットする場合 <i>*CLS</i>	下記のイベントレジスタの、すべてのビットを 0 にリセットしてください。 標準イベントレジスタ 動作イベントレジスタ 測定イベントレジスタ 疑問イベントレジスタ	注記 1
<i>:STATus:PRESet</i>	下記のイベントレジスタの、すべてのビットを 0 にリセットしてください。 動作イベントイネーブルレジスタ 測定イベントイネーブルレジスタ 疑問イベントイネーブルレジスタ	注記 1
エラー待ち行列をクリアする場合 <i>*CLS</i>	エラー待ち行列からのすべてのメッセージをクリアしてください。	注記 2
<i>:STATus:QUEue:CLEar</i>	エラー待ち行列からのメッセージをクリアしてください。	注記 3
<i>:SYSTEM:ERRor:CLEar</i>	エラー待ち行列からのメッセージをクリアしてください。	注記 3

注記：

1. 標準イベントイネーブルレジスタは、*STATus:PRESet* または *\*CLS* を使ってもクリアされません。0 パラメータ値を *\*ESE* とともに送って、当該のイネーブルレジスタのすべてのビットを 0 にリセットしてください。([ステータスバイトとサービスリクエストのコマンド] 参照)
2. *STATus:PRESet* は、エラー待ち行列に影響を与えません。
3. エラー待ち行列をクリアする 2 個のクリアコマンドのどちらかを使ってください。

## レジスタをプログラムし、読み取る

### イネーブルレジスタをプログラムする

ユーザによるプログラムが可能なただ一つのレジスタは、イネーブルレジスタです。ステータス体系中のほかのレジスタは、すべて読み取り専用レジスタです。以下で説明するのは、イネーブルレジスタのプログラミングに使用する各種のコマンドのパラメータ値を確認する方法です。実際のコマンドは、この部の後半で説明します(表 15-3 と 15-6 を参照)。

イベントイネーブルレジスタをプログラムするコマンドを送るときには、該当するレジスタの中のそれぞれのビットの所要の状態(1 または 0) を決定する値もいっしょに送出されます。イネーブルレジスタをプログラムするときには、パラメータ値のフォーマットとして 2 進、10 進、16 進、または 8 進のどれかを使用します。

レジスタのビット位置(図 15-2 参照) は、2 進パラメータ値を示します。たとえば、ビット B4、B3、B1 を 1 に設定しようとする、2 進値は 11010 (ここで B4=1、B3=1、B1=1、ほかのビットはすべて 0) です。ほかのフォーマットのどれかを使う場合は、2 進数を等価の 10 進数、16 進数、または 8 進数に変換してください。

2 進数 11010 = 10 進数 26 = 16 進数 1A = 8 進数 32

図 15-2 はそれぞれのレジスタビットについての 10 進加重を含むことに留意してください。

ビット B4、B3、B1 を 1 に設定すると、10 進パラメータ値は、これらのビットに対する 10 進加重値の合計となるでしょう ( $16 + 8 + 2 = 26$ )。

図 15-2

16 ビットステータスレジスタ

A) ビット 0 から 7 まで

ビット位置	B7	B6	B5	B4	B3	B2	B1	B0
2 進値	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
10 進値	128	64	32	16	8	4	2	1
加重	(2 <sup>7</sup> )	(2 <sup>6</sup> )	(2 <sup>5</sup> )	(2 <sup>4</sup> )	(2 <sup>3</sup> )	(2 <sup>2</sup> )	(2 <sup>1</sup> )	(2 <sup>0</sup> )

B) ビット 8 から 15 まで

ビット位置	B15	B14	B13	B12	B11	B10	B9	B8
2 進値	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
10 進値	32768	16384	8192	4096	2048	1024	512	256
加重	(2 <sup>15</sup> )	(2 <sup>14</sup> )	(2 <sup>13</sup> )	(2 <sup>12</sup> )	(2 <sup>11</sup> )	(2 <sup>10</sup> )	(2 <sup>9</sup> )	(2 <sup>8</sup> )

<NDN>(非 10 進値数) パラメータタイプを使い、非 10 進値を送信します。これらの値にはヘッダ(#B, #H または #Q) が必要で、これによって送信中のデータフォーマットを識別します。ヘッダの中の文字は、大文字でも小文字でも使えます。<NRf>(数値表現フォーマット) パラメータタイプを使い、10 進値を送信します。<NRf> はヘッダを使用しません。次の例は、ビット B5、B3、B2 を 1 に設定する場合の、適切なパラメータシンタクスを示します。

#b101100	2 進フォーマット (<NDN> パラメータタイプ)
#h2C	16 進フォーマット (<NDN> パラメータタイプ)
#q54	8 進フォーマット (<NDN> パラメータタイプ)
44	10 進フォーマット (<NRf> パラメータタイプ)

非 10 進パラメータ値に有効な文字は次のとおりです。

<NDN> フォーマット	有効な文字
2 進	1 と 0
16 進	0 から 9 までと A から F まで
8 進	0 から 7 まで

## レジスタを読み取る

ステータス体系中のレジスタは、関連する照会 (?) コマンドを使用して読み取ることができます。以下では、戻された値 (応答メッセージ) を解釈する方法を説明します。実際の照会コマンドについては、この部の後半で説明します (表 15-3、表 15-4、表 15-7 参照)。

応答メッセージとは、レジスタの中のどのビットが 1 に設定されているかを示す値です。この値 (すでに 2 進化されてない場合は) を 2 進等価値に変換する必要があります。たとえば、100101 という 2 進値の場合は、ビット B5、B3、B0 が 1 に設定されています。

戻される値のフォーマットは、2 進、10 進、16 進、8 進のどれでもかまいません。FORMat: SREGister コマンドを使い、戻された値のためのデータフォーマットを選択します (表 15-2 参照)。

非 10 進フォーマットの場合は、下記のヘッダの 1 つが戻された値に付随しており、どのフォーマットが選択されたかを示します。

#B = 2 進値のヘッダ  
 #H = 16 進値のヘッダ  
 #Q = 8 進値のヘッダ

表 15-2

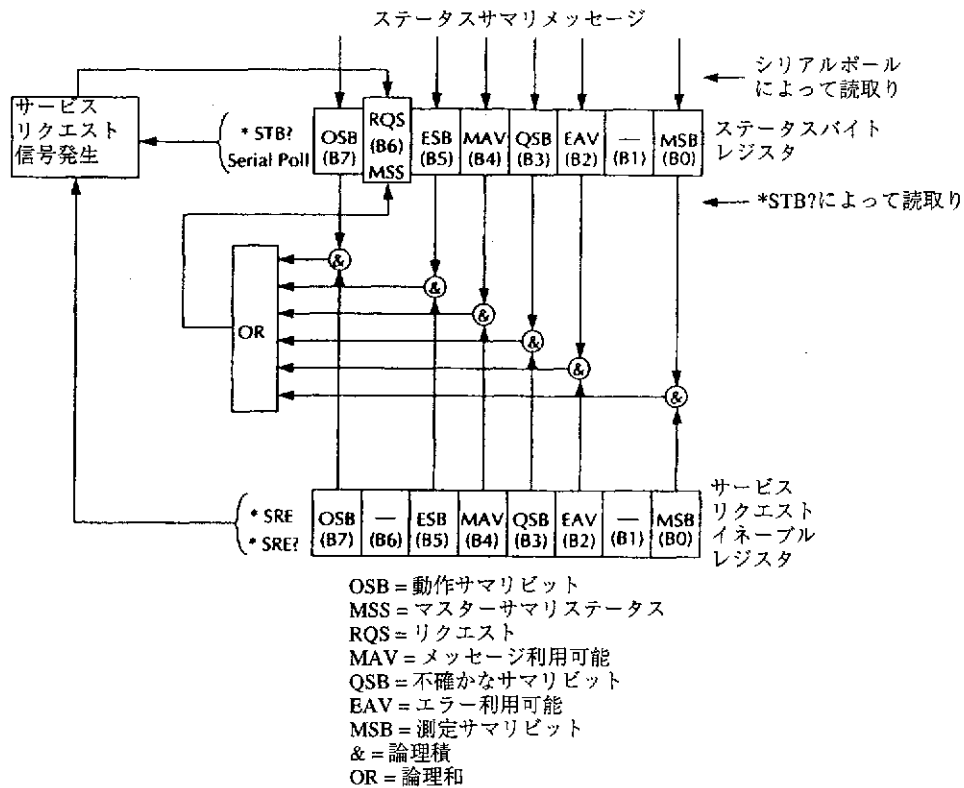
ステータスレジスタ読み  
 取り用データフォーマットコマンド

コマンド	内容	デフォルト
:FORMat:SREGister <name>	ステータスレジスタを読み取るためのデータフォーマットを選択してください。 <name> =    ASCii            10 進フォーマット HEXadecimal    16 進フォーマット OCTal            8 進フォーマット BINary            2 進フォーマット	ASCii

## ステータスバイトとサービスリクエスト(SRQ)

サービスリクエストを制御するのは、ステータスバイトレジスタとサービスリクエストイネーブルレジスタという、2個の8ビットレジスタです。図 4-9 は、これらのレジスタの体系を示します。

図 15-3  
ステータスバイトとサービスリクエスト(SRQ)



## ステータスバイトレジスタ

ステータスバイトレジスタステータスレジスタと待ち行列を使用して、ステータスバイトレジスタの関連ビット (B0、B2、B3、B4、B5、B7) を設定またはクリアします。これらのビットはラッチしません。そしてその状態 (0 または 1) を決めるのは、サマリメッセージ (0 または 1) だけです。たとえば、標準イベントステータスレジスタを読み取ると、そのレジスタはクリアされます。その結果、サマリメッセージは 0 にリセットされ、これがステータスバイトレジスタ中の ESB ビットをクリアします。

ステータスバイトレジスタの各ビットの概要は、次のとおりです。

- ・ ビット B0、測定サマリビット (MSB) – 1 に設定されたサマリビットは、使用可能になった測定イベントが発生したことを示します。
- ・ ビット B1、未使用
- ・ ビット B2、エラーアベイラブル (EAV) – 1 に設定されたサマリビットは、エラーまたはステータスメッセージがエラー待ち行列の中に存在することを示します。
- ・ ビット B3、疑問サマリビット (QSB) – 1 に設定されたサマリビットは、使用可能になった疑問イベントが発生したことを示します。
- ・ ビット B4、メッセージアベイラブル (MAV) – 1 に設定されたサマリビットは、応答メッセージが出力待ち行列の中に存在することを示します。
- ・ ビット B5、イベントサマリビット (ESB) – 1 に設定されたサマリビットは、使用可能になった標準イベントが発生したことを示します。
- ・ ビット B6、リクエストサービス (RQS)/ マスタサマリステータス (MSS) – 1 に設定されたビットは、ステータスバイトレジスタ中の使用可能になったサマリビットが 1 に設定されたことを示します。
- ・ ビット B7、動作サマリ (OSB) – 1 に設定されたサマリビットは、使用可能になった動作イベントが発生したことを示します。

使い方にもよりますが、ステータスバイトレジスタのビット B6 は、サービスのリクエスト (RQS) またはマスタサマリステータス (MSS) のどちらかです。

- ・ ソース・メータのシリアルポールシーケンスを使ってステータスバイト (a.k.a. シリアルポールバイト) を得ようとするときは、B6 が RQS ビットです。シリアルポールシーケンスの使用法の詳細は、「シリアルポーリングと SRQ」を参照してください。
- ・ ステータスバイトを読み取るために \*STB? コマンド (表 15-3 参照) を使用するときは、B6 は MSS ビットです。



## サービスリクエストイネーブルレジスタ

サービスリクエストの生成を制御するのは、サービスリクエストイネーブルレジスタです。このレジスタはお客様がプログラムしたものです。このレジスタを使うのは、ステータスバイトレジスタのステータスサマリメッセージビット (B0、B2、B3、B4、B5、および B7) によってビット B6 (RQS/MSS) の設定を使用可能にしたり使用禁止にする場合です。図 15-3 に示すように、サマリビットは、サービスリクエストイネーブルレジスタの対応イネーブルビットと、論理的に AND の関係にあります。1 に設定されたサマリビット (1) がイネーブルレジスタの使用可能 (1) ビットと AND 関係にあれば、論理 "1" の出力が OR ゲートに印加され、ステータスバイトレジスタの MSS/RQS ビットを 1 に設定します。

サービスリクエストイネーブルレジスタの一つ一つのビットは、\*SRE コモンコマンドを使って設定またはクリアすることができます。サービスリクエストイネーブルレジスタを読み取るには、\*SRE? コマンドを使用してください。サービスリクエストイネーブルレジスタがクリアされるのは、電源再投入を行う場合か、ZERO のパラメータ値を \*SRE (すなわち \*SRE 0) とともに送る場合です。SRQ イネーブルレジスタをプログラムし読み取るためのコマンドを、表 15-3 に一覧形式で示します。

## シリアルポーリングと SRQ

0 から 1 に移行する、使用可能状態のイベントサマリビットであればどのビットでも、B6 を設定し、サービスリクエスト (SRQ) 信号を発生します。ユーザの試験プログラムの中で、ユーザは定期的にステータスバイトレジスタを読み取り、サービスリクエストが発生しているかどうか、発生していれば何が原因になっているかを、チェックすることができます。SRQ が発生した場合、プログラムは、たとえば、リクエストに対応する適切なサブルーチンに分岐することができます。

通常は、サービスリクエストは、2400 型のシリアルポールシーケンスによって管理されます。SRQ が発生しない場合は、ステータスバイトレジスタのビット B6 (RQS) はクリアされたままで、シリアルポールが行われたあと、プログラムは正常に進行するだけとなります。SRQ が発生すれば、ステータスバイトレジスタのビット B6 が設定され、シリアルポールが SRQ を検出すれば、プログラムはサービスルーチンに分岐することができます。

シリアルポールは、ステータスバイトレジスタの RQS を自動的にリセットします。これによって、これ以後のシリアルポールは、ほかの種類のイベントによって SRQ が発生していないかをモニタすることが可能になります。シリアルポールのあと、最初の SRQ を発生したイベントレジスタがクリアされていなくても、同じイベントが別の SRQ を発生することがあります。

シリアルポールは、RQS をクリアしますが、MSS をクリアしません。MSS ビットは、すべてのステータスバイトイベントサマリビットがクリアされるまで、残ります。

## SPE、SPD (シリアルポーリング)

SPE と SPD の汎用バスコマンドシーケンスを使い、ソース・メータのシリアルポーリングを行います。シリアルポーリングによって、シリアルポールバイト (ステータスバイト) が得られます。代表的な例として、コントローラがシリアルポーリングを使い、これによっていくつかの計測器のうち、どれが SRQ ラインにサービスをリクエストしたかを確認します。



## ステータスレジスタセット

図 15-1 に示すように、ソース・メータのステータス体系には、標準イベントステータス、動作イベントステータス、測定イベントステータス、疑問イベントステータスという 4 つのステータスレジスタがあります。

### レジスタビットの説明

#### 標準イベントレジスタ

標準イベントレジスタで使用するビット (図 15-4 に示す) の内容は、次のとおりです。

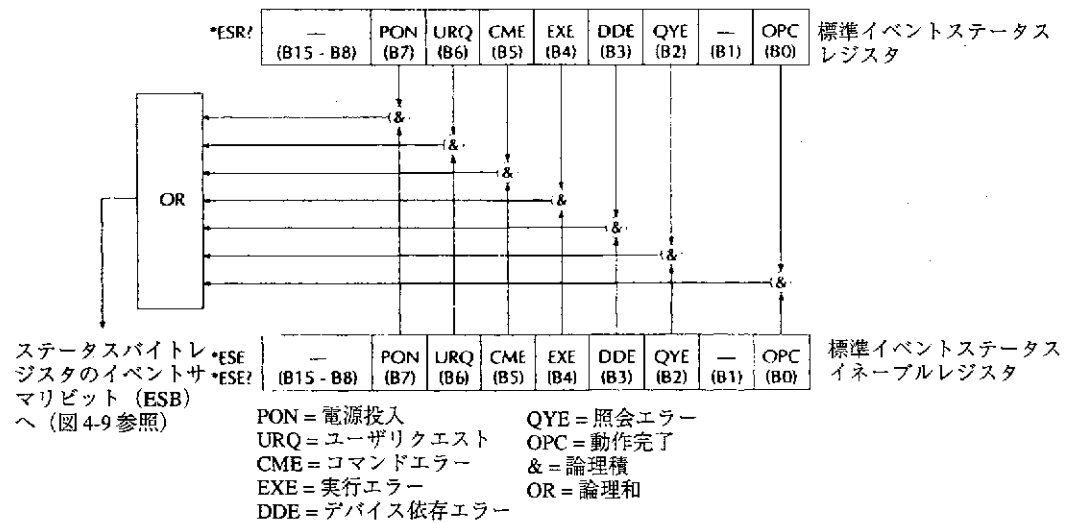
- ・ ビット B0、動作完了—1 に設定されたビットは、進行中の選択されたデバイス動作がすべて完了し、ソース・メータは新しいコマンドを受け入れる用意ができていることを示します。このビットが 1 に設定されるのは、\*OPC? 照会コマンドに応答する場合だけです。\*OPC と \*OPC? の詳細は、第 16 部を参照してください。
- ・ ビット B1—未使用
- ・ ビット B2、照会エラー (QYE)—1 に設定されたビットは、ユーザが空の出力待ち行列からデータを読み取ろうとしたことを示します。
- ・ ビット B3、デバイス依存エラー (DDE)—1 に設定されたビットは、内部条件が原因となって、計測器の動作が正しく行われなかったことを示します。
- ・ ビット B4、実行エラー (EXE)—1 に設定されたビットは、コマンドの実行を試みている間にソース・メータがエラーを検出したことを示します。
- ・ ビット B5、コマンドエラー (CME)—1 に設定されたビットは、コマンドエラーが発生したことを示します。

コマンドエラーは、次のエラーを含みます。

- ・ IEEE-488.2 シンタクスエラー—ソース・メータが受信したメッセージは、IEEE-488.2 規格で定義されるシンタクスに一致しません。
- ・ 意味論的エラー—ソース・メータが受信したコマンドは、スペルが誤っていました。またはソース・メータが受信したオプションの IEEE-488.2 コマンドは、実行されませんでした。
- ・ 機械はプログラム内でグループ実行トリガ (GET) を受け取りました。シリアルポートは、RQS をクリアしますが、MSS をクリアしません。MSS ビットは、すべてのステータスバイトイベントサマリビットがクリアされるまで、残ります。
- ・ ビット B6、ユーザリクエスト (URQ)—1 に設定されたビットは、ソース・メータ前面パネルの LOCAL キーが押されたことを示します。
- ・ ビット B7、パワーオン (PON)—1 に設定されたビットは、このレジスタの前の読み取り以来、ソース・メータがオフ状態にされ、またオン状態に戻されていることを示します。

図 15-4

標準イベントステータス

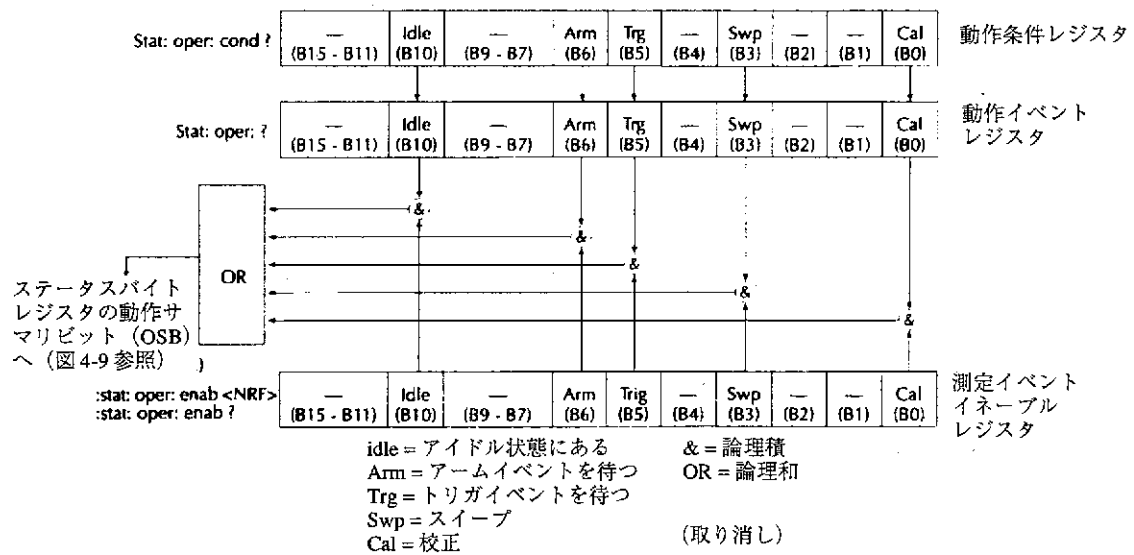


## 動作イベントレジスタ

動作イベントレジスタ (図 15-5 に示す) に使用するビットの内容は、次のとおりです。

- ・ ビット B0、校正 (Cal) - 1 に設定されたビットは、ソース・メータが校正中であることを示します。
- ・ ビット B1 と B2 - 未使用
- ・ ビット B3、スウィーピング (Swp) - 1 に設定されたビットは、計測器がスウィープ動作を行っていることを示します。
- ・ ビット B4 - 未使用
- ・ ビット B5、トリガイイベント待ち (Trig) - 1 に設定されたビットは、ソース・メータがトリガレイヤーにあり、TLINK イベントの発生を待っていることを示します。
- ・ ビット B6、アームイベント待ち (Arm) - 1 に設定されたビットは、ソース・メータがアームレイヤーにあり、アームイベントの発生を待っていることを示します。
- ・ ビット B7 と B9 - 未使用
- ・ ビット B10、アイドル状態 (Idle) - 1 に設定されたビットは、ソース・メータがアイドル状態にあることを示します。
- ・ ビット B11 から B15 - 未使用

図 15-5  
動作イベントレジスタ



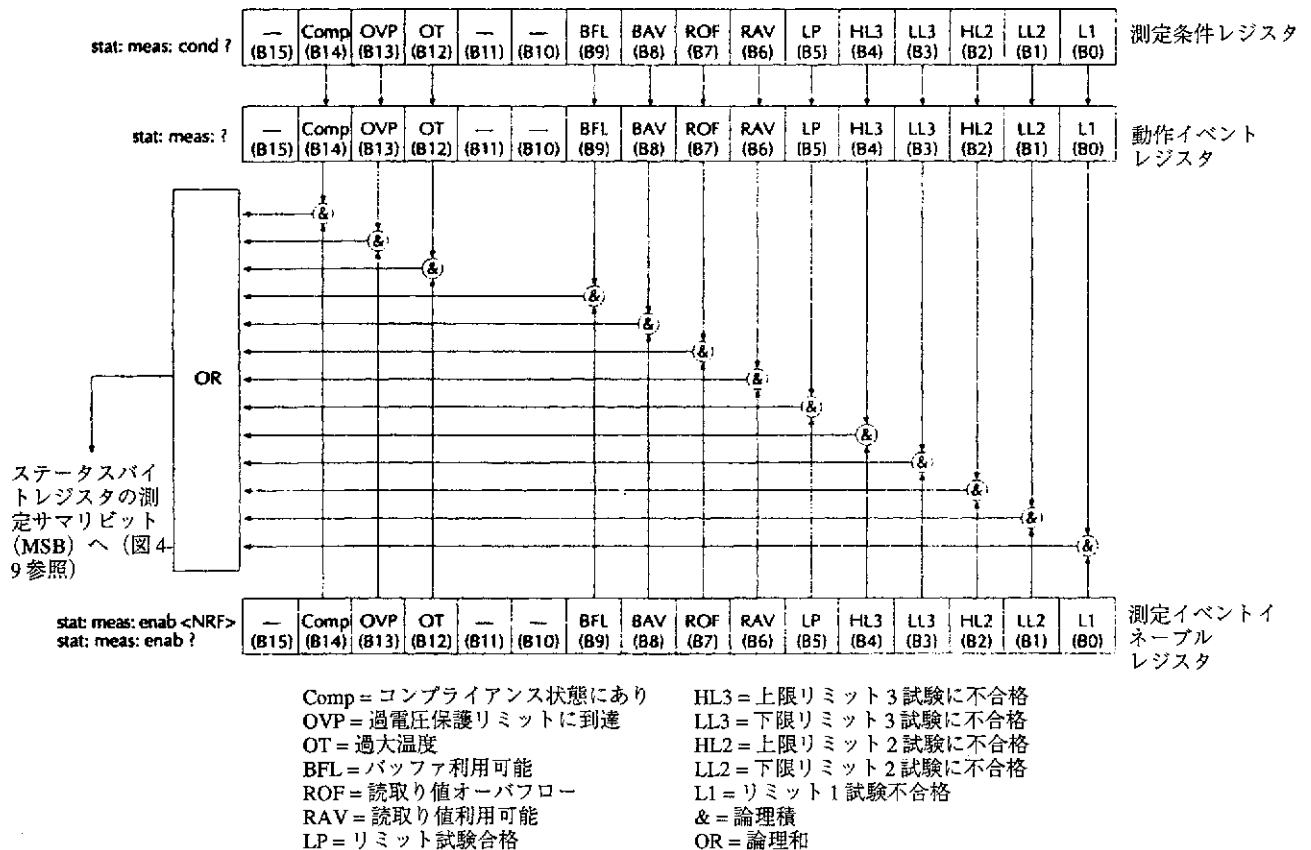
## 測定イベントレジスタ

測定イベントレジスタ (図 15-6 に示す) に使用するビットの内容は、次のとおりです。

- ビット B0、リミット 1 不合格 (L1) - 1 に設定されたビットは、リミット 1 の試験に不合格になったことを示します。
- ビット B1、ロウリミット 2 不合格 (LL2) - 1 に設定されたビットは、ロウリミット 2 の試験に不合格になったことを示します。
- ビット B2、ハイリミット 2 不合格 (HL2) - 1 に設定されたビットは、ハイリミット 2 の試験に不合格になったことを示します。
- ビット B3、ロウリミット 3 不合格 (LL3) - 1 に設定されたビットは、ロウリミット 3 の試験に不合格になったことを示します。
- ビット B4、ハイリミット 3 不合格 (HL3) - 1 に設定されたビットは、ハイリミット 3 の試験に不合格になったことを示します。
- ビット B5、リミット合格 (LP) - 1 に設定されたビットは、すべてのリミット試験に合格したことを示します。
- ビット B6、読み取りアベイラブル (RAV) - 1 に設定されたビットは、読み取りが行われ処理されたことを示します。
- ビット B7、読み取り値オーバーフロー (ROF) - 1 に設定されたビットは、電圧または電流の読み取り値が選択したソース・メータ測定レンジを越えることを示します。
- ビット B8、バッファアベイラブル (BAV) - 1 に設定されたビットは、少なくとも 2 個の読み取り値がバッファの中にあることを示します。
- ビット B9、バッファフル (BFL) - 1 に設定されたビットは、トレースバッファが満杯であることを示します。
- ビット 10 - リミット 4-コンタクトチェック (付録 F 参照)
- ビット 11、インタロックアサート (Int) - 1 に設定されたビットは、インタロックラインがデジタルロー (アサートされた) 状態にあることを示します。ソース出力をオン状態にすることができます。
- ビット B12、過大温度 (OT) - 1 に設定されたビットは、過大温度条件が存在することを示します。ソース出力をオン状態にすることはできません。

- ・ ビット B13、過電圧保護 (OVP) - 1 に設定されたビットは、ソースがプログラムされたレベルに制限されていることを示します。
- ・ ビット B14、コンプライアンス (Comp) - 1 に設定されたビットは、ソースがコンプライアンス状態にあることを示します。
- ・ ビット B15 - 未使用

図 15-6  
測定イベントステータス

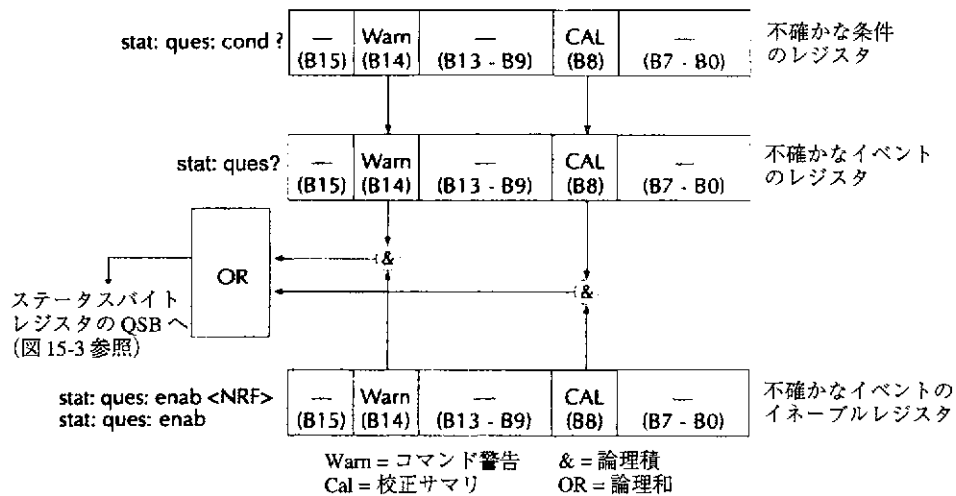


### 疑問イベントレジスタ

疑問イベントレジスタ (図 15-7 に示す) に使用するビットの内容は、次のとおりです。

- ・ ビット B0 から B7 - 未使用
- ・ ビット B8、校正サマリ (Cal) - 1 に設定されたビットは、無効な校正係数が電源投入シーケンスが進んでいる間に検出されたことを示します。このエラーは、計測器の校正が順調に終わったあと、クリアします。
- ・ ビット B9 から B13 - 未使用
- ・ ビット B14、コマンド警告 (Warn) - 1 に設定されたビットは、信号指向測定コマンドが無視されていることを示します。
- ・ ビット B15 - 未使用

図 15-7  
疑問イベントのステータス



## 条件レジスタ

図 15-1 に示すように、それぞれのステータスレジスタセット (標準イベントレジスタセットを除く) は条件レジスタを持っています。条件レジスタとは、リアルタイムの読み取り専用レジスタで、計測器の現在の動作条件を反映するために常時アップデートを行います。たとえば、ソース・メータがこの状態にあると、動作条件レジスタのビット B10 (アイドル) が 1 に設定されます。計測器がアイドル状態から外されると、ビット B10 はクリアします。

条件レジスタを読み取るためのコマンドを、表 15-5 に一覧形式で示します。レジスタ読み取りの詳細は、「レジスタを読み取る」を参照してください。

表 15-5  
条件レジスタコマンド

コマンド	内容
:STATus:OPERation:CONDition?	動作条件レジスタを読み取ってください。
:STATus:MEASurement:CONDition?	測定条件レジスタを読み取ってください。
:STATus:QUEStionable:CONDition?	疑問条件のレジスタを読み取ってください。

## イベントレジスタ

図 15-1 が示すように、それぞれのステータスレジスタセットは、イベントレジスタを持っています。あるイベントが発生すると、該当するイベントレジスタビットは1に設定されます。このビットはレジスタがリセットされるまで、1にラッチされた状態に留まります。イベントレジスタを読み取ると、このレジスタのビットはクリアされます。\*CLSは4個のレジスタすべてをリセットします。

イベントレジスタを読み取るためのコマンドを、表 15-6 に一覧形式で示します。レジスタ読み取りの詳細は、「レジスタを読み取る」を参照してください。

表 15-6  
イベントレジスタコマンド

コマンド	内容	デフォルト
*ESR?	標準イベントステータスレジスタを読み取ってください。	(注記)
:STATus:OPERation:[EVENT]?	動作イベントレジスタを読み取ってください。	
:STATus:MEASurement:[EVENT]?	測定イベントレジスタを読み取ってください。	
:STATus:QUESTionable:[EVENT]?	疑問イベントのレジスタを読み取ってください。	

注記：電源投入と \*CLS はすべてのイベントレジスタを0にリセットします。STATus:PRESet は影響を与えません。

## イベントイネーブルレジスタ

図 15-1 が示すように、それぞれのステータスレジスタセットはイネーブルレジスタを持っています。それぞれのイベントレジスタビットは、イネーブルレジスタの対応イネーブルビットと、論理的に AND (&) の関係にあります。したがって、あるイベントビットが1に設定され、対応するイネーブルビットも1に設定されると(ユーザがプログラム下方値で)、そのレジスタの出力(サマリ)も1に設定されます。その結果、ステータスバイトレジスタのサマリビットも1に設定されます。

イベントイネーブルレジスタを読み取るためのコマンドを、表 15-7 に一覧形式で示します。レジスタのプログラミングと読み取りの詳細は、「イネーブルレジスタをプログラムする」と「レジスタを読み取る」を参照してください。

注記 0 パラメータ値に関連するイネーブルコマンド(すなわち STATus:OPERation:ENABLE 0)とともに送ることにより、どのイネーブルレジスタのビットでも0にリセットすることができます。



表 15-7

イベントイネーブルレジスタ・コマンド

コマンド	内容	デフォルト	
*ESE <NDN> or <NRF>	標準イベントイネーブルレジスタを読み取ってください。 (「パラメータ」参照)	(注記)	
*ESE?	標準イベントイネーブルレジスタを読み取ってください。		
STATus	STATus サブシステム		
:OPERation	動作イベントイネーブルレジスタ		
:ENEBle <NDN> or N<Rf>	イネーブルレジスタをプログラムしてください。		
:ENEBle?	イネーブルレジスタを読み取ってください。		
:MEASurement	測定イベントイネーブルレジスタ		
:ENEBle <NDN> or N<Rf>	イネーブルレジスタをプログラムしてください。		
:ENEBle?	イネーブルレジスタを読み取ってください。		
:QUESTionable	疑問イベントイネーブルレジスタ		
:ENEBle <NDN> or N<Rf>	イネーブルレジスタをプログラムしてください。		
:ENEBle?	測定イベントイネーブルレジスタを読み取ってください。		
パラメータ:			
<NDN>	= #Bxx...x		2進フォーマット (それぞれの x=1 または 0)
	= #Hx	16進フォーマット (x=0 から FFFF)	
	= #Qx	8進フォーマット (x=0 から 1777777)	
<NRF>	= 0 から 65535	10進フォーマット	

注記：電源投入と STATus:PRESet は、すべてのイネーブルレジスタのすべてのビットを 0 にリセットします。\*CLS は影響を与えません。

### プログラミング例 — レジスタセットをプログラムし、読み取る

表 15-8 のコマンドシーケンスは、測定レジスタセットをプログラムし、読み取ります。レジスタは、(直接セットされているビットを示す) バイナリフォーマットを使用して、読み取られます。フォーマットを選択するためのコマンド (FORMat:SREGISTER) は、表 15-2 に示されています。

表 15-8

レジスタをプログラムし読み取るプログラミングの例

コマンド	内容
FROM:SREG BIN	2進フォーマットを選択して、レジスタを読み取ってください。
STAT:MEAS:ENAB 512	BFL を有効にしてください (バッファ満杯)。
STAT:MEAS:COND?	測定条件レジスタを読み取ってください。
STAT:MEAS?	測定イベントレジスタを読み取ってください。

## 待ち行列

2400 型は、2つの待ち行列を使用します。これらは、先入れ先出し（FIFO）レジスタです。

- ・ 出力待ち行列—読取り値と応答メッセージを保持するために使用
- ・ エラー待ち行列—エラーメッセージとステータスメッセージを保持するために使用

上記の2つの待ち行列が、ほかのレジスタとともに、どのような体系を設定するかを、2400 型のステータスモデル（図 4-4）に示します。

### 出力待ち行列

出力待ち行列は、計測器の正常動作に関係のあるデータを保持します。たとえば、照会コマンドが送られた場合、応答メッセージは出力待ち行列に入れられます。

データが出力待ち行列に入れられた場合には、ステータスバイトレジスタの中のメッセージ利用可能（MAV）ビットが設定されます。データメッセージは、読取りが完了すると出力待ち行列からクリアされます。出力待ち行列は、空になると、クリアされたと見なされます。空の出力待ち行列は、ステータスバイトレジスタの MAV をクリアします。

適切な照会を送ったあと、2400 型に対して、talk するようにアドレスされることにより、出力待ち行列からメッセージを読み取ってください。

### エラー待ち行列

エラー待ち行列は、エラーメッセージとステータスメッセージを保持します。エラーまたはステータスイベントが発生すると、エラー/ステータスを定義するメッセージがエラー待ち行列の中に置かれます。

メッセージがエラー待ち行列の中に置かれると、ステータスバイトレジスタのエラーアベイラブル（EAV）ビットが 1 に設定されます。エラー/ステータスメッセージは、読み取りが終わると、エラー待ち行列からクリアされます。エラー待ち行列は、空になるとクリアされたものと見なされます。空になったエラー待ち行列は、ステータスバイトレジスタ中の EAV をクリアします。

エラー待ち行列は、最大 10 個のエラー/ステータスメッセージを保持します。エラー待ち行列を読み取るためのコマンドを、表 15-9 に一覧形式で示します。エラー待ち行列中の、単一メッセージを読み取る時、「最も古い」メッセージが読み取られ、そのあと待ち行列から除去されます。待ち行列がフル状態になると、メッセージ "350"、すなわち「待ち行列オーバフロー」が最終記憶位置に入ります。電源投入すると、エラー待ち行列は空になります。空になると、メッセージ「0、No Error」が待ち行列に置かれます。

エラー待ち行列のメッセージの先頭には、コード番号が付きます。負 (-) の番号を使って SCPI で定義したメッセージを表し、正 (+) の番号を使ってケースレーが定義したメッセージを表します。これらのメッセージは、付録 B に一覧表として示してあります。表 15-7 に示すように、完全なメッセージ（コードとメッセージ）またはコードのみを読み取るためのコマンドがあります。

電源投入するとすべてのエラーメッセージが使用可能となり、メッセージは発生の都度、エラー待ち行列に入ります。表 15-9 に示すように、メッセージを使用可能と使用禁止、またはどちらかにするコマンドがあります。このようなコマンドには、<list> パラメータを使用し、使用可能または使用禁止にするメッセージを指定します。メッセージを指定するのは、それぞれのコードです。次の例は、<list> パラメータを使用する場合のいろいろな形を示します。

<list>	=(-110)	単一メッセージ
	=(-110:-222)	メッセージのレンジ (-110 から -222)
	=(-110:-222, -220)	レンジ入力と単一入力 (コンマで分離)

メッセージを使用可能にする場合、リストで指定されていないメッセージは使用禁止になります。メッセージを使用禁止にする場合、リストされたメッセージは、それぞれ、使用可能メッセージリストから除去されます。

注記 すべてのメッセージがエラー待ち行列に入るのを防ぐために、次のようにしてイネーブルコマンドを、ナルリストパラメータとともに送ってください。

**STATus:QUEue:ENABle**

表 15-9  
エラー待ち行列のコマンド

コマンド	内容	デフォルト
<b>STATus</b>	STATus サブシステム	
:QUEue	エラー待ち行列を読み取ってください。	
[:NEXT]?	最も古いエラー/ステータスを読み取り、クリアしてください (コードとメッセージ)。	(注記 1)
:ENABle <list>	エラー待ち行列に入るエラーメッセージとステータスメッセージを指定してください。	
:ENABle?	有効になったメッセージを読み取ってください。	(注記 2)
:DISABle <list>	待ち行列に入れないメッセージを指定してください。	
:DISABle?	無効になったメッセージを読み取ってください。	(注記 2)
:CLEar	エラー待ち行列から来たメッセージをクリアしてください。	
<b>SYSTem</b>	SYSTem サブシステム	
:ERRor	エラー待ち行列を読み取ってください。	(注記 1)
[:NEXT]?	最も古いエラー/ステータスを読み取り、クリアしてください (コードとメッセージ)。	
:ALL?	すべてのエラー/ステータスを読み取り、クリアしてください (コードとメッセージ)。	
:COUNt?	待ち行列にあるメッセージの数を読み取ってください。	
:CODE	コード番号のみ	
[:NEXT]?	最も古いエラー/ステータスを読み取り、クリアしてください (コードのみ)。	
:ALL	すべてのエラー/ステータスを読み取り、クリアしてください (コードのみ)。	
:CLEar	エラー待ち行列から来たメッセージをクリアしてください。	

注記：

1. 電源投入と \*CLS は、エラー待ち行列を空にします。STATus:PRESet は影響を与えません。
2. 電源投入はエラーメッセージを有効にし、ステータスメッセージを無効にします。\*CLS と STATus:PRESet は影響を与えません。

### プログラミングの例—エラー待ち行列を読み取る

次のコマンドは得アラー待ち行列を読み取ります。

**STAT:QUE?**