

KEITHLEY

Model 7001

7001J-901-01 Rev. A / 3-02

A GREATER MEASURE OF CONFIDENCE

改訂状況

下掲の改訂状況は、本書に関して作成された全ての改訂版と補遺の刊行日を示すものです。“改訂水準”文字は、内容更新の都度、アルファベット順に上がっていきます。補遺は、ある改訂版から次の改訂版までの間に刊行され、ユーザが即刻取り入れる必要のある重要な変更情報が記載されます。

補遺には連続番号が付けられます。新改訂版が作成されると、前改訂版に関連する全ての補遺が新改訂版に組み入れられます。各々の新改訂版には、この改訂状況ページが改訂されて掲載されます。

改訂版A (ドキュメント番号7001-901-01) ...1991年11月

補遺A (ドキュメント番号7001-901-02)1991年11月

安全上の注意

以下に記載されている安全上の注意事項は、本製品および関連した機器を使用する前に、必ず確認して下さい。ある機器とアクセサリは、通常危険のない電圧で使用されますが、危険な障害が存在するような状況下で使用されているかもしれません。

本製品は、危険な障害を認識でき、考えられる障害を避けるために必要な安全上の注意に精通している、資格を備えた者によって使用することを意図しています。製品を使用する前に、必ず操作説明を読んで下さい。

障害が存在するときには、特別な注意を払って下さい。致命的な電圧が、ケーブルの接続ジャックまたはテスト装置に存在しているかもしれません。米国標準協会 (ANSI) は、電圧レベルが30V RMS、また60VDCが確認されるときに障害が存在すると述べています。よい安全対策は、計測を行なう前にあらゆる未知の回路に危険な電圧が存在していると考えることです。

機器を使用する前に、ライン・コードが適切にアースされた電力コンセントに接続されていることを確認して下さい。毎回使用する前には、接続ケーブル、テスト・リード、ジャンパーに磨耗やヒビ、または故障がないかどうかを検査して下さい。

安全な作業を行なうために、テストが行なわれている回路に電力が適用されている間は、製品やテスト・ケーブル、または他の機器に触れないで下さい。ケーブルまたはジャンパーの接続および取り外し、スイッチング・ボードの設置または取り外し、ジャンパーの設置や取り外しのような内部の変更を行なう前には必ずテスト・システム全体から電力を除去してコンデンサーを放電して下さい。

テストが行なわれている回路のコモン側または電力ラインのアースへ電流の道を提供することになりかねない物体には触れないで下さい。常に、計測が行なわれる電圧に耐えることができるように絶縁されている乾燥した表面に立ち、乾いた手で計測を行なって下さい。


スペックと操作説明に定義されている、また機器またはテスト装置の後部パネルまたはスイッチング・カード上に示されている、機器およびアクセサリの最大信号レベルを越えないで下さい。


無制限の電力回路にスイッチング・カードを直接接続しないで下さい。カードはインピーダンスによって制限されたソースと共に使用するよう意図されています。決してスイッチング・カードを直接ACメインに接続しないで下さい。ソースをスイッチング・ボードに接続するときは、カードへの誤った電流と電圧を制限するための保護装置を設置して下さい。

製品にヒューズが使用されているときは、火災障害に対する予防として、同じタイプおよびレートのもものと交換して下さい。

本体のコネクションは、安全対策のアース接続としてではなく、回路計測のためのシールド・コネクションとしてだけに使用されるべきです。

もしテスト機器を使用しているならば、テストが行なわれている装置に電力が提供されている間、フタを閉めて下さい。安全を確実にするには、フタのインターロックを使用します。

もし  スクリューが機器に装備されている場合は、#18AWGまたはラージ・ワイヤーを使って、これをセーフティー・アース・グラウンドに接続します。

機器またはアクセサリの  シンボルは、ターミナル上に1000Vまたはそれ以上の電圧が存在することを示しています。詳細な操作説明については、製品の取り扱い説明書を参照して下さい。

機器およびアクセサリを人体に接続しないで下さい。

メンテナンスは、有資格のサービス要員によって行なわれるべきです。メンテナンスを実行する前には、ライン・コードとすべてのテスト・ケーブルを取り外して下さい。

Model7001スイッチングメインフレーム

システム

容量：2プラグインカード/メインフレーム。

メモリ：100スイッチパターン、バッテリバックアップ。

スイッチセットリング時間：カードごとにメインフレームが自動的に選択。それ以上は0から99999.99秒まで1msずつ増加可能。

トリガソース：

外部トリガ (TTLコンパチブル、プログラマブルエッジ、600ns最小パルス、リヤパネルBNC)

IEEE-488バス (GET、*TRG)

トリガリンク

マニュアル (フロントパネル)

内部トリガ、1msから99999.99秒まで1msずつプロドラム可能。

ステータス出力：チャンネルレディー (TTLコンパチブル信号、リヤパネルBNC)。リレーセットリング時間の経過後にLOWパルス発生。異なる2枚のカードに対しては7001を最遅リレーセットリング時間に設定。

スイッチングシーケンス：自動ブレイク・ビフォア・メイク。

メインフレームデジタルI/O：オープンコレクタ出力

(30V最大プルアップ電圧、100mA最大シンク電流、10Ω出力インピーダンス)、1TTLコンパチブル入力、1コモン。

リレードライブ：両カードスロットとも700mA以下。

カードサイズ：32mm (高さ) × 114mm (幅) × 272mm (長さ)。

カード互換性：705、706、7XXXカードと互換性あり。

スループット

スキャンリストの実行速度：200チャンネルまたはメモリアクション/秒。

トリガ実行時間 (トリガソースの起動からスイッチの閉鎖の開始までの時間²)：

ソース	待ち時間	ジッタ
GET ¹	400 μs	<100 μs
*TRG ¹	5.0m	<13 μs
トリガリンク	400 μs	<13 μs
外部	400 μs	<13 μs
タイマ	400 μs	<13 μs

¹いかなるIEEE-488コマンドも実行中ではないものとする。

²スイッチセットリング時間を除く。

IEEE-488コマンド実行時間

コマンド	実行時間 ¹
:ROUT:OPEN (@1!1)	<29ms + リレー開放時間
:ROUT:CLOS (@1!1)	<29ms + リレー閉鎖時間
:ROUT:MEM:REC M1	<32ms + リレーセットリング時間

¹バスからコマンドターミネータが受信された時点からリレー付勢までの時間を測定したもの。

アナログバックプレーン

信号：4つの3ボール行 (Hi、Lo、Guard)。これらの信号により1メインフレーム内でマトリックス拡張とマルチプレクサ拡張が可能。

最大電圧：250VDC、250VRMS、350VACピーク、信号バス間または信号バスと本体間。

最大電流：1Aピーク。

バスアイソレーション：

>1010Ω、<50pF バス間 (任意のHi、Lo、Guardと別のHi、Lo、Guard間)。

>1010Ω、<50pF 差動 (HiとLo間またはHi、LoとGuard間)。

>109Ω、<75pF バスと本体間。

チャンネルクロストーク：<-65dB、1MHz (50Ω負荷) 時。

バンド幅：<3dBロス、100MHz (50Ω負荷) 時。

IEEE-488バス

準拠規格：SCPI-1990、IEEB-488.2、IEEB-488.1に準拠。

マルチラインコマンド：DCL、LIO、SDC、GET、GTL、UNT、UNL、SPE、SPD。

ユニラインコマンド：IFC、REN、EOI、SRQ、ATN。

インタフェースファンクション：SH1、AH1、T5、TE0、L4、LE0、SR1、RL1、PP0、DC1、DT1、C0、E1。

一般仕様

ディスプレイ：

デュアルライン真空蛍光ディスプレイ。

第1ライン：20文字 (英数字)。

第2ライン：32文字 (英数字)。

リヤパネルコネクタ：

IEEE-488

8ピン・マイクロDINコネクタ、デジタルI/O用

8ピン・マイクロDINコネクタ、トリガリンク用

8ピン・マイクロDINコネクタ、トリガリンク拡張用

BCN、外部トリガ用

BNC、チャンネルレディー用

電源：90-260VACユニバーサル入力、47-440Hz、40VA以下。

安全設計：UL 1244、IEC 348に適合。

EMI/RFI：VDE 0871B、FCC Class Bに適合。

環境条件：

動作：0~50℃、<80% RH (0~35℃)。

保存：-25℃~+65℃。

寸法、重量：89mm (高さ) × 216mm (幅) × 375mm (長さ)、3.4kg。

仕様は予告なく変更されることがあります。

目次

1.1	概要	1-1
1.2	特徴	1-1
1.3	保証関係	1-2
1.4	取扱説明書補遺	1-2
1.5	安全記号および用語	1-2
1.6	仕様	1-2
1.7	検査	1-2
1.8	オプションアクセサリ	1-3
2.1	概要	2-1
2.2	Model 701Xシリーズのカードの装置	2-2
2.3	701Xシリーズ以外のカード	2-2
3.1	概要	3-1
3.2	フロントパネルおよびリアパネルの構成	3-1
3.2.1	フロントパネルコントロール類	3-1
3.2.2	リアパネル	3-1
3.2.3	チャンネルステータスディスプレイ	3-4
3.3	オペレーションデモ	3-7
3.3.1	イニシャルコンフィギュレーション	3-7
3.3.2	チャンネルの開閉	3-7
3.3.3	チャンネルのスキャン	3-8
3.4	スキャンプロセスのあらまし	3-10
3.5	イニシャルコンフィギュレーション	3-11
3.5.1	スイッチングカードシュミレータ	3-12
3.5.2	スイッチングカードの装置	3-12
3.5.3	カードタイプ	3-12
3.6	フロントパネル操作	3-13
3.6.1	開閉操作例	3-14
3.6.2	スキャン操作例	3-15
3.7	IEEE-488.2 および SCPI の基礎事項	3-19
3.7.1	バスの接続	3-19
3.7.2	一次アドレス	3-19
3.7.3	略記コモンコマンドサマリ	3-19
3.7.4	略記コマンドサマリ	3-19
3.7.5	シンタックスルール（構文規則）	3-21
3.7.6	プログラミング例	3-22

4.1	概要	4-1
4.2	電源投入手順	4-1
4.2.1	電源接続	4-1
4.2.2	電源スイッチ	4-2
4.2.3	電源投入シーケンス	4-2
4.3	ディスプレイ	4-3
4.3.1	チャンネルステータスディスプレイ	4-4
4.3.2	インフォメーションメッセージとエラーメッセージ	4-5
4.3.3	アナンシエータ	4-5
4.4	アナログバックプレーン	4-7
4.5	メインフレームプログラミング	4-8
4.5.1	チャンネルアサインメント	4-8
4.5.2	チャンネルリストとスキャンリスト	4-10
4.5.3	チャンネルの開閉	4-13
4.5.4	チャンネルのスキャン	4-14
4.5.5	チャンネルパターンのストア(STOREおよびRECALL)	4-21
4-6	MENU	4-23
4.6.1	SAVESETUP	4-24
4.6.2	GPIB	4-25
4.6.3	DIGITAL I/O	4-26
4.6.4	TEST	4-27
4.6.5	LANGUAGE	4-28
4.6.6	GENERAL	4-28
4.7	CARD CONFIG	4-29
4.7.1	TYPE	4-30
4.7.2	#-OF-POLES	4-31
4.7.3	CARD-PAIR	4-32
4.7.4	DELAY	4-32
4.7.5	READ-I/O-CARD	4-33
4.8	SCAN CONFIG	4-34
4.8.1	CHAN-CONTROL	4-36
4.8.2	SCAN CONTROL	4-39
4.8.3	ARM CONTROL	4-41
4.8.4	チャンネル制限	4-43
4.9	デジタル I/O ポート	4-44
4.9.1	デジタル出力チャンネル	4-44
4.9.2	デジタル入力チャンネル	4-44
4.9.3	I/O ポートの接続	4-45
4.10	外部トリガ	4-47
4.10.1	外部トリガ	4-47
4.10.2	チャンネルレディ	4-47
4.10.3	外部トリガ例	4-48
4.11	Trigger Link (トリガリンク)	4-50
4.11.1	非同期オペレーション	4-50
4.11.2	半同期オペレーション	4-59

5.1	概要	5-1
5.2	IEEE-488バスの接続	5-2
5.3	一次アドレスの選択	5-4
5.4	コントローラプログラミング	5-4
5.5	IEEE-488オペレーションのフロントパネル関連事項	5-5
5.5.1	フロントパネルエラーメッセージ	5-5
5.5.2	IEEE-488ステータスインジケータ	5-6
5.5.2	LOCALキー	5-7
5.6	ステータス構造	5-7
5.6.1	標準イベントステータス	5-11
5.6.2	オペレーションイベントステータス	5-12
5.6.3	アームイベントステータス	5-15
5.6.4	シーケンスイベントステータス	5-18
5.6.5	トリガイイベントステータス	5-20
5.6.6	被疑イベントステータス	5-23
5.6.7	待ち行列	5-23
5.6.8	ステータスおよびサービス要求 (SRQ)	5-24
5.7	トリガモデル (IEEE-488 スキャンプロセス)	5-26
5.8	汎用バスコマンド	5-23
5.8.1	REN (リモートイネーブル)	5-32
5.8.2	IFC (インターフェースクリア)	5-33
5.8.3	LLO (ローカルロックアウト)	5-33
5.8.4	GTL (go to local)とローカル	5-33
5.8.5	DCL (装置クリア)	5-33
5.8.6	SDC (選択的装置クリア)	5-34
5.8.7	GET (グループ実行トリガ)	5-34
5.8.8	SPE,SPD (シリアルポーリング)	5-34
5.9	コモンコマンド	5-35
5.9.1	*CLS - clear status	5-37
5.9.2	*ESE - event status enable	5-38
5.9.3	*ESE? - event status enable query	5-40
5.9.4	*ESR? - event status register query	5-41
5.9.5	*IDN? - identification query	5-43
5.9.6	*OPC - operation complete	5-44
5.9.7	*OPC? - operation complete query	5-45
5.9.8	*OPT? - option identification query	5-46
5.9.9	*RCL - recall	5-47
5.9.10	*RST - reset	5-48
5.9.11	*SAV - save	5-50
5.9.12	*SRE - service request enable	5-52
5.9.13	*SRE? - service request enable query	5-54
5.9.14	*STB? - status byte query	5-55
5.9.15	*TRG - trigger	5-57
5.9.16	*TST - self-test query	5-58
5.9.17	*WAI - wait-to-continue	5-59
5.10	SCPIコマンドサブシステム	5-61
5.10.1	:DISPLay サブシステム	5-62
5.10.2	:OUTPut サブシステム	5-68

5.10.3	[:ROUTe] サブシステム	5-70
5.10.4	:SENSe[1] サブシステム	5-87
5.10.5	:SOURce サブシステム	5-88
5.10.6	:STATus サブシステム	5-94
5-10.7	SYSTem サブシステム	5-128
5.10.8	トリガサブシステム	5-131

図一覽

図2-1	マルチピンカードの装置	2-4
図2-2	ねじ込み端子式カードの装置	2-5
図3-1	Model 7001フロントパネル	3-2
図3-2	Model 7001リアパネル	3-3
図3-3	チャンネルステータスディスプレイ	3-5
図3-4	チャンネルアサインメント (マトリックスカード)	3-5
図3-5	チャンネルアサインメント (マトリックスタイプ以外のカード)	3-6
図3-6	スキャンオペレーションの簡略モデル	3-10
図4-1	表示フォーマット	4-3
図4-2	チャンネルステータスディスプレイ	4-4
図4-3	チャンネルステータスディスプレイの解釈の仕方	4-5
図4-4	Model 7001アナログバックプレーン	4-7
図4-5	バックプレーンへのマトリックス行の接続	4-8
図4-6	チャンネルアサインメント (非マトリックスタイプのカード)	4-9
図4-7	チャンネルアサインメント (マトリックスカード)	4-10
図4-8	フロントパネルスキャンオペレーション	4-17
図4-9	デジタル I/O コンセント	4-45
図4-10	Trigger Link ケーブの仕様によるデジタル I/O 接続	4-46
図4-11	外部トリガ用コネクタ(BNC)	4-47
図4-12	外部トリガおよび非同期 Trigger Link 入力パルス仕様	4-47
図4-13	チャンネルレディおよび非同期 Trigger Link 出力パルス仕様	4-47
図4-14	DUT テストシステム	4-48
図4-15	外部トリガコネクタ	4-49
図4-16	Trigger Link コネクタ	4-50
図4-17	DUT テストシステム	4-51
図4-18	Trigger Link コネクタ (非同期例#1)	4-52
図4-19	非同期 Trigger Link 例#1のオペレーションモデル	4-54
図4-20	Trigger Link アダプタの使用による接続	4-55
図4-21	DUT テストシステム (非同期例#2)	4-56
図4-22	Trigger Link の使用による接続 (非同期例#2)	4-56
図4-23	非同期 Trigger Link 例#2のオペレーションモデル	4-58
図4-24	半同期 Trigger Link 仕様	4-59
図4-25	標準的な半同期モード接続	4-60
図4-26	Trigger Link 接続 (半同期例)	4-60
図4-27	半同期 Trigger Link 例のオペレーションモデル	4-62

図5-1	IEEE-488 コネクタ	5-2
図5-2	IEEE-488 の接続	5-2
図5-3	IEEE-488 コネクタの位置	5-3
図5-4	接点アサインメント	5-3
図5-5	Model 7001 ステータスレジスタ構造	5-9
図5-6	標準イベントステータス	5-11
図5-7	オペレーションイベントステータス	5-13
図5-8	アームイベントステータス	5-16
図5-9	シーケンスイベントステータス	5-19
図5-10	トリガイベントステータス	5-21
図5-11	ステータスバイトおよびサービス要求 (SEQ)	5-25
図5-12	トリガモデル (IEEE-488 スキャンプロセス)	5-28
図5-13	標準イベントステータスイネーブルレジスタ	5-39
図5-14	標準イベントステータスレジスタ	5-42
図5-15	サービス要求イネーブルレジスタ	5-53
図5-16	ステータスバイトレジスタ	5-56
図5-17	オペレーションイベントステータスレジスタ	5-97
図5-18	オペレーションイベントトランシジョンフィルタ	5-101
図5-19	オペレーションイベントステータスイネーブルレジスタ	5-103
図5-20	アームイベントステータスレジスタ	5-104
図5-21	アームイベントトランシジョンフィルタ	5-107
図5-22	アームイベントステータスイネーブルレジスタ	5-110
図5-23	シーケンスイベントステータスレジスタ	5-112
図5-24	シーケンスイベントトランシジョンフィルタ	5-115
図5-25	シーケンスイベントステータスイネーブルレジスタ	5-117
図5-26	トリガイベントステータスレジスタ	5-119
図5-27	トリガイベントトランシジョンフィルタ	5-122
図5-28	トリガイベントステータスイネーブルレジスタ	5-124

表一覽

表3-1	略記コモンコマンドサマリ	3-20
表3-2	略記SPCIコマンドサマリ	3-20
表4-1	エラーメッセージとステータスメッセージ	4-6
表4-2	MENU構造	4-23
表4-3	デフォルトの状態	4-26
表4-4	CARD CONFIG MENUの構造	4-29
表4-5	TYPE指定	4-30
表4-6	CONFIGURE SCANメニュー構造	4-34
表5-1	IEEE接点アサイメント	5-3
表5-2	HP BASIC 4.0によるIEEE-488ステートメント	5-5
表5-3	エラーメッセージとステータスメッセージ	5-5
表5-4	汎用バスコマンドおよび付随のBASICステートメント	5-32
表5-5	IEEE-488.2コモンコマンドおよび問い合わせ	5-36
表5-6	デフォルト	5-49
表5-7	*SAVおよび*RCLオプション	5-51
表5-8	DISPlay コマンドサマリ	5-62
表5-9	OUTPUT コマンドサマリ	5-68
表5-10	ROUTE コマンドサマリ	5-70
表5-11	ASENSE コマンドサマリ	5-87
表5-12	SOURCE コマンドサマリ	5-88
表5-13	STATUS コマンドサマリ	5-94
表5-14	SYSTEM コマンドサマリ	5-128
表5-15	トリガコマンドサマリ	5-131

第1部

一般事項

1.1 概要

本節では、Model 7001スイッチシステムに関する一般事項について記述します。

1.2 特長

1.3 保証関係

1.4 取扱説明書補遺

1.5 安全記号および用語

1.6 仕様

1.7 検査

1.8 オプションアクセサリ

1.2 特長

Model 7001の重要な特長をいくつか挙げると、次の通りです。

- 高密度スイッチング - 2枚のスイッチングカードを使うことで1メインフレームにつき最大80チャンネルまで可能です。

- スwitchingカード — Model 7001用に設計されているスイッチングカード（すなわち、Model 7011、7012および7013）に加え、もともとModel 705および706スキャナ用に設計されたスイッチングカードも使用できます。
- アナログバックプレーン — 一方のスロットに装着されているModel 701Xシリーズカードの行とバンクを他方のスロットに装着されている別のModel 701Xシリーズカードの行とバンクに内部接続することができます。
- チャンネル開閉またはスキャン — Model 7001は1つまたはそれ以上のチャンネルを簡単に開閉することができ、あるいは指定されたチャンネルリストを簡単にスキャンできます。
- チャンネルステータスディスプレイ — リアルタイムステータスディスプレイで全ての使用可能チャンネルの状態（閉じているか開いているか）をモニタします。
- メモリ — 最大100のチャンネルパターン（開チャンネルと閉チャンネルのパターン）と、カスタム化された10の機器セットアップをメモリにセーブし、後でリコールすることができます。
- IEEE-488バス — バス操作はIEEE-488.2およびSCPI規格に準拠します。
- トリガリンク — 標準のIN/OUT BNC外部トリガに加え、より汎用性が高く精確な外部トリガが可能な新しいトリガ方式を採用しています。

1.3 保証関係


保証に基づくサービスを期間内に必要とされる場合には、ケースレーインストルメンツ株式会社サービスセンターまでお問い合わせ下さい。担当が、修理に必要な情報を尋ねた上で、本機の返送に関する指示をお伝えします。


1.4 取扱説明書補遺

機器または取扱説明書に関する改良または変更については、取扱説明書とともに同包される補遺の中で説明します。このような変更に注意し、変更内容を取扱説明書に組み入れてください。

1.5 安全記号および用語

機器上には以下の記号が表示されていることがあります。また、本書中では以下の用語が使われることがあります。

機器上の記号  は、ユーザが本書中の操作説明を参照する必要があることを表わすものです。

機器上の記号  は、端子上に高電圧が存在している可能性があることを表わすものです。このような電圧に人体が触れないように安全対策を講じてください。

本書中の見出し**警告**は、傷害や死亡につながる恐れのある危険について説明するものです。指示された手順を実行する前に、必ず、付随の説明をよく読んでください。

本書中の見出し**注意**は、スキャナカードの損傷につながる恐れのある危険について説明するものです。このような損傷には保証が適用されないことがあります。

1.6 仕様

Model 7001の仕様は本書の初めの部分に記載してあります。

1.7 検査

Model 7001は、出荷の前に、電気関係、機械関係とも入念な検査を済ませてあります。解梱して全品目を取り出したら、輸送中に物理的損傷が生じていないかどうか検査してください。損傷があった場合は、ただちに輸送業者にその旨報告してください。元の荷箱は、後で返送が必要になる場合に備えて保存しておいてください。Model 7001のご発注に対してお送りする梱包内容は次の通りです。

- Model 7001スイッチシステム
- Model 7001取扱説明書
- Model 8501-1トリガリンクケーブル
- ご注文のあった付加アクセサリ

1.8 オプションアクセサリ

Model 7001用として以下のアクセサリをお求めいただけます。

Model 4288-1単式固定ラックマウントキット；

1台のModel 7001を19インチの標準ラック内に取り付けるもの。

Model 4288-2複式固定ラックマウントキット；

2台のModel 7001を19インチの標準ラック内に並列に取り付けるもの。

Model 4288-3複式固定ラックマウントキット；

1台のModel 7001とModel 199 DMMを19インチの標準ラック内に並列に取り付けるもの。

Model 8501トリガリンクアダプタ；

Model 7001のトリガリンクを、標準的なBNC (IN/OUT) 外部トリガ法を使う機器に接続可能にするもの。

Model 8501-1および8501-2トリガリンクケーブル；

Model 8501-1は長さ1m、Model 8501-2は2mです。

Model 7007 シールドIEEE-488ケーブル；

電磁干渉 (EMI) を低減するためにModel 7001をシールドケーブルでIEEE-488バスに接続するもの。Model 7007-1は長さ1mで、各端にEMIシールドIEEE-488コネクタが付いています。Model 7007-2もModel 7007-1と同じですが、長さが2mです。

第2部

カードの装着

初めてお使いになる方のために

Model 7001を初めてお使いになる方は、スイッチングカードをメインフレーム内に装着する前に、第3節「基礎的事項」に記載されているオペレーションデモ（3.3項）を実行することをお勧めします。このデモで基本的なフロントパネル操作を知っていただけます。

警告

本節に記載されている手順は、資格のあるサービススタッフだけが使うことを前提としています。資格がない限り、これらの手順を実行しないでください。一般的な安全上の注意事項を確認、順守することを怠ると、傷害や死亡につながる恐れがありますから、十分注意してください。

本節に記載の装着手順は次のような構成にまとめられています。

2.2 Model 701Xシリーズのカードの装着：

Model 701Xシリーズのカード（Model 7011、7012、7013など）をModel 7001内に装着する方法について説明します。

2.3 701Xシリーズ以外のカードの装着：

Model 705 / 706スキャナ用に設計されたカードをModel 7001内に装着する方法について説明します。

2.1 概要

Model 7001は、Model 701Xシリーズのスイッチカード（すなわち、Model 7011、7012、7013）ならびに、もともとKeithley Model 705/706スキャナ用に作られたカードと共に使用するように設計されています。

2.2 Model 701Xシリーズのカードの装着

警告

全機器 (Model 7001メインフレームを含む) の電源を切り、それぞれのラインコードを外してください。電力がすべて排除され、外部回路内の蓄積エネルギーが放出されたことを確認してください。

注

ねじ込み端子式コネクタカードを使用する場合は、カードアセンブリをModel 7001メインフレーム内に装着する前に、外部回路が (カード取扱説明書の説明に従って) カードに配線接続されていることを確認してください。

注意

スイッチカードが汚れると性能低下を来すことがあります。これを防止するため、カードアセンブリを取り扱うときはそのエッジとシールド部分以外には触れないようにしてください。汚れた場合は、カード取扱説明書の説明に従ってカードをクリーニングしてください。

カードの装着

以下のステップを実行することにより、カードアセンブリをModel 7001メインフレーム内に装着します。

1. コネクタカードとリレーカードが分かれている場合は、前者を後者に組み付けます。カードを取り扱うときは、汚れないようにエッジとシールド部分以外には触れないでください。
2. Model 7001のリヤパネル側に向かい、カードを装着したいスロット (CARD 1またはCARD 2) を選択します。
3. 図2-1または図2-2を参照しながら、リレーカードのエッジがレールに掛かるようにしてカードアセンブリを目的のスロットに送り込みます。
4. エゼクタアームがアンロック位置にある状態で、アームがエゼクタカップとかみ合うまでカードアセンブリをメインフレーム内に押し込みます。そして、両方のアームを内向きに押してカードを完全にメインフレーム内にロックします。

5. マルチピンコネクタカードの場合、図2-1に示すねじを取り付けます。

注

メインフレーム内に701Xシリーズのカードが装着されれば、Model 7001は電源投入時に自動的にそれを識別します。このように識別することで、Model 7001は正しい動作が可能な設定になります。

カードの取出し

カードアセンブリを取り出すには、エゼクタアームを外向きに引いてアンロックし、カードアセンブリをメインフレームから引き抜きます。カードアセンブリを取り扱うときは、性能低下につながる汚れを避けるため、エッジとシールド部分以外には触れないようにすることを忘れないでください。

2.3 701Xシリーズ以外のカード

警告

全機器 (Model 7001メインフレームを含む) の電源を切り、それぞれのラインコードを外してください。電力がすべて排除され、外部回路内の蓄積エネルギーが放出されたことを確認してください。

注

それ自体での接続を必要とするカードを使用する場合は、カードアセンブリをModel 7001メインフレーム内に装着する前に、外部回路が (カード取扱説明書の説明に従って) カードに配線接続されていることを確認してください。

注意

スイッチカードが汚れると性能低下を来すことがあります。これを防止するため、カードアセンブリを取り扱うときはそのエッジ以外には触れないようにしてください。汚れた場合は、カード取扱説明書の説明に従ってカードをクリーニングしてください。

カードの装着

以下のステップを実行することにより、カードアセンブリをModel 7001メインフレーム内に装着します。

1. Model 7001のリヤパネル側に向かい、カードを装着したいスロット (CARD 1またはCARD 2) を選択します。
2. 図2-1または図2-2を参照しながら、カードのエッジがレールに掛かるようにしてカードを目的のスロットに送り込みます。
3. エゼクタアームがアンロック位置にある状態で、アームがエゼクタカップとかみ合うまでカードアセンブリをメインフレーム内に押し込みます。そして、両方のアームを内向きに押しカードを完全にメインフレーム内にロックします。

注

Model 7001は、701X以外のカードを自動的に識別することはできません。メインフレームの電源を投入した後、ユーザはCARD CONFIG MENUのメニュー項目であるTYPEを使って型番を入力しなければなりません (第3節「基礎的事項」の3.5.3項参照)。

カードの取出し

カードを取り出すには、エゼクタアームを外向きに引いてアンロックし、カードをメインフレームから引き抜きます。カードを取り扱うときは、性能低下につながる汚れを避けるため、エッジ以外には触れないようにすることを忘れないでください。

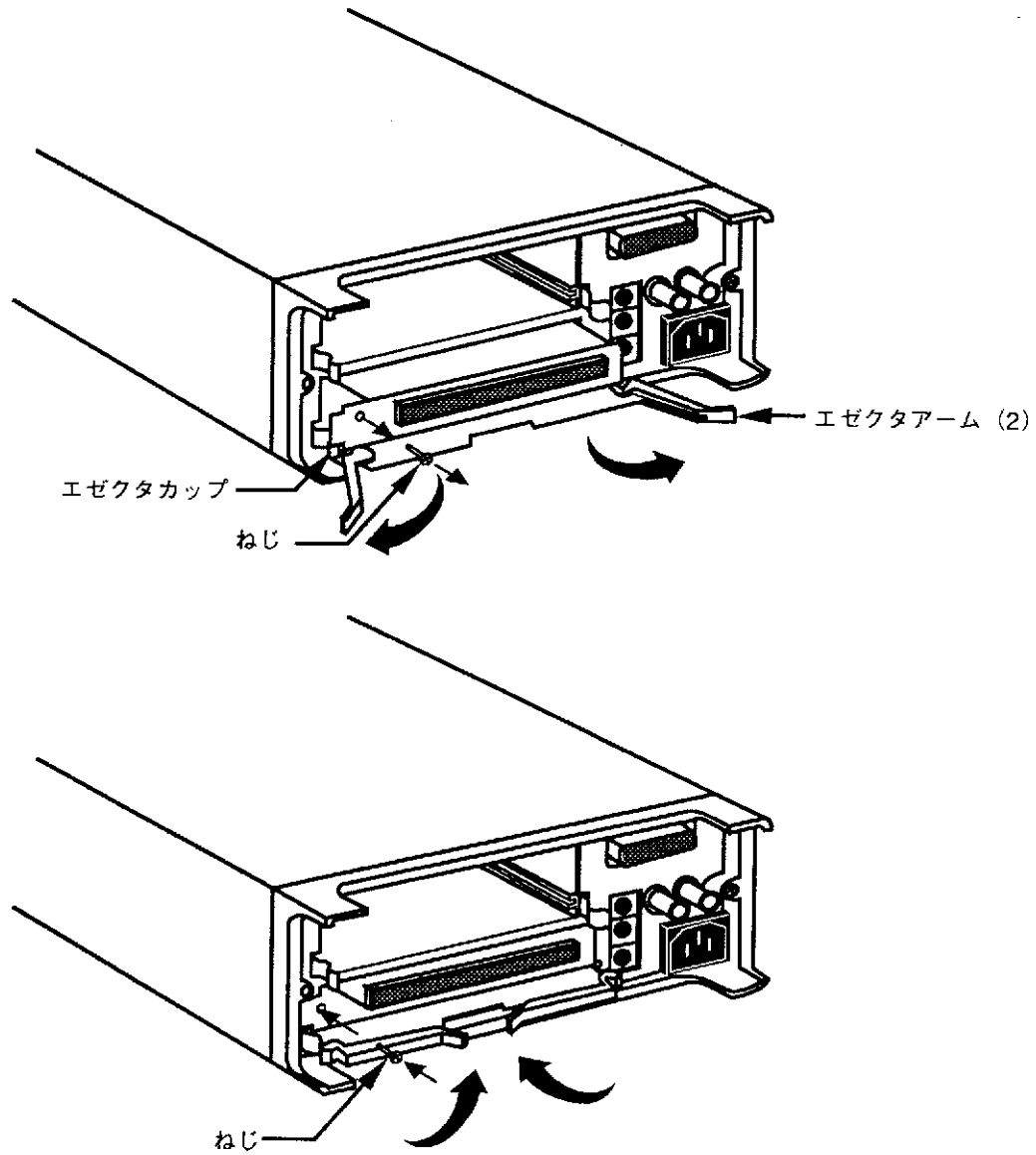


図2-1
マルチピンカードの装着

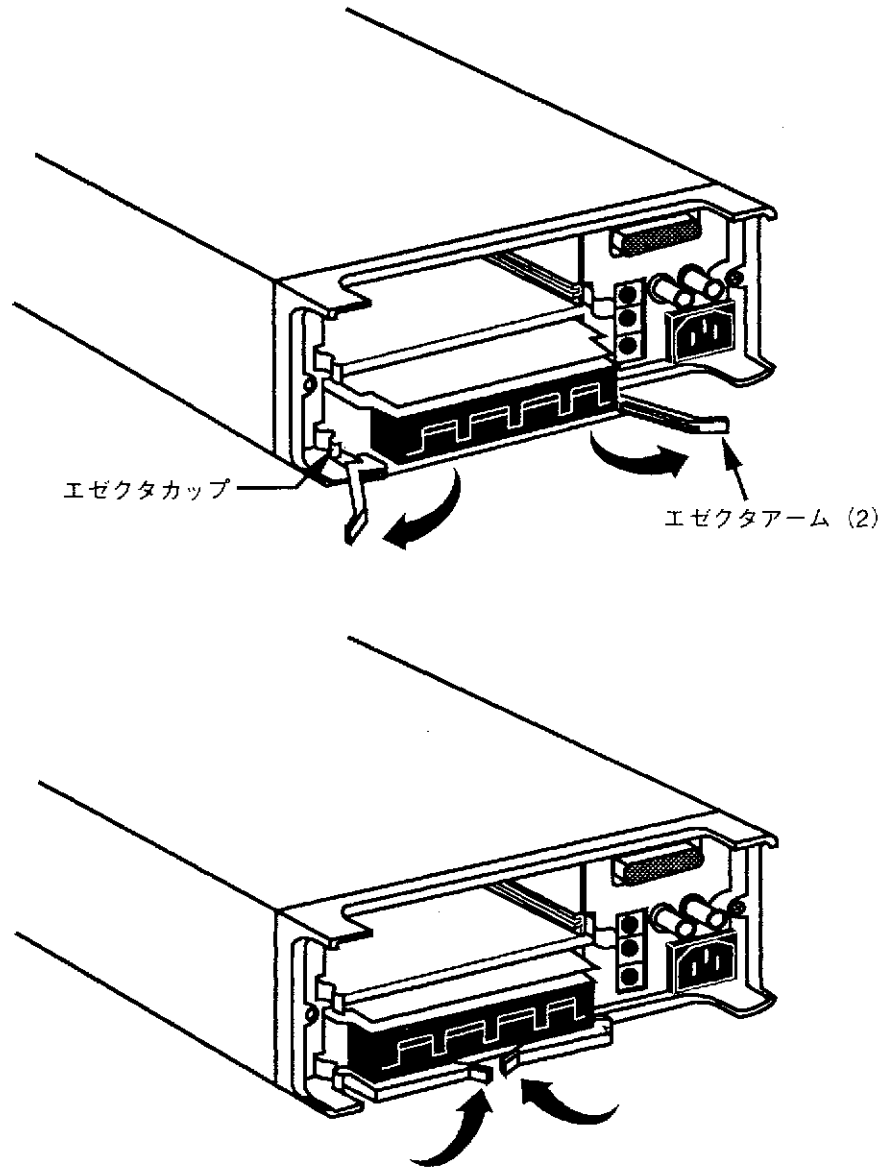


図2-2
ねじ込み端子式カードの装着

第3部

基礎的事項

3.1 概要

本節では、Model 7001スイッチシステムに関する基礎的事項について記述します。フロントパネル操作とIEEE-488バス操作の詳細は、それぞれ第4節と第5節を参照してください。

本節の記載内容は次の通りです。

3.2 フロントパネルおよびリヤパネルの構成：

フロントパネル上のコントロール類とディスプレイについて簡単に説明します。

3.3 オペレーションデモ：

内蔵の40チャンネル・マルチプレクサシミュレータを使って基本操作をデモ実行します。Model 7001を初めてお使いになる方にお勧めします。

3.4 スキャンプロセスのあらまし：

スキャンプロセスのあらましを示します。

3.5 イニシャルコンフィギュレーション：

Model 7001を動作させる前に考慮すべきイニシャルコンフィギュレーション情報についてみてみます。

3.6 フロントパネル操作：

簡単な例を使って基本的なフロントパネル操作をデモ実行します。

3.7 基本的なIEEE-488バス操作：

簡単な例を使ってバス操作をデモ実行します。

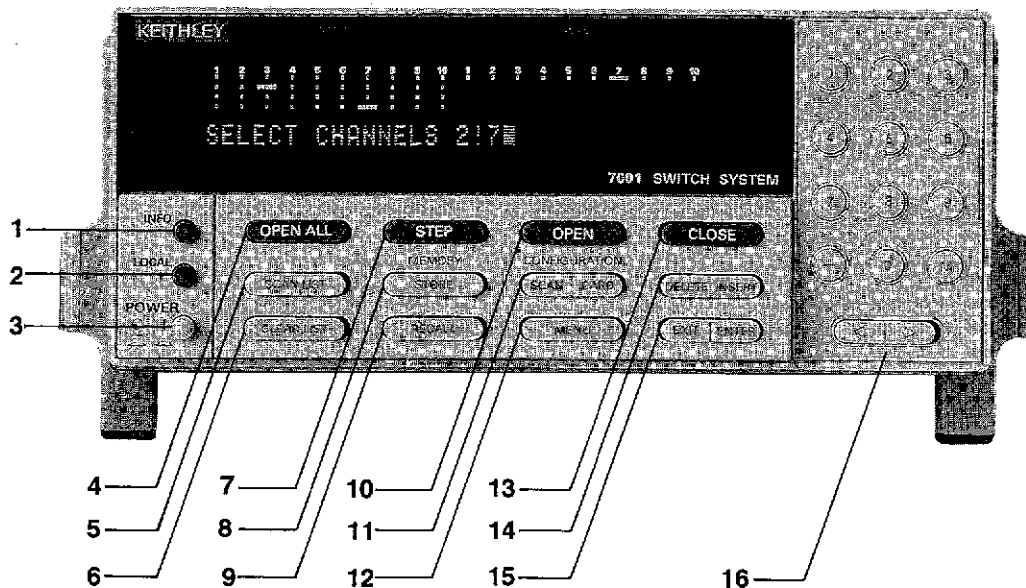
3.2 フロントパネルおよびリヤパネルの構成

3.2.1 フロントパネルコントロール類

Model 7001のフロントパネルコントロール類を図3-1に示します。この図には、Model 7001を動作させる前によく知っておくべき重要な簡略記載情報が含まれます。コントロール類のうちいくつかはデュアルファンクションのロックアクション型キーですから、注意してください。これらのキーとしては、SCAN/CARD CONFIGURATION、DELETE/INSERT、EXIT/ENTER、<I>があります。

3.2.2 リヤパネル

Model 7001のリヤパネルを図2に示します。この図には、Model 7001を動作させる前によく知っておくべき重要な簡略記載情報が含まれます。



1 INFO

オンラインヘルプ情報を表示する。もう一度押すと（またはEXITを押すと）、メッセージがキャンセルされる。

2 LOCAL

リモートコントロールを取り消し、フロントパネルコントロールに戻る。

3 POWER

0 = オフ
1 = オン

4 OPEN ALL

両スロットの全チャンネルを開く。また、進行中のスキャンを打ち切り、7001をアイドル状態にする。

5 SCAN LIST

スキャンリストとチャンネルリストをトグル切替える。

6 CLEAR LIST

表示されているチャンネルリストまたはスキャンリストをクリアする。

7 STEP

7001をアイドル状態から抜け出させ、スキャンリストをステップスルーする。

8 STORE

チャンネルパターンを指定されたメモリロケーションにストアする。

9 RECALL

7001を指定されたメモリロケーションにストアされているチャンネルパターンに復元する。

10 OPEN

チャンネルリスト中の、指定された（表示されている）チャンネルを開く。

11 SCAN CONFIGURATION

以下に示すCONFIGURE SCAN MENUを表示させる。

CHAN-CONTROL
SCAN-CONTROL
ARM-CONTROL
CHAN-RESTRICTIONS

CARD CONFIGURATION

以下に示すCARD CONFIGURE MENUを表示させる。

TYPE
#-OF-POLES
CARD-PAIR
DELAY
READ-I/O-CARD

12 MENU

以下に示すMAIN MENUを表示させる。

SAVESETUP
GPIB
DIGITAL-I/O
TEST
LANGUAGE
GENERAL

13 CLOSE

チャンネルリスト中の、指定された（表示されている）チャンネルを閉じる。

14 DELETE

チャンネルリストまたはスキャンリストから選択された項目を削除する。

INSERT

チャンネルリストまたはスキャンリスト中の、選択された位置に次の項目を挿入する。

15 EXIT

前のメニュー選択項目に戻るか、またはメニューから抜け出る。

ENTER

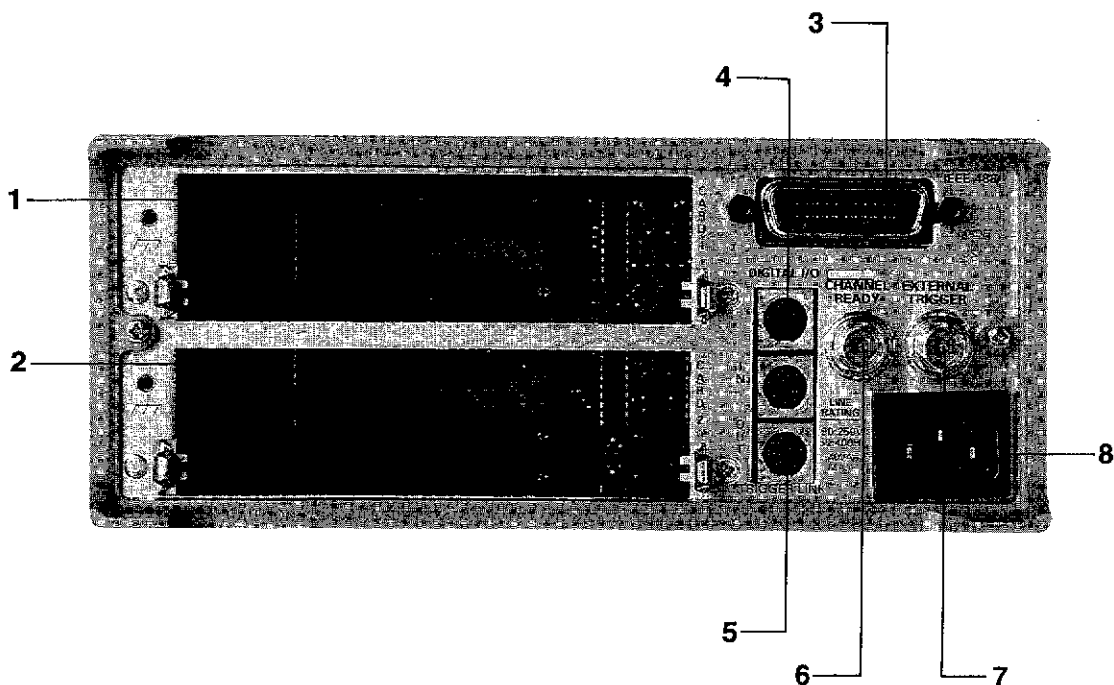
保留中のオペレーションを実行する。また、チャンネルを分離し、チャンネルリストまたはスキャンリストを終了させる。

16 KEYPAD

0~9; 数字の入力に使用。
-; チャンネルのレンジ（すなわち、1!1~1!40）を指定するのに使用。
M; 指定されたメモリロケーション番号の前置として使用。
<; カーソルを左に移動させるのに使用。
>; カーソルを右に移動させるのに使用。また、チャンネルを分離し、チャンネルリストまたはスキャンリストを終了させるのにも使用。

図3-1

Model 7001フロントパネル



- | | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>1 CARD 1
スイッチカード用スロット1。
Card 1のチャンネルステータスディスプレイに対応。</p> <p>2 CARD 2
スイッチカード用スロット2。
Card 2のチャンネルステータスディスプレイに対応。</p> <p>3 IEEE-488 CONNECTOR
標準IEEE-488ケーブルを使用。</p> <p>4 DIGITAL I/O
マイクロ8ピンDINコネクタ。ポートは4本のTTL出力ラインと1本のTTL入力ライン（コモンライン1本）からなる。</p> | <p>5 TRIGGER LINK IN AND OUT
マイクロ8ピンDINコネクタ2個。</p> <p>6 CHANNEL READY
TTL出力トリガパルス用メスBNCコネクタ。</p> <p>7 EXTERNAL TRIGGER
TTL入力トリガパルス用メスBNCコネクタ。</p> <p>8 LINE POWER INPUT
警告：3線電源コードで、アースされたアウトレットに接続すること。</p> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

図3-2
Model 7001 リヤパネル

3.2.3 チャンネルステータスディスプレイ

チャンネルステータスディスプレイは、2スロットそれぞれの使用可能チャンネルのステータスをリアルタイムで表示させます。表示されるのは使用可能チャンネルだけです。チャンネルステータス表示モードは、チャンネルリスト入力モード (SELECT CHANNEL) とスキャンリスト入力モード (SCAN CHANNEL) の2つがあります。電源が投入されると、チャンネルリスト入力モードが選択されます。SCAN LISTキーによってチャンネルリストとスキャンリストをトグル切替えます。EXITキーを1回または複数回押すことによってチャンネルステータスディスプレイに戻ることができます。

チャンネルステータスディスプレイを図3-3に示します。このディスプレイは、2つの10列グリッドとして構成されています。左側のグリッドには、スロット1 (CARD 1) に装着されたカードのチャンネルステータスが表示され、右側のグリッドには、スロット2 (CARD 2) に装着されたカードのチャンネルステータスが表示されます。図示のように、開チャンネルは“ドット”として表わされ、閉チャンネルは“ダッシュ”として表わされます。

チャンネルアサインメント — チャンネルアサインメントフォーマットは、カードタイプによって自動的に決定されます。マトリックスカードチャンネルを表現するためのものとして3整数フォーマットがあり、他のあらゆるタイプのスイッチングカードを表現するためのものとして2整数フォーマットがあります。チャンネルアサインメントを構成する個々の整数は、感嘆符 (!) によって区切られます。

マトリックスカード: マトリックスカードでは、チャンネルアサインメントを表わすのに3整数フォーマットを使います。最初の整数は、カードが装着されるスロット番号を指定します。

すなわち、“1”はスロット1 (CARD 1) を指定し、“2”はスロット2 (CARD 2) を指定します。2番目の整数はマトリックスカードの行番号を指定します。行は全部で4行あり、一番上が行1、その次が行2、その次が行3、一番下が行4です。3番目の整数はマトリックスカードの列を指定します。チャンネルステータスディスプレイには10列が表示されます。全マトリックスカードチャンネルのアサインメントを図3-4に示します。

たとえば、図3-3のCARD 1がModel 7012マトリックスカードであるとすれば、下記のチャンネルが閉じていることになります。

チャンネル1!2!3 (スロット1、行2、列3)
チャンネル1!4!9 (スロット1、行4、列9)

他のカードタイプ: マトリックスタイプ以外のスイッチングカードでは、いずれも、チャンネルアサインメントを表わすのに2整数フォーマットを使います。最初の整数は、マトリックスカードの場合と同様、スロット番号を指定します。2番目の整数は、スイッチカードのチャンネル番号を指定します。これらのカードタイプのチャンネルアサインメントを図3-5に示します。

たとえば、図3-3のCARD 2がModel 7011マルチプレクサカードであるとすれば、下記のチャンネルが閉じていることになります。

チャンネル2!14 (スロット2、チャンネル14)
チャンネル2!36 (スロット2、チャンネル36)

Model 7011マルチプレクサカードの場合、チャンネル14はバンク1、リレー4に対応し、チャンネル36はバンク3、リレー6に対応します。詳細は7011取扱説明書を参照してください。

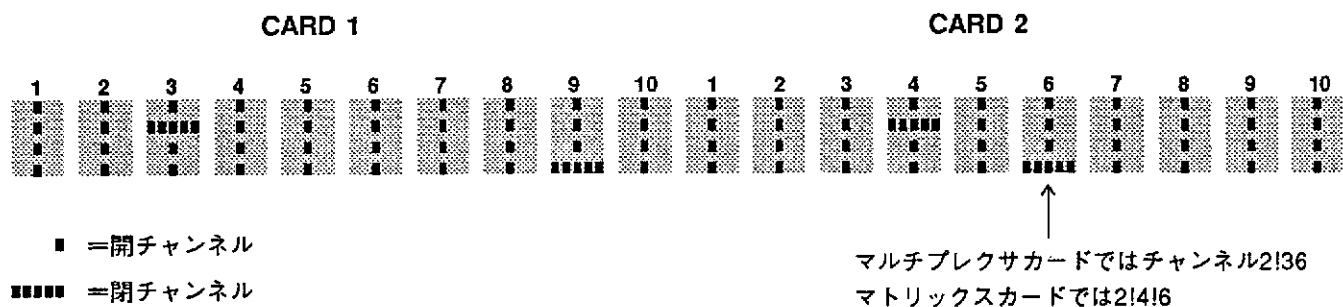
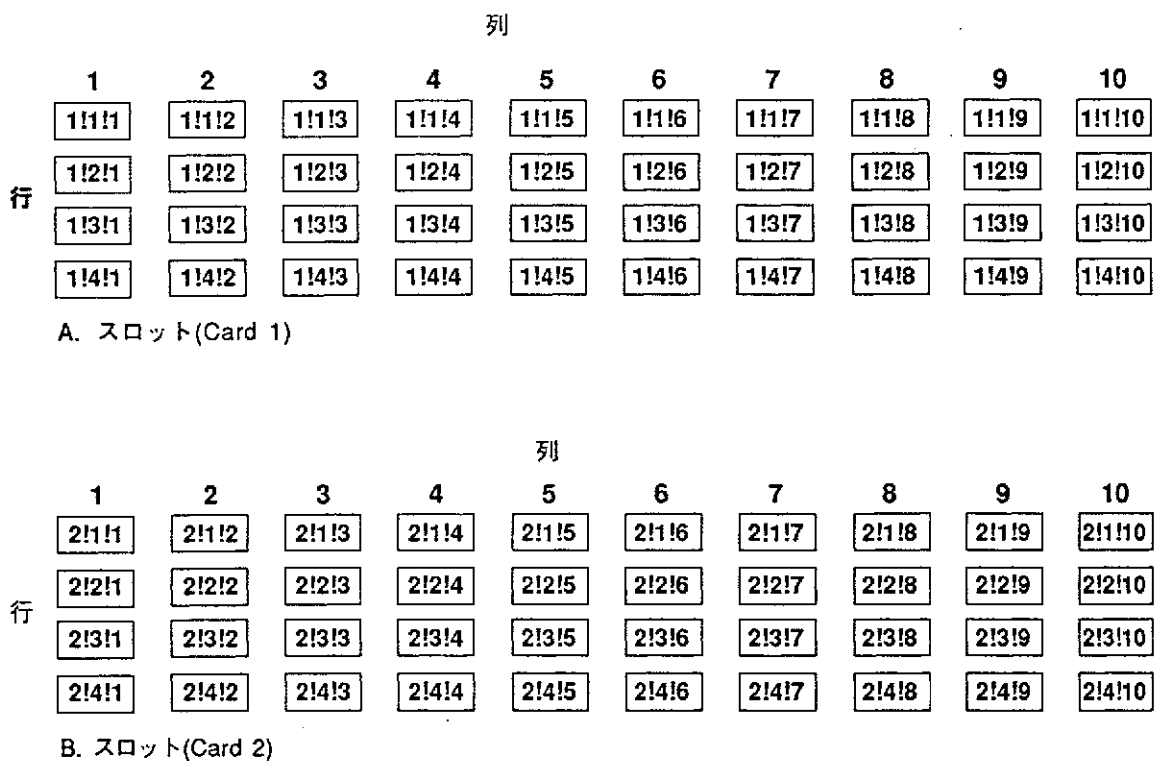


図3-3
 チャンネルステータスディスプレイ



例：1!2!4 = スロット1、行2、列4
 2!3!6 = スロット2、行3、列6

図3-4
 チャンネルアサインメント (マトリックスカード)

1	2	3	4	5	6	7	8	9	10
1!1	1!2	1!3	1!4	1!5	1!6	1!7	1!8	1!9	1!10
1!11	1!12	1!13	1!14	1!15	1!16	1!17	1!18	1!19	1!20
1!21	1!22	1!23	1!24	1!25	1!26	1!27	1!28	1!29	1!30
1!31	1!32	1!33	1!34	1!35	1!36	1!37	1!38	1!39	1!40

A. スロット (Card 1)

1	2	3	4	5	6	7	8	9	10
2!1	2!2	2!3	2!4	2!5	2!6	2!7	2!8	2!9	2!10
2!11	2!12	2!13	2!14	2!15	2!16	2!17	2!18	2!19	2!20
2!21	2!22	2!23	2!24	2!25	2!26	2!27	2!28	2!29	2!30
2!31	2!32	2!33	2!34	2!35	2!36	2!37	2!38	2!39	2!40

B. スロット (Card 1)

例： 1!18 = スロット1、チャンネル18
 2!36 = スロット2、チャンネル36

図3-5
 チャンネルアサインメント (マトリックスタイプ以外のカード)

3.3 オペレーションデモ

基本的なフロントパネル操作に慣れていただく近道は、以下のオペレーションデモを使うことです。このデモでは、内蔵の40チャンネル・マルチプレクサシミュレータが使われます。これによって、スイッチングカードの必要なく操作をシミュレートすることができます。

3.3.1 イニシャルコンフィギュレーション

警告

Model 7001の電源を投入する前に、用意されている電源コードまたは同等品で、アースされたコンセントに接続されていることを確認してください。適切にアースすることを怠ると、感電の危険があり、傷害や死亡につながる恐れがあります。

以下の手順のステップ1は、Model 7001をデモ手順（3.3.2項、3.3.3項参照）に必要なデフォルトのスキャンコンフィギュレーションにRESETするものです。また、ステップ2は、40チャンネル・マルチプレクサシミュレータをスロット2に指定するものです。

ステップ1 Model 7001をRESETする

1. MENUキーを押します。MAIN MENUが表示されます。
2. < および > カーソルキーを使ってカーソルをSAVESETUP上に位置付け、ENTERキーを押します。SETUP MENUが表示されます。
3. カーソルをRESET上に位置付け、ENTERキーを押します。“RESETTING INSTRUMENT”メッセージが表示されます。
4. 再びENTERキーを押します。“RESET COMPLETE”メッセージが表示されます。
5. ENTERキーを押してディスプレイをSETUP MENUに戻します。
6. EXITキーを2回押してMAIN MENUから抜け出します。

ステップ2 マルチプレクサシミュレータを選択する

注

Model 701Xシリーズのカードがスロット2に装着されている場合は、シミュレータは選択できません。

1. CARD CONFIGURATIONキーを押します。CARD CONFIG MENUが表示されます。
2. カーソルをTYPE上に位置付け、ENTERキーを押します。SET CARD TYPEメニューが表示されます。
3. カーソルをSLOT-2上に位置付け、ENTERキーを押します。“SLOT-2 CARD: XXXX”メッセージが表示されます（XXXXはスロット2の現在のカード指定です）。
4. カーソルキーを使って型番“9990”を表示させます。
5. “SLOT-2 CARD: 9990”が表示されている状態で、ENTERキーを押します。SET CARD TYPEメニューが表示されます。
6. EXITキーを2回押してCARD CONFIG MENUから抜け出します。

3.3.2 チャンネルの開閉

以下の手順は、3.3.1項のイニシャルコンフィギュレーション手順が実行済みであることを前提とします。この初期設定により、下記の操作例のために40チャンネル・マルチプレクサシミュレータがスロット2に指定されます。

注

Model 7001は、チャンネルリスト表示モード（“SELECT CHANNELS”プロンプトが表示される）になっていなければなりません。チャンネルステータスディスプレイは、SCAN LISTキーによって、“SELECT CHANNELS”（チャンネルリスト）と“SCAN CHANNELS”（スキャンリスト）をトグル切替えます。Model 7001がチャンネルステータス表示モードになっていない場合は、そのモードになるまでEXITキーを押します。

チャンネルリストの作成：

チャンネルリストは開閉したいチャンネルを指定するものです。チャンネルリストは、単一のチャンネルで構成することも複数のチャンネルで構成することもできます。連続したチャンネルを1レンジとして指定することもできます。

以下の手順は、チャンネル1、チャンネル2、チャンネル6～9を含むチャンネルリストを作成するものです。

1. チャンネルリストが現在空でない場合は、CLEAR LIST キーを押します。チャンネルリストが空であることを示す下記のメッセージが表示されます。

SELECT CHANNELS

2. チャンネル個別入力 - 以下のステップを実行することにより、チャンネル1および2をチャンネルリストに入力します。

- A. “2”を押してから“1”を押し、チャンネル1をチャンネルリストに入力します。“2”はスロット2を選択するものです。

SELECT CHANNELS 2!1

- B. ▷ またはENTERキーを押してチャンネルセパレータ(,)を入力します。

SELECT CHANNELS 2!1

- C. “2”を押しから“2”を押し、チャンネル2を入力します。次に、またはENTERキーを押してチャンネルセパレータ(,)を入力します。

SELECT CHANNELS 2!1,2!2

3. レンジ入力 - 以下のステップを実行することにより、チャンネル6～9を1レンジとして入力します。

- A. “2”を押してから“6”を押し、レンジリミット(チャンネル6)を入力します。

SELECT CHANNELS 2!1,2!2,2!6

- B. “-”を押してレンジセパレータを入力します。なお、次のチャンネルのスロット番号は自動的に入力されます。

SELECT CHANNELS 2!1,2!2,2!6-2!

- C. “9”を押してレンジリミットを入力します。

SELECT CHANNELS 2!1,2!2,2!6-2!9

チャンネルを開閉する

1. CLOSEを押し、チャンネルリスト中の指定されたチャンネルを閉じます。
2. OPENを押し、チャンネルリスト中の指定されたチャンネルを開きます。なお、OPEN ALLでは全チャンネルが開きます(指定されていないものも開きます)。

3.3.3 チャンネルのスキャン

以下の手順は、3.3.1項のイニシャルコンフィギュレーション手順が実行済みであることを前提とします。この初期設定により、下記のスキャン例のために40チャンネル・マルチプレクサシミュレータがスロット2に指定されます。

注

Model 7001は、スキャンリスト表示モード(“SCAN CHANNELS”プロンプトが表示される)になっていなければなりません。チャンネルステータスディスプレイは、SCAN LISTキーによって、“SELECT CHANNELS”(チャンネルリスト)と“SCAN CHANNELS”(スキャンリスト)をトグル切替えます。Model 7001がチャンネルステータス表示モードになっていない場合は、そのモードになるまでEXITキーを押します。

スキャンリストを作成する

以下のステップを実行することにより、チャンネル1から20までをスキャンするスキャンリストを作成します。

1. スキャンリストが現在空でない場合は、CLEAR LISTキーを押します。スキャンリストが空であることを示す下記のメッセージが表示されます。

SCAN CHANNELS

2. “2”を押してから“1”を押し、スロットおよびレンジリミットを入力します。

SCAN CHANNELS 2!1

3. “-”を押してレンジリミットのセパレータを入力します。なお、次のチャンネルのスロット番号は自動的に入力されます。

SCAN CHANNELS 2!1-2!

4. “2”を押してから“0”を押し、レンジリミットを入力します。

SCAN CHANNELS 2!1-2!20

5. ▷ またはENTERキーを押し、スキャンリストターミネータ (,) を入力します。

SCAN CHANNELS 2!1-2!20

注： スキャンリスト中に20チャンネルをレンジとして指定する代わりに、それぞれ別々に入力することも可能です (2!1, 2!2, 2!3, ... 2!20)。

マニュアルスキャン

RESETデフォルトでは、Model 7001は無数のチャンネルをマニュアルスキャンする構成となります。20番目のチャンネルが入力された後、スキャンリストの先頭 (チャンネル1) にラップアラウンドします。

- STEPキーを押し、Model 7001をアイドル状態から抜け出させます。ARMインジケータが点灯します。
- STEPキーを押し、最初のチャンネルをスキャンします。このチャンネルは、別のチャンネルがスキャンされるまで閉じたままです。
- STEPキーを押し、2番目のチャンネルをスキャンします。最初のチャンネルが開き、2番目のチャンネルが閉じます。
- STEPキーを押すたびに、前のチャンネルが開き、次のチャンネルが閉じます (ブレーク・ビフォア・メーク)。
- 終了したら、OPEN ALLキーを押し、スキャンを打ち切り、全チャンネルを開きます。Model 7001はアイドル状態になります (ARMインジケータ点灯)。

自動スキャン

以下のステップを実行することにより、20チャンネルを自動的にスキャンします。

- 閉じているチャンネルがあるか、またはARMインジケータが点灯しているときは、OPEN ALLキーを押します。
- SCAN CONFIGURATIONキーを押します。CONFIGURE SCANメニューが表示されます。

注

スペースに制約があるため、全部のメニュー項目を同時に表示させることはできません。矢印 (◀ または ▶) が表示されているときは、選択できるメニュー項目がほかにもまだあるということです。それらを表示させるには ◀ および ▷ カーソルキーを使います。

- ◀ および ▷ キーを使って、カーソルをCHAN-CONTROL上に位置付け、ENTERキーを押します。CHANNEL CONTROLメニューが表示されます。
- カーソルをCHANNEL-SPACING上に位置付け、ENTERキーを押します。SELECT CHAN SPACINGメニューが表示されます。

SELECT CHAN SPACING
TRIGLINK IMMEDIATE HOLD

- カーソルをIMMEDIATE上に位置付け、ENTERキーを押します。ディスプレイがCHANNEL CONTROLメニューに戻ります。
- EXITキーを2回押し、CONFIGURE SCANメニューから抜け出し、チャンネルステータスディスプレイに戻ります。
- CARD CONFIGURATIONキーを押します。CARD CONFIG MENUが表示されます。

CARD CONFIG MENU
DELAY READ-I/O-CARD

- カーソルをDELAY上に位置付け、ENTERキーを押します。SET DELAY FOR:メニューが表示されます。
- カーソルをSLOT-2上に位置付け、ENTERキーを押します。ディレイ時間 (秒単位) が表示されます。

(2)DELAY=00000.000

- キーパッドを使ってディレイを0.5秒に設定します。

(2)DELAY=00000.500

このディレイは、それぞれのチャンネルが閉じるたびに発生します。

- ENTERキーを押します。ディスプレイがSET DELAY FOR:メニューに戻ります。
- EXITキーを2回押し、チャンネルステータスディスプレイに戻ります。

13. スキャンを開始するには、STEPキーを押すだけです。
Model 7001がアイドル状態から抜け出し（ARMインジケータ点灯）、直ちに最初のチャンネルをスキャンします。0.5秒間のデレイ後、最初のチャンネルが開き、2番目のチャンネルが閉じます。この自動スキャンは0.5秒のスキャンレートで続きます。
14. 終了したら、OPEN ALLキーを押します。

TIMERコントロールスキャン

内部タイマを使うことで、スキャン間の時間間隔をコントロールすることが可能です。TIMERを使うことにより、上記の20チャンネルスキャンを修正し、それぞれのスキャンの開始から次のスキャンの開始までの間に20秒の時間間隔を置いたものにすることができます。

1. OPEN ALLキーを押し、Model 7001をアイドル状態に維持させます。
2. SCAN CONFIGURATIONキーを押します。CONFIGURE SCANメニューが表示されます。
3. カーソルをSCAN-CONTROL上に位置付け、ENTERキーを押します。SCAN CONTROLメニューが表示されます。
4. カーソルをSCAN-SPACING上に位置付け、ENTERキーを押します。SELECT SCAN SPACINGメニューが表示されます。

SELECT SCAN SPACING
TIMER EXTERNAL GPIB MANUAL

5. カーソルをTIMER上に位置付け、ENTERキーを押します。タイマ時間間隔（秒単位）が表示されます。

INTERVAL=00000.001

6. キーパッドを使って20秒の時間間隔をキーインします。

INTERVAL=00020.000

7. ENTERキーを押します。ディスプレイがSCAN CONTROLメニューに戻ります。
8. EXITキーを2回押し、CONFIGURE SCANメニューから抜け出し、チャンネルステータスディスプレイに戻ります。
9. スキャンを開始するためにSTEPキーを押します。直ちに最初のスキャンが開始され、20番目のチャンネルが閉じて終了となります。さらに10秒経過した後（スキャンが開始されてから20秒後）、次のスキャンが開始されます。

10. 終了したら、OPEN ALLキーを押します。

3.4 スキャンプロセスのあらまし

以下の説明は、スキャンの基本原則を知っていただくためのものです。強化された機能の細かな点については煩雑となるので省略してあります。3.5項（フロントパネル）と3.6項（IEEE-488バス）の操作例に対する基礎知識としてはこれで十分です。スキャンプロセスの詳細な説明は、4.5.4項に記載してありますから参照してください。

スキャンオペレーションを単純化したモデルを図3-6に示します。図示のように、スキャンオペレーションは、アームレイヤとスキャンレイヤとチャンネルレイヤの3つのレイヤで構成されています。

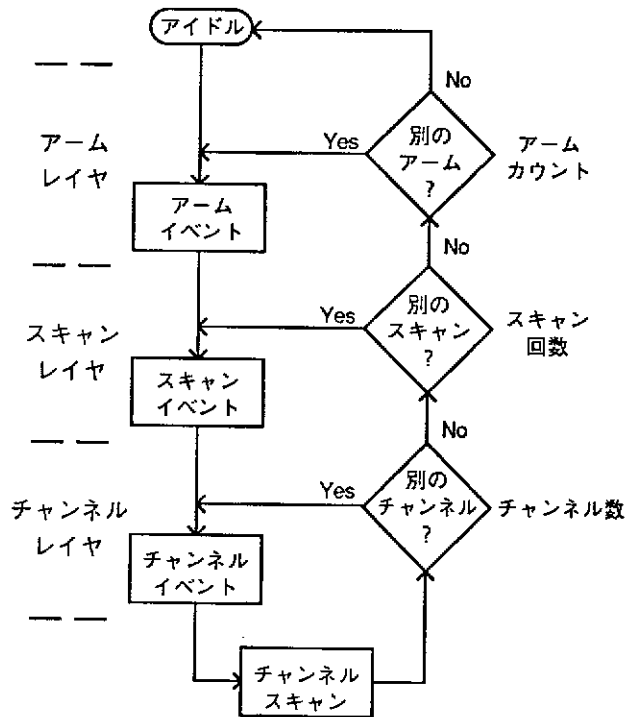


図3-6
スキャンオペレーションの簡略モデル

アイドル

Model 7001は、モデルのレイヤのどれか1つの中で動作していない限り、アイドル状態にあるとみなされます。Model 7001がアイドル状態のときは、フロントパネルのARMインジケータは消えています。

STEPを押すことによって（あるいはIEEE-488バスを通じて：INITまたは：INIT*CON ONコマンドを送信することによって）、Model 7001をアイドル状態から抜け出させると、ARMインジケータが点灯し、オペレーションがアームレイヤに入ります。

アームレイヤ

原則として、Model 7001では、オペレーションをスキャンレイヤに進ませるにはアームイベントが必要です。Immediateアームスペーシングが選択されている場合には、Model 7001がアイドル状態から抜け出ると、直ちにオペレーションが次のレイヤに進みます。他のアームスペーシングイベントのどれか1つが選択されている場合には、Model 7001は該当するイベントが発生するまで待ちます。

Manualアームスペーシングが選択されている場合には、フロントパネルのSTEPキーが押されるまで待ちます。 GPIBアームスペーシングが選択されている場合には、バストリガ（GETまたは*TRG）が受信されるまで待ちます。 Externalアームスペーシングが選択されている場合には、入力トリガ（リヤパネル上のEXTERNAL TRIGGERコネクタ経由）が受信されるまで待ちます。 Triger Linkアームスペーシングが選択されている場合には、入力トリガが（TRIGGER LINKを介して）受信されるまで待ちます。

他の全てのスキャンオペレーションが完了した後、Model 7001を付加アームにプログラムすることによってアームレイヤに戻すことができます。アームカウントは、有限値（1～9999）に設定することも無限に設定することもできます。

Model 7001がアームレイヤから出た後、オペレーションはスキャンレイヤに入ります。

スキャンレイヤ

原則として、Model 7001では、オペレーションをチャンネルレイヤに進ませるにはスキャンイベントが必要です。Immediateスキャンスペーシングが選択されている場合には、直ちにオペレーションが次のレイヤに進みます。

トリガスキャンスペーシングイベント（Timer、External、GPIB、Manual、Trigger Link）のどれか1つが選択されている場合には、該当するイベントが発生するまで待ちます。Timerスキャンスペーシングが選択されている場合には、直ちに1回目のバス（スキャンレイヤ通過）が生じます。付加スキャンにプログラムされている場合には、タイマがタイムアウトとなるまで待ちます。タイマは、1msecから99999.999秒までの時間間隔に設定することができます。

スキャンカウント（スキャン回数）は、有限値（1～9999）に設定することも無限に設定することもできます。

Model 7001がスキャンレイヤから出た後、オペレーションはチャンネルレイヤに入ります。

チャンネルレイヤ

原則として、チャンネルスキャンレートはチャンネルイベントによってコントロールされます。Immediateチャンネルスペーシングが選択されている場合には、直ちにチャンネルがスキャンされます。他のチャンネルスペーシングイベント（Timer、External、GPIB、ManualまたはTrigger Link）のどれか1つが選択されている場合には、該当するイベントが発生するのを待ち、それからチャンネルがスキャンされます。Timerチャンネルスペーシングが選択されている場合には、直ちに最初のチャンネルがスキャンされます。付加チャンネルごとに、それぞれ、タイマがタイムアウトとなるまで待ち、それからスキャンされます。

チャンネルカウント（スキャンするチャンネルの数）は、スキャンリストの中に入っているチャンネルの数（スキャンリスト長）に設定するのが普通です。ただし、チャンネルカウントは、有限値（1～9999）に設定することも無限に設定することもできます。

3.5 イニシャルコンフィギュレーション

警告

Model 7001の電源を投入する前に、用意されている電源コードまたは同等品で、アースされたコンセントに接続されていることを確認してください。適切にアースすることを怠ると、感電の危険があり、傷害や死亡につながる恐れがあります。

基本操作（チャンネルの開閉またはスキャン）を実行する前に、まず、いくつかのイニシャルコンフィギュレーションについてみておく必要があります。今、Model 701Xシリーズのカード（すなわち、Model 7011、7012、7013）をお使いで、それらが既に（第2節の説明に従って）メインフレーム内に装着されている場合は、以下のコンフィギュレーション情報は読み飛ばし、3.5項（フロントパネル操作）と3.6項（IEEE-488バス操作）にお進みください。

3.5.1 スイッチングカードシミュレータ

スイッチングカードシミュレータを使えば、スイッチングカードを装着しないで機器操作を行なうことができます。適切なスイッチングカードシミュレータを選択することにより、Model 7001は、40チャンネル・スイッチングカードまたは4x10マトリックスカードが装着されているかのよう動作します。カードを装着しないでテストプログラムを開発したいときに便利です。

スロットに型番9990が指定されると、メインフレームは40チャンネル・スイッチングカードが装着されているかのように動作します。また、スロットに型番9991が指定されると、4x10マトリックスカードが装着されているかのように動作します。これらのシミュレータ型番を指定するには、CARD CONFIG MENUでTYPE（カードタイプ）を設定します（3.5.3項参照）。シミュレータを使って3.6項の操作例を実行する場合は、スロット1を非マトリックススイッチング（9990）、スロット2をマトリックススイッチング（9991）にそれぞれ設定してください。

注

Model 701Xシリーズのカードが既にスロットに装着されている場合、スロットにシミュレータを指定することはできません。シミュレータを使うには、Model 7001の電源を切り、スロットからカードを取り出してください。

3.5.2 スイッチングカードの装着

メインフレーム内にスイッチングカードを装着する手順は第2節に説明してあります。前項の説明に従ってスイッチングカードシミュレータを使用するときは、スロットを空のままにしておいてください。

3.5.3 カードタイプ

Model 7001が正しく動作するには、Model 7001は2つのスロットのそれぞれにどのタイプのカードが装着されているかを知っていなければなりません。Model 701Xシリーズのカード（Model 7011、7012、7013など）がスロットに装着されている場合、メインフレームは電源投入時にカードタイプを自動的に識別し、正しい動作が可能な設定になります。したがって、メインフレーム内にModel 701Xシリーズのカードが装着されているときは、本項の手順は無視し、フロントパネル基本操作（3.5項）に進んで構いません。

他のカードタイプ（701Xシリーズ以外のカード）およびシミュレータを使う場合、ユーザは正しい型番をスロットに指定しなければなりません。いったんユーザによって型番が指定されると、Model 701Xシリーズのカードが装着されない限り、その型番は次の電源投入時に記憶されています。なお、スロットを空に指定することもできます。この場合、そのスロットのフロントパネルステータスディスプレイは見えなくなります。

以下の手順を実行することにより、メインフレームスロットのどちらか一方または両方に型番を指定します。

1. Model 7001の電源を投入します。電源投入シーケンス中に、Model 7001は、現在どのModel 701Xシリーズカードが装着されているかを、または前回のスロット指定を一時的に表示します。

例 —スロット1に対してModel 7013カードが指定され、前回、スロット2が空に指定されていた場合には、電源投入シーケンス中に下記のメッセージが一時的に表示されます。

```
#1=7013      #2=NONE
```

スロット2が空で、それにシミュレータを指定するつもりがない場合は、ここでストップし、フロントパネル基本操作（3.5項）に進みます。スロット2に701Xシリーズ以外のカード（すなわち、Model 7156）が装着されている場合には、この手順を続行して正しい型番をスロットに指定します。

2. CARD CONFIGURATIONキーを押して下記のメッセージを表示させます。

```
CARD CONFIG MENU
TYPE          #-OF-POLES  CARD-PAIR ►
◀ DERAY      READ-I/O-CARD
```

注

矢印（◀または▶）が表示されているときは、選択できるメニュー項目がほかにもまだあるということです。それらを表示させるには◀および▶カーソルキーを使います。

3. ◀および▶カーソルキーを使って、カーソル（点滅しているメニュー項目）をTYPE上に位置付け、ENTERキーを押します。下記のメッセージが表示されます。

SET CARD TYPE
SLOT-1 SLOT-2

4. 指定する必要があるスロット上にカーソルを位置付け、ENTERキーを押します。たとえば、スロット1を選択した場合、下記の標準メッセージが表示されます。

SLOT-1 CARD:NONE

このメッセージは、スロット1にはModel 701Xシリーズのカードが装着されておらず、前回、スロット1が空に指定されていたことを意味します。

5. 以下のステップのどれか1つを実行します。
- A. 空スロット — スロットを空に指定したければ、カーソルキー（◀および▶）を使って“NONE”メッセージを表示させます。たとえば、

SLOT-1 CARD:NONE

このメッセージは、ENTERキーを押すと（ステップ6）、スロット1が空スロットとして動作する設定になることを意味します。

- B. シミュレータ — 40チャンネルスイッチングカードをシミュレートするようにスロット指定したければ、カーソルキー（◀および▶）を使って型番9990を表示させます。4x10マトリックスカードをシミュレートするようにスロット指定したければ、カーソルキーを使って型番9991を表示させます。たとえば、

SLOT-1 CARD:9990

このメッセージは、ENTERキーを押すと（ステップ6）、スロットが40チャンネルスイッチングカードとして動作する設定になることを意味します。

- C. 701Xシリーズ以外のカード — スロットを701Xシリーズ以外のカードに指定するには、カーソルキー（◀および▶）を使ってそのカードの型番を表示させます。たとえば、スロット1にModel 7156を装着する場合、カーソルキーを使って下記のメッセージを表示させます。

SLOT-1 CARD:7156

このメッセージは、ENTERキーを押すと（ステップ6）、スロット1がModel 7156装着状態で正しく動作することを意味します。

6. 適切なスロット指定メッセージが表示されている状態で、ENTERキーを押してスロット指定を実行させます。ディスプレイがSET CARD TYPEメニューに戻ります。
7. 他のスロットを指定するときは、ステップ4~6を繰り返します。
8. CARD CONFIG MENUから抜け出すために、EXITキーを2回押します。

注

EXITキーはCARD CONFIG MENU構造の中のどこでも使用できます。EXITキーを押すたびに、ディスプレイは前のメニューレベルに戻り、そのとき保留中のオペレーションは実行されません。

3.6 フロントパネル操作

以下に、2つの例を使って基本的なフロントパネル操作をデモ実行します。一方の例では開閉操作をデモ実行し、他方の例ではスキャンをデモ実行します。なお、これらの例は基本操作を知っていただくためのものにすぎません。フロントパネルからModel 7001を操作する場合の各種機能の使い方については第4節で詳しく説明します。

注

以下の操作例は、Model 7001が3.5項の説明に従って正しく動作するように初期設定されていることを前提とします。

3.6.1 開閉操作例

Model 7001の基本ファンクションの1つとして、指定された1つまたはそれ以上のチャンネルを開閉させるだけのものがあります。以下の例はこの操作をデモ実行するものです。

この操作例は、非マトリックススイッチングカードまたはマトリックスカードあるいはその両方が装着されている場合に実行できます。カードシミュレータを使う場合は、3.5.3項で説明したように、スロット1を非マトリックススイッチング (9990)、スロット2をマトリックススイッチング (9991) にそれぞれ設定してください。

この例でコントロールされるチャンネルは次の通りです。

スイッチングカードチャンネル： 1、3、4、5、6
 マトリックスカードチャンネル： 行1列1、行2列2、
 行2列3、行2列4、行2列5、行2列6。

以下のステップを実行することにより、非マトリックススイッチングカードおよび/またはマトリックスカードのチャンネルを開閉させます。

ステップ1 Model 7001をチャンネルリスト表示状態にする

チャンネルステータス表示状態になっていない場合は、EXITキーを1回または複数回押します。チャンネルステータス表示状態になったら、SCANキーで、チャンネルリスト (SELECT CHANNELS) とスキャンリスト (SCAN CHANNELS) をトグル切替えます。

SELECT CHANNELS

現在、チャンネルリストが表示されている場合は、CLEAR LISTキーを押してそれをキャンセルします。また、現在、閉じているチャンネルがあれば、OPEN ALLキーを押してそれらを開きます。

ステップ2 チャンネルリストを入力する

スイッチングカード — 非マトリックススイッチングカードの場合、スロット1に対してチャンネルリスト (チャンネル1、3、4、5、6) を以下の手順で入力します。

注

以下の手順は、スロット1が非マトリックススイッチングに設定されていることを前提とします。

スロット2を非マトリックススイッチングに使う場合は、ステップ1および2の (A) を変更してください。すなわち、“1”を押す操作を“2”を押す操作に変えてください。

1. シングルチャンネル入力法を使って最初のチャンネル (チャンネル1) を入力します。
 - A. “1”を押す。 スロットを入力。
 - B. “1”を押す。 チャンネルを入力。
 - C. “▷”または“ENTER”を押す。チャンネルセパレータ (カンマ) を入力。

SELECT CHANNELS 1!1,

2. チャンネルレンジ入力法を使って残りのチャンネル (チャンネル3~6) を入力します。
 - A. “1”を押す。 スロットを入力。
 - B. “3”を押す。 最初のチャンネルを入力。
 - C. “-”を押す。 レンジ指示子とスロットを入力。
 - D. “6”を押す。 最後のチャンネルを入力。
 - E. “▷”または“ENTER”を押す。チャンネルリスターミネータ (カンマ) を入力。

SELECT CHANNELS 1!1, 1!3-1!6,

注：チャンネル3、4、5、6は、ステップ1に示したシングルチャンネル入力法を使って入力することも可能です。

マトリックスカード — マトリックスカードの場合、チャンネルリスト (スロット2に対して行1列1、行2列2、行2列3、行2列4、行2列5) を以下の手順で入力します。

注：以下の手順は、スロット2がマトリックススイッチングに設定されていることを前提とします。スロット1をマトリックススイッチングに使う場合は、ステップ1および2の (A) を変更してください。すなわち、“1”を押す操作を“2”を押す操作に変えてください。

1. シングルチャンネル入力法を使って最初のチャンネル (行1列1) を入力します。
 - A. “2”を押す。 スロットを入力。
 - B. “1”を押す。 行を入力。
 - C. “1”を押す。 列を入力。
 - D. “▷”または“ENTER”を押す。チャンネルセパレータ (カンマ) を入力。

SELECT CHANNELS 2!1!1,

2. チャンネルレンジ入力法を使って残りのチャンネル（行2列2～行2列5）を入力します。
- A. “2” を押す。 スロットを入力。
 - B. “2” を押す。 最初のチャンネルの行を入力。
 - C. “2” を押す。 最初のチャンネルの列を入力。
 - D. “-” を押す。 レンジ指示子を入力。
 - E. “2” を押す。 最後のチャンネルの行を入力。
 - F. “5” を押す。 最後のチャンネルの列を入力。
 - G. “▷” または “ENTER” を押す。チャンネルリストターミネータ（カンマ）を入力。

SELECT CHANNELS 211!1, 212!2-2!2!5

注：行2のチャンネルは、ステップ1に示したシングルチャンネル入力法を使って入力することもできます。

ステップ3 リスト中のチャンネルを閉じる（または開く）

1. チャンネルリスト中に指定されているチャンネルを閉じるには、CLOSEキーを押すだけです。閉じたチャンネルは、チャンネルステータスディスプレイ上に表示されます。
2. チャンネルリスト中に指定されている閉じたチャンネルを開くには、OPENキーを押します。チャンネルリスト中に指定されているチャンネルだけしか開きませんが、注意してください。指定されていないチャンネル（チャンネルリスト中のチャンネルも含めて）を開くには、OPEN ALLキーを押します。

付加チャンネルを閉じる

チャンネルリスト中のチャンネルを閉じた後、そのほかに1つまたはそれ以上のチャンネルを閉じたい場合もあります。付加チャンネルを閉じるには、そのチャンネルをチャンネルリストに入力し、“CLOSE”を押すだけです。たとえば、チャンネルリストが次のようなものであるとしてみましよう。

SELECT CHANNELS 1!1, 1!3-1!6,

チャンネル1!8を閉じるには、カーソルをチャンネルリストの終わりに位置付け、“1”、“8”を押し、CLOSEキーを押します。これでチャンネルリストは次のようなものになります。

SELECT CHANNELS 1!1, 1!3-1!6, 1!8,

付加チャンネルの閉じ方としては、チャンネルリストをクリアしてから（CLEAR LISTキーを押してから）、チャンネルを入力し、閉じることも可能です。ただし、いったんチャンネルリストをクリアすると、リコールすることはできませんから注意してください。この場合、リスト全体を再入力しなければなりません。

閉じたチャンネルを開く

前に説明したように、OPENキーを押すと、チャンネルリスト中に指定されている全チャンネルが開きます。全チャンネルではなく1つまたはそれ以上のチャンネルを開きたい場合は、CLEAR LISTキーを押してチャンネルリストをクリアしてから、開きたい閉チャンネルを指定する新しいチャンネルリストを入力する必要があります。新しいチャンネルリストが入力されたら、OPENキーを押してそれらのチャンネルを入力するだけです。

3.6.2 スキャン操作例

Model 7001は、指定されたチャンネルのリストをスキャンすることができます。チャンネルをスキャンする順序は、スキャンリスト中のチャンネルの順序によって決まります。以下の例はこの操作をデモ実行するものです。

以下の操作例は、非マトリックススイッチングカードまたはマトリックスカードあるいはその両方が装着されている場合に実行できます。カードシミュレータを使う場合は、3.5.3項で説明したように、スロット1を非マトリックススイッチング（9990）、スロット2をマトリックススイッチング（9991）にそれぞれ設定してください。

この例でスキャンされるチャンネルは次の通りです。

スイッチングカードチャンネル：1、3、4、5、6
マトリックスカードチャンネル：行1列1、行2列2、
行2列3、行2列4、行2列5、行2列6。

以下のステップを実行することにより、非マトリックススイッチングカードおよび/またはマトリックスカードのチャンネルをスキャンします。指定されたスキャンリストについて3回のスキャンが行なわれます。

ステップ1 Model 7001をRESETする

RESETを行なうことにより、Model 7001は、スキャンをコントロールするのに通常使用されるデフォルトの設定になります。

- MENUキーを押します。下記のメニューが表示されます。

```

MAIM MENU
SAVESETUP  GPIB          DIGITAL I/O ▶
◀ TEST      LANGUAGE    GENERAL
    
```

- カーソルキー（◀ および ▶）を使ってカーソル（点滅しているメニュー項目）をSAVESETUP上に位置付け、ENTERキーを押します。下記のメニューが表示されます。

```

SETUP MENU
SAVE  RESTORE  PWRON  RESET
    
```

- カーソルをRESET上に位置付け、ENTERキーを押します。下記のメッセージが表示されます。

```

RESETTING INSTRUMENT
    
```

- ENTERキーを押します。下記のメッセージが表示されます。

```

RESETTING COMPLETE
    
```

- 再びENTERキーを押し、SETUP MENUに戻ります。
- EXITキーを2回押し、メニュー構造から抜け出ます。

ステップ2 Model 7001をスキャンリスト表示状態にする

Model 7001がチャンネルリスト表示状態になっているとき、SCAN LISTキーを押してスキャンリスト表示状態を選択します。下記のメッセージプロンプトが表示されます。

```

SCAN CHANNELS
    
```

現在、スキャンリストが表示されている場合は、CLEAR LISTキーを押してそれをキャンセルします。また、現在、閉じているチャンネルがあれば、OPEN ALLキーを押してそれらを開きます。

ステップ2 スキャンリストを入力する

スイッチングカード — 非マトリックススイッチングカードの場合、スロット1に対してスキャンリスト（チャンネル1、3、4、5、6）を以下の手順で入力します。

注：以下の手順は、スロット1が非マトリックススイッチングに設定されていることを前提とします。スロット2を非マトリックススイッチングに使う場合は、ステップ1および2の（A）を変更してください。すなわち、“1”を押す操作を“2”を押す操作に変えてください。

- シングルチャンネル入力法を使って最初のチャンネル（チャンネル1）を入力します。
 - “1”を押す。 スロットを入力。
 - “1”を押す。 チャンネルを入力。
 - “▷”または“ENTER”を押す。チャンネルセパレータ（カンマ）を入力。

```

SCAN CHANNELS 1!1,
    
```

- チャンネルレンジ入力法を使って残りのチャンネル（チャンネル3～6）を入力します。
 - “1”を押す。 スロットを入力。
 - “3”を押す。 最初のチャンネルを入力。
 - “-”を押す。 レンジ指示子とスロットを入力。
 - “6”を押す。 最後のチャンネルを入力。
 - “▷”または“ENTER”を押す。チャンネルリストターミネータ（カンマ）を入力。

```

SCAN CHANNELS 1!1, 1!3-1!6,
    
```

注：チャンネル3、4、5、6は、ステップ1に示したシングルチャンネル入力法を使って入力することも可能です。

マトリックスカード — マトリックスカードの場合、スキャンリスト（スロット2に対して行1列1、行2列2、行2列3、行2列4、行2列5）を以下の手順で入力します。

注：以下の手順は、スロット2がマトリックススイッチングに設定されていることを前提とします。スロット1をマトリックススイッチングに使う場合は、ステップ1および2の（A）を変更してください。すなわち、“1”を押す操作を“2”を押す操作に変えてください。

- シングルチャンネル入力法を使って最初のチャンネル（行1列1）を入力します。
 - “2”を押す。 スロットを入力。
 - “1”を押す。 行を入力。
 - “1”を押す。 列を入力。
 - “▷”または“ENTER”を押す。チャンネルセパレータ（カンマ）を入力。

```

SCAN CHANNELS 2!1!1,
    
```

- チャンネルレンジ入力法を使って残りのチャンネル（行2列2～行2列5）を入力します。
 - “2”を押す。 スロットを入力。
 - “2”を押す。 最初のチャンネルの行を入力。

- C. “2”を押す 最初のチャンネルの列を入力。
- D. “-”を押す。 レンジ指示子とスロットを入力。
- E. “2”を押す。 最後のチャンネルの行を入力。
- F. “5”を押す。 最後のチャンネルの列を入力。
- G. “ ”または“ENTER”を押す。チャンネルリストターミネータ（カンマ）を入力。

SCAN CHANNELS 2!1!1, 2!2!2-2!2!5,

注：行2のチャンネルは、ステップ1に示したシングルチャンネル入力法を使って入力することもできます。

ステップ4 スキャンのチャンネルレイヤを設定する

注

RESET（ステップ1）を行なったとき、チャンネルレイヤはMANUALチャンネルスキャンにリセットされました。MANUALでは、初めてSTEPを押すと、Model 7001がアイドル状態から抜け出します。それ以後、STEPを押すたびに、マニュアルでスキャンをステップスルーします。実際に試してみてください。終わったら、OPEN ALLを押し、Model 7001をアイドル状態にします。

この例では、チャンネルスキャンレートはタイマによってコントロールされます。原則として、タイマがタイムアウトになるたびに、次のチャンネルが選択されます。

Model 7001をRESETした後は、プログラムする必要があるのはCHANNEL SPACINGだけです。

以下の手順でチャンネルレイヤを設定します。

1. SCAN CONFIGURATIONを押し、下記のメニューを表示させます。

```

CONFIGURE SCAN
CHAN-CONTROL  SCAN-CONTROL ►
◀ ARM-CONTROL  CHAN-RESTRICTIONS
    
```

2. カーソルキー（◀および▶）を使ってカーソルをCHAN-CONTROL（チャンネルレイヤ）上に位置付け、ENTERキーを押します。下記のメニューが表示されず。

```

CHANNEL CONTROL
CHANNEL-SPACING  NUMBER-OF-CHANS ►
◀ CONTROL
    
```

3. カーソルをCHAN-SPACING上に位置付け、ENTERキーを押します。下記のメニューが表示されます。

```

SELECT CHAN SPACING
TIMER          EXTERNAL GPIB  MANUAL ►
◀ TRIGLINK  IMMEDIATE HOLD
    
```

4. カーソルをTIMER上に位置付け、ENTERキーを押します。現在のタイマ時間間隔（秒単位）が表示されます。

INTERVAL=00000.001

このメッセージは、現在、タイマが1msecにセットされていることを意味します。

5. カーソルキーと数字キーを使ってタイマを0.5秒の時間間隔にセットします。タイマ時間間隔が次のように表示されるはずですが。

INTERVAL=00000.500

6. ENTERキーを押し、チャンネルスキャンをコントロールするイベントとしてタイマを選択します。ディスプレイが下記のメニューに戻ります。

```

CHANNEL CONTROL
CHANNEL-SPACING  NUMBER-OF-CHANS ►
◀ CONTROL
    
```

注：Model 7001をRESETしたとき、スキャンするチャンネルの数はスキャンリスト長にセットされました。これがこの例でスキャンしたいチャンネル数です。

7. EXITを押し、CONFIGURE SCANメニューに戻ります。

ステップ5 スキャンのスキャンレイヤを設定する

RESET（ステップ1）を行なったとき、スキャンレイヤはIMMEDIATEスキャンスペーシングにリセットされました。これがこの例に適したスキャンスペーシングです。IMMEDIATEスキャンスペーシングとは、プログラムされた各スキャンが即時に行なわれ、イベントが発生するのを待たないということです。

また、スキャンレイヤは、無限回数のスキャンを行なうようにRESETされました。この例ではスキャンを3回行なうだけでよいので、NUMBER OF SCANを次のようにして変更します。

1. 現在、下記のメッセージが表示されているはずで

```
CONFIGURE SCAN
CHAN-CONTROL  SCAN-CONTROL ►
◀ ARM-CONTROL  CHAN-RESTRICTIONS
```

2. カーソルをSCAN-CONTROL上に位置付け、ENTERキーを押します。下記のメッセージが表示されます。

```
SCAN CONTROL
SCAN SPACING  NUMBER-OF-SCANS ►
◀ CONTROL
```

3. カーソルをNUMBER-OF-SCANS上に位置付け、ENTERキーを押します。下記のメッセージが表示されます。

```
NUMBER OF SCANS
INFINITE  ENTER-SCAN-COUNT
```

4. カーソルをENTER-SCAN-COUNT上に位置付け、ENTERキーを押します。スキャンカウントを示す下記の標準メッセージが表示されます。

```
SCAN COUNT=0000
```

スキャンカウントがゼロであるということは、現在、スキャンレイヤが無限回数のスキャンに設定されているということです。

5. 3のスキャンカウントをキーインします。

```
SCAN COUNT=0003
```

6. Model 7001を3回のスキャンに設定するために、ENTERキーを押します。ディスプレイがSCAN CONTROLメニューに戻ります。

7. EXITキーを2回押し、メニュー構造から抜け出ます。

注：RESET（ステップ1）を行なうことで、アームレイヤはIMMEDIATEアームおよび1のアームカウントに設定されました。これらのアーム条件はどちらもこの例に適したものです。したがって、アームレイヤ設定は必要ありません。

ステップ6 スキャンを実行する

スキャンを実行するには、STEPキーを押すだけです。ディスプレイ上で、スキャンがアームされ（ARMインジケータ点灯）、メインフレームがスキャンリストのチャンネルを3回スキャンします。3回目のスキャンが終了すると、Model 7001はアイドル状態に戻ります（ARMインジケータが消えます）。ただし、スキャンリスト中の最後のチャンネルは閉じたままです。

注

スキャンリスト中の最後のチャンネルは閉じたままです。テストのとき、スキャン後に最後のチャンネルを開く必要がある場合は、次のステップに進んでください。

ステップ7 最後のチャンネルを開く

3回目のスキャンの終了時に最後のチャンネルを開かせるには、メモリにストアされているブランクチャンネルパターン（全チャンネル開）をスキャンリスト中への最後のチャンネル入力として指定する必要があります。

ブランクチャンネルパターンをメモリにセーブするには、以下のステップを実行します。

1. チャンネルリストを表示させ（SELECT CHANNEL）、OPEN ALLキーを押して全チャンネルを開きます。
2. STOREキーを押します。下記の標準メッセージが表示されます。

```
STORE CHANS AT #001
```

3. ブランクチャンネルパターンをメモリロケーション1にストアするために、ENTERキーを押します。なお、ブランクチャンネルパターンは100のメモリロケーションの任意の1つにストアすることもできます。そうするには、目的の値をキーインし、ENTERキーを押します。

ブランクチャンネルパターンがメモリロケーション1（M1）にストアされている状態で、それを次のようにしてスキャンリストに加えます。

1. スキャンリストを表示させます（必要あれば、SCAN LISTキーを押します）。
2. ▷ カーソルキーを使ってカーソルをスキャンリストの終わりに位置付けます。
3. “M”、“1”を押してから ▷ またはENTERキーを押すことにより、ブランクチャンネルパターンを入力します。

SCAN CHANNELS 1!1, 1!3-1!6, 2!1!1, 2!2!2-2!2!5, M1,

再びSTEPキーを押してスキャンを繰り返してください。
3回目のスキャンが終了すると、最後のチャンネル（上記に
スキャンリストを使用している場合、チャンネル2!2!5）が
開きます。

3.7 IEEE-488.2およびSCPIの 基礎事項

以下の各項では、IEEE-488バスを介しての操作に関する基礎的事項について説明します。Model 7001のバス操作の詳細は第5節に記載してあります。

SCPIの概要

IEEE-488. 2には、機器へあるいは機器からデータを送信するためのシンタックスが定義されています。また、ステータスレジスタの読取り、トリガの発生、デフォルト状態へのリセットなどの基本的な操作を実行するのに使われるコマンドも定義されています。

SCPI (Standard Commands for Programmable Instruments) には、その他機器操作に関する一切をコントロールするための標準コマンドセットが定義されています。IEEE-488.2とSCPIで、あらゆるプログラマブル機器のためのコマンド構造を構築できます。

互換性

IEEE-488.2規格とSCPIを採用するからといって、ハードウェアに関して何か特別なことが必要になることはありません。旧規格で使っていたIEEE-488インタフェースは新規格でも動作します。Model 7001は、IEEE-488インタフェースを装備しているコンピュータに接続するだけでよいのです。

注

本書ならびにModel 7001のメニュー構造では、 GPIB（汎用インタフェースバス）という用語が使われていますが、GPIBはIEEE-488バスの別名にすぎません。

3.7.1 バスの接続

Model 7001をバスを介して使用するには、その前に、リヤパネル上のIEEE-488コネクタをユーザ側のコントローラのIEEE-488コネクタに接続しなければなりません。この接続には、Keithley Model 7007または同種のIEEE-488ケーブルを使用してください。

3.7.2 一次アドレス

Model 7001の一次アドレスは、コントローラのプログラム言語で指定しようとする一次アドレスと一致していなければなりません。一次アドレスの出荷時設定は7ですから、そのアドレスを使うのであれば、変更する必要はありません。ただし、MENUキーで表示させることができるGPIBセットアップメニューを使うこと（ADDRESS選択）によって、一次アドレスを0から30までの任意の値に設定することができます。

3.7.3 略記コモンコマンドサマリ

IEEE-488.2規格によって定義されているコモンコマンドの略記リストを表3-1に示します。これらは、主としてバス操作に使われるコモンコマンドです。なお、各コモンコマンドの前には星印（*）が付いていますから、注意してください。

3.7.4 略記SCPIコマンドサマリ

ほとんどの機器操作はSCPIコマンドを使ってコントロールされます。いくつかの基本操作を実行するのに必要なSCPIコマンドの略記リストを表3-2に示します。

この表には、:ARMおよび:TRIGgerサブシステムコマンドについては付随の問合せコマンド

(:COUNt?、:DELay?、:SOURce?、:TIMer?) は示されていません。たとえば、:TRIGger:SOURce?問合せコマンドは、現在選択されているコントロールソースを要求するのに使います。この問合せコマンドが送信され、Model 7001がトークにアドレッシングされた後、選択されているコントロールソースを確認するメッセージがコンピュータに送られます。

表3-1

略記コモンコマンドサマリ

ニモニック	名前	説明
*CLS	Clear status	エラーキューおよびイベントレジスタをクリアする。
*RST	Resct	7001を*RSTデフォルト (表5-6参照) に戻す。
*TRG	Trigger	バストリガを発生する (グループ実行トリガコマンドFGETと同じ)。
*SAV<n>	Save	現在のセットアップコンフィギュレーションをメモリにセーブする (n=0~9)。
*RCL<n>	Recall	7001をメモリにストアされているセットアップコンフィギュレーションに戻す (n=0~9)。

表3-2

略記SCPIコマンドサマリ

<pre> :SYSTem :PRESet [:ROUte] :CLOSe<list> :STATe? :OPEN<list> ALL :SCAN<list> :INITiate :ABORt :ARM :LAYer2 :COUNt<n> INF :DELay<num> :SOURce HOLD IMMEDIATE TIMer MANual BUS TLINk EXTernal :TIMer<num> :TRIGger :COUNt<n> INF :AUTo ON/OFF :DELay<num> :SOURce HOLD IMMEDIATE MANual BUS TLINk EXTernal TIMer :TIMer<num> </pre>	<p>サブシステムコマンドパス。 スキャンをデフォルトコンフィギュレーション (表5-6参照) に設定する。</p> <p>サブシステムコマンドパス。 指定されたチャンネルを閉じるためのパスおよびコマンド: 閉じているチャンネルを要求する。 指定された (または全部の) チャンネルを開く。 チャンネルリストを指定するためのパスおよびコマンド:</p> <p>1トリガ (スキャン) サイクルを開始させる。 トリガシステムをリセットし、アイドル状態に入る。 スキャンを設定するためのサブシステムコマンドパス: スキャンレイヤをプログラムするためのパス: スキャン回数 (1~9999またはINFinite) をプログラムする。 デイレイ (0~99999.999秒) をプログラムする。 スキャンスペーシングをコントロールするためのイベントを選択する。</p> <p>タイマ時間間隔 (0.001~99999.999秒) を設定する。</p> <p>チャンネルレイヤをプログラムするためのサブシステムコマンドパス: チャンネル数 (1~9999またはINFinite) をプログラムする。 自動スキャンカウントをイネーブル/ディスエーブルする。 デイレイ (0~99999.999秒) をプログラムする。 チャンネルスペーシングをコントロールするためのイベントを選択する。</p> <p>タイマ時間間隔 (0.001~99999.999秒) を設定する。</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

注: コマンド短縮形は大文字によって表わされます。たとえば、“:arm:layer2:source:immediate”を送信する代わりに、“:arm:lay2:sour:imm”を送信することができます。

3.7.5 シンタクスルール〔構文規則〕

一般形式

SCPIコマンドの一般形式は表3-2に示してあります。これらのコマンドは階層的性格をもっており、ルートコマンドから始まります。たとえば、全チャンネルを開くには、下記のコマンドを送信します。

```
:OPEN ALL
```

この例でのルートコマンドは:ROUTEです。これはオプションコマンドワード(表中、大括弧で囲んで示す)で、使う必要はありません。

コマンドワードの前にはコロン(:)が付きます。また、コマンドとパラメータの間にはスペースがなければなりません。上記の例では、:OPENコマンドワードとALLパラメータの間にスペースがあります。

SCPIコマンドワードおよびコモンコマンドはケースセンシティブではありません。大文字でも小文字でも送信できます。表3-2のコマンドは、大文字と小文字を組み合わせた形で示してあります。大文字はコマンドの短縮形バージョンを表わしています。たとえば、以下に同一コマンドのバージョンをいくつか示しますが、これらはすべて有効です。

```
:ARM:LAYER2:SOURCE MANUAL
:ARM:LAY2:SOUR MAN
:arm:layer2:source manual
:arm:lay2:sour man
:ARM:LAY2:Sour Man
:ArM;LaY2;SouR MaN
```

パラメータ

前述のように、パラメータはスペースによってコマンドワードから区切られます。パラメータは、チャンネルリスト、実数、整数、ストリング、名前あるいはブール値などの1つまたはそれ以上のデータタイプで構成されます。たとえば、次のようなものがあります。

1. :open (@ 1!1, 1!2)
2. :open all
3. :scan (@ 1!1:1!10)
4. :trigger:count:auto on
5. :trigger:delay 0.5

1. (@ 1!1, 1!2) — このチャンネルリストパラメータはチャンネルリストを指定するものです。チャンネルのリストは括弧()で囲まれます。リスト中の最初のチャンネルの前には@記号がなければなりません。感嘆符は、スロット番号とカードリレー番号を区切るためのSCPI区切文字です。リスト中の各チャンネルはカンマ(,)によって区切らなければなりません。
2. All — この名前パラメータは、全チャンネルを開きたい場合に、チャンネルリストを指定する代わりに:OPENコマンドのパラメータとして使うことができます。
3. (@ 1!1:1!10) — このチャンネルリストパラメータはスキャンリストを指定するものです。例1と同じタイプですが、この例では、チャンネル1~10が指定されます。レンジリミットのセパレータとしてコロン(:)を使うことに注意してください。
4. On — このブール値パラメータは自動スキャンリストカウントをイネーブルするものです。これの代わりに“1”を使うこともできます。“Off”または“0”では、自動スキャンリストカウントがディスエーブルされます。
5. 0.5 — この実数パラメータは秒単位のデレイ時間を設定するものです。

複数のコマンド

複数のコマンドを送信する場合、それらをセミコロン(:)によって区切れば、同一メッセージで送信することができます。たとえば、

```
:open (@1!1)
:close (@1!2)
```

を送信する代わりに、

```
:open (@1!1);:close (@1!2)
```

を送信することができます。

上記のメッセージが送信されると、最初のコロン(:)によって、それに続くコマンドワードがSCPIルートコマンドであることが指示されます。次のコロンが検出されると、バスポイントは次のコマンドレベルに下がり、コマンドを実行します。バスポイントはセミコロン(;)の後のコロン(:)を検出すると、ルートにリセットされ、最初から同じことを繰り返します。

バスポイントを上手に使うことで、コマンドバス全体を再タイプする必要なく、複数のコマンドを同一コマンドレベル内で扱うことができます。

たとえば、:ARM:LAYer2:SOURce MANualコマンドでは、バスポイントはバス内の最後のコマンドレベルまで下がります。したがって、バス全体を繰り返すことなく、:COUNT、:DELAY、:TIMERコマンドならびにそれぞれ付随の間合せコマンドを同一メッセージに含めることができます。たとえば、

```
:arm:lay2:sour man;:arm:lay2:sour?
```

を送信する代わりに、

```
:arm:lay2:sour man;sour?
```

を送信することができます。

```
:trig:coun 1;:trig:del 1;:trig:tim 1
```

を送信する代わりに、

```
:trig:coun 1;del 1;tim 1
```

を送信することができます。

なお、付加コマンドについてはコロン(:)は含めません。コロンが検出されると、バスポイントは次のコマンドレベルに下がりますから注意してください。たとえば、

```
:trig:del 1;coun:auto on
```

このメッセージが送信されると、バスポイントはDELAYとCOUNTの同一コマンドレベルを下がります。COUNTコマンドの後のコロンの、ポイントは次のコマンドレベルに下がり、AUTOをイネーブルします。

ここで重要なのは次の二点です。

1. バスポインタは下がることしかできない。
2. それぞれの新しいメッセージ(行)はルートコマンドで始まる。

3.7.6 プログラミング例

以下のプログラミング例は、Hewlett-Packard BASIC 4.0プログラミング言語で書かれています。プログラムは、Model 7001が一次アドレス7にセットされていることを前提とします。

プログラミング例 #1 —チャンネルの開閉

以下のプログラムは、スロット1のチャンネル1、3、4、5、6を閉じ、短時間ディレイの後、それらを開くものです。

```
10 OUTPUT 707;:"*RST;:open all"
20 OUTPUT 707;:"clos (@!11, !13-!16)"
30 OUTPUT 707;:"clos:stat?"
40 ENTER 707;AS
50 PRINT AS
60 WAIT 3
70 OUTPUT 707;:"rout:open (@!11, !13-!16)"
80 END
```

行10 閉じているチャンネルを開く。
 行20 スロット1のチャンネル1、3、4、5、6を閉じる。
 行30 閉じたチャンネルを要求する。
 行40 7001をトークにアドレッシングする。
 行50 閉じているチャンネルを表示する。
 行60 3秒のディレイ。
 行70 チャンネルリスト中に指定されているチャンネルを開く。

プログラミング例 #2 — マニュアルスキャン

以下のプログラムは、Model 7001を、スロット1の10チャンネルについて無限回数のマニュアルスキャンを行なうように設定するものです。

```
10 OUTPUT 707;:"syst:pres"
20 OUTPUT 707;:"scan (@!11, !1!0)"
30 OUTPUT 707;:"init"
40 END

行10 7001を:SYSTEM:PRESetデフォルトコンフィギュレーション(スキャンカウント = 無限、チャンネルカウント = 10チャンネル、チャンネルコントロールソース = マニュアル)に戻す。
行20 スキャンリストを定義する。
行30 7001をアイドル状態から抜け出させる。
```

このプログラムが実行されると、スキャンがアームされ(Armインジケータ点灯)、チャンネルスキャンをコントロールするためにフロントパネルのSTEPキーが押されるのを待ちます。Model 7001をリモートコントロールから抜け出させた後(LOCALキーを押した後)、STEPキーを押すたびに、チャンネルリスト中の次のチャンネルがスキャンされます。

プログラミング例 #3 — バストリガコントロールスキャン

以下のプログラムは、例 #2のプログラムを修正し、チャンネルスキャンをバストリガ (*TRGまたはGET) によってコントロールするようにしたものです。

```

10  OUTPUT 707;“:syst:pres”
20  OUTPUT 707;“:scan (@1!1:1!10)”
25  OUTPUT 707;“:trig:sour bus”
30  OUTPUT 707;“:init”
40  END

行10  7001を:SYSTem:PRESetデフォルトコンフィギュレーション (スキャンカウント = 無限、チャンネルカウント = 10チャンネル、チャンネルコントロールソース = マニュアル) に戻す。
行20  スキャンリストを定義する。
行25  チャンネルコントロールソースをバストリガにプログラムする。
行30  7001をアイドル状態から抜け出させる。

```

このプログラムが実行されると、スキャンがアームされ、チャンネルスキャンをコントロールするためのバストリガを待ちます。バストリガが発生させるには、下記の2つのプログラミングステートメントのどちらでも使えます。

```

OUTPUT 707;“*TRG”
TRIGGER 707

```

これらのステートメントの1つが実行されるたびに、バストリガが発生し、チャンネルリスト中の次のチャンネルをスキャンさせます。

プログラミング例 #4 — タイマコントロールスキャン

以下のプログラムは、例 #2のプログラムを修正し、チャンネルスキャンをタイマによってコントロールするようにしたものです。

```

10  OUTPUT 707;“:syst:pres”
20  OUTPUT 707;“:scan (@1!1:1!10)”
25  OUTPUT 707;“:trig:sour tim”
26  OUTPUT 707;“:trig:tim 0.5”
30  OUTPUT 707;“:init”
40  END

```

```

行10  7001を:SYSTem:PRESetデフォルトコンフィギュレーション (スキャンカウント = 無限、チャンネルカウント = 10チャンネル、チャンネルコントロールソース = マニュアル) に戻す。
行20  スキャンリストを定義する。
行25  チャンネルコントロールソースをタイマにプログラムする。
行26  タイマを0.5秒の時間間隔に設定する。
行30  7001をアイドル状態から抜け出させる。

```

このプログラムが実行されると、スキャンがアームされ、チャンネルを0.5秒のレートで連続的にスキャンします。

プログラミング例 #5 — タイマコントロールスキャン

以下のプログラムは、例 #4のプログラムを修正し、2回のスキャンを行なうようにしたものです。最初のスキャンは即時に開始され、2回目のスキャンは最初のスキャンが開始されてから10秒後に開始されます。

```

10  OUTPUT 707;“:syst:pres”
20  OUTPUT 707;“:scan (@1!1:1!10)”
22  OUTPUT 707;“:arm:lay2:coun 2”
23  OUTPUT 707;“:arm:lay2:sour tim”
24  OUTPUT 707;“:arm:lay2:sour tim 10”
25  OUTPUT 707;“:trig:sour tim”
26  OUTPUT 707;“:trig:tim 0.5”
30  OUTPUT 707;“:init”
40  END

行10  7001を:SYSTem:PRESetデフォルトコンフィギュレーション (スキャンカウント = 無限、チャンネルカウント = 10チャンネル、チャンネルコントロールソース = マニュアル) に戻す。
行20  スキャンリストを定義する。
行22  スキャンカウントを2にプログラムする。
行23  スキャンコントロールソースをタイマにプログラムする。
行24  タイマを10秒の時間間隔に設定する。
行25  チャンネルコントロールソースをタイマにプログラムする。
行26  タイマを0.5秒の時間間隔に設定する。
行30  7001をアイドル状態から抜け出させる。

```


第4部

フロントパネル操作

4.1 概要

本節ではフロントパネル操作について詳しく説明します。各項の構成は次の通りです。

4.2 電源投入手順：

Model 7001の電源への接続と電源投入シーケンスについて説明します。

4.3 ディスプレイ：

表示フォーマットについて説明し、Model 7001の使用中表示されることがあるメッセージを示します。

4.4 アナログバックプレーン：

2枚のModel 701Xシリーズスイッチングカードが、Model 7001のアナログバックプレーンを通じて、どのようにして互いに内部接続されるかを説明します。

4.5 メインフレームプログラミング：

チャンネルリストとスキャンリストの作成の仕方について説明します。また、基本的な開閉操作とスキャン操作について説明します。

4.6 MENU：

MAIN MENUから選択される各種の操作とメニュー項目について説明します。

4.7 CARD CONFIG：

CARD CONFIG MENUから行なう、2つのスロット (Card 1およびCard 2) に対する各種の設定操作について説明します。

4.8 SCAN CONFIG：

スキャンオペレーションの各種のレイヤを設定するのに使うCONFIGURE SCANメニューについて説明します。

4.9 デジタル/I/Oポート：

4出力ラインと1入力ラインで構成されている入出力ポートについて説明します。

4.10 外部トリガ：

リヤパネル上のBNCコネクタの使用による外部トリガについて説明します。

4.11 トリガリンク：

トリガリンクの使用による外部トリガについて説明します。

4.2 電源投入手順

Model 7001は、電源電圧90~260VAC、周波数47~440Hzで動作させることができます。

4.2.1 電源接続

用意されている電源コードでModel 7001を適切なAC電源に接続してください。コードのメス端はリヤパネル上のACコンセントに接続し、他端はアースされたACアウトレットに接続します。

警告

感電防止のため、Model 7001はアースされたアウトレットに接続しなければなりません。アースされたアウトレットを使用しないと、感電して傷害や死亡に至る恐れがあります。

4.2.2 電源スイッチ

電源を投入するには、フロントパネル上のPOWERスイッチを押すだけです。スイッチが内側（1）位置にあるとき、電源が入っています。電源を切るには、POWERスイッチをもう一度押します。

4.2.3 電源投入シーケンス

電源投入時、Model 7001は以下のシーケンスを経ます。

1. Model 7001は、そのEPROMおよびRAM記憶素子についてセルフテストを行ないます。障害が検出された場合、Model 7001は下記のメッセージをルックアップし、表示します。

No Comm Link

注：Model 7001の保証期間中に問題が生じた場合は、修理いたしますからModel 7001を東陽テクニカに返送してください。

2. Model 7001がセルフテストに通ると、ファームウェア改訂水準および現在選択されているIEEE-488アドレスが表示されます。

Model 7001
Rev.AXX AY IEEE Addr=07

Rev. AXXはメインCPUのファームウェア水準です。Rev. AYYはディスプレイCPUのファームウェア水準です。7は現在のIEEE-488アドレスです。なお、Model 7001はIEEE-488アドレスを7に設定して出荷されます。

3. 次に、Model 7001はカードの識別（Card ID）を行ないます。装着されたカードに対してメインフレームが正しく動作できる設定になるには、Card IDが必要です。以下に、Card IDについて説明します。

Model 701Xシリーズのカード — Model 701Xシリーズのカード（すなわち、7011、7012または7013）がメインフレーム内に装着されている場合、それは自動的に識別されます。オペレータは何の操作もする必要がありません。たとえば、スロット1にModel 7011マルチプレクサカード、スロット2にModel 7012マトリックスカードがそれぞれ装着されている場合、下記のメッセージが短時間表示されます。

#1=7011 #2=7012

701Xシリーズ以外のカード — Model 7001は、701Xシリーズ以外のカードを自動的に識別することはできません。これらのカードの型番は、ユーザがCard Configurationメニュー（4.7項参照）を使って入力します。電源投入時、メインフレームはメモリをチェックし、ユーザが前回入力した型番を表示します。たとえば、

#1=7052 #2=NONE

“7052”は、前回、スロット1にModel 7052カードが指定されたことを表わします。現在も、スロット1にModel 7052カードが装着されている場合もありますし、あるいは装着されていない場合もあります。“NONE”は、前回、ユーザがスロット2を空に指定したか、あるいは現在は空のスロットに前回装着されたカードがModel 701Xシリーズのカードであることを表わします。スロット2は、空の場合もありますし、あるいは指定されていない701Xシリーズ以外のカードが装着されている場合もあります。

“9990”または“9991”は、前回、スイッチングカードをシミュレートするようにスロット指定が行なわれたことを表わします。“9990”が指定されると、メインフレームは40チャンネル・マルチプレクサカードが装着されているかのように動作します。また、“9991”が指定されると、4×10マトリックスカードが装着されているかのように動作します。このシミュレーションモードでは、カードを装着しないでスイッチングシステムを開発、テストすることができます。

注

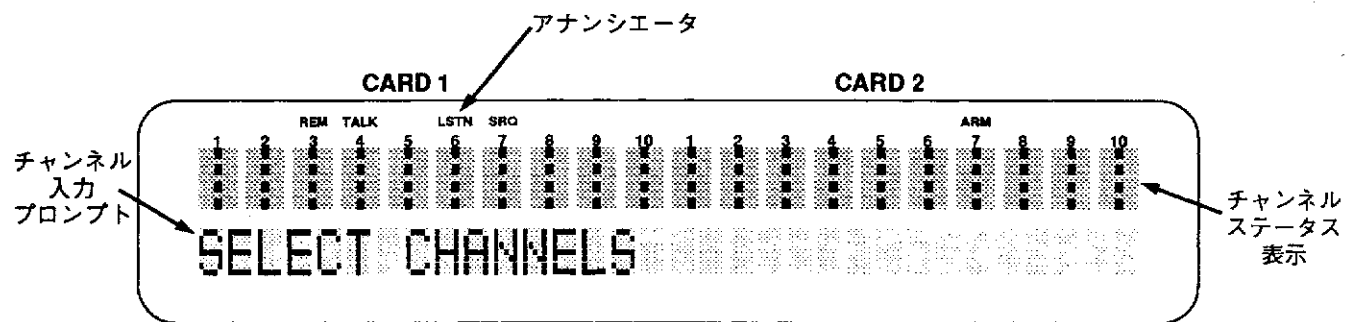
701Xシリーズ以外のカードの場合、電源投入時のCard ID型番メッセージがメインフレーム内に実際に装着されているものと対応していることを確認してください。Card Configurationメニュー（4.7項参照）を使って正しい型番をスロットに指定してください。

4. カード識別後、Model 7001はチャンネルステータス表示状態になります（4.3項「ディスプレイ」参照）。

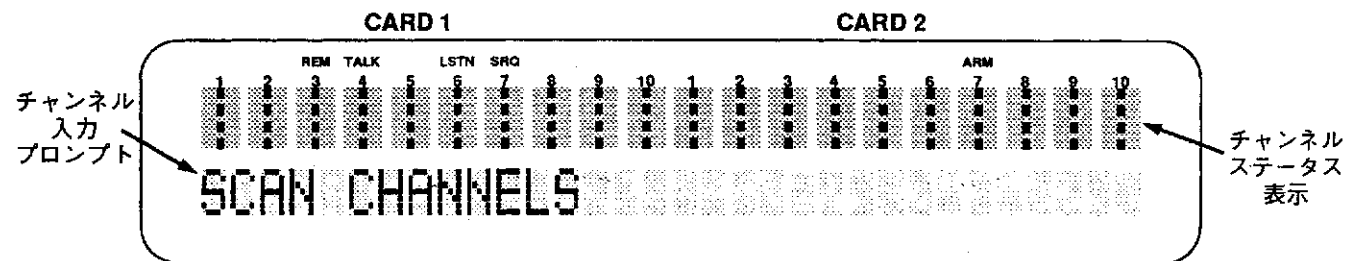
4.3 ディスプレイ

Model 7001のディスプレイ（図4-1）は、主に、チャンネルステータスを入力されたチャンネルリストまたはスキャンリストとともに表示するのに使われます。

チャンネルステータスを表示しないときは、メニューの見出しや選択項目などのインフォメーションメッセージの表示に使われます。ディスプレイの上部には、種々のオペレーションの状態を表示するためのアナンシエータがあります。



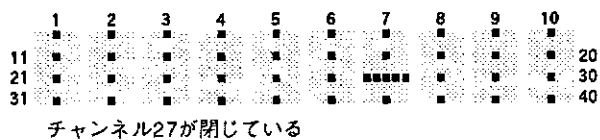
A)チャンネルリスト表示（電源投入時のデフォルト）



B)スキャンリスト表示

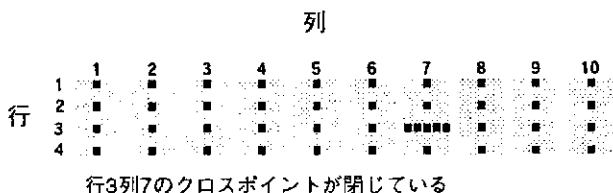
図4-1
表示フォーマット

CARD 1



A) 非マトリックスタイプのカード

CARD 1



B) マトリックスカード

図4-3

チャンネルステータスディスプレイの解釈の仕方

4.3.2 インフォメーションメッセージとエラーメッセージ

フロントパネルディスプレイは、チャンネルステータスを表示しないときは、メニュー項目、ストア情報、リコール情報など他のインフォメーションメッセージ、エラーメッセージおよびステータスメッセージの表示に使われます。Model 7001のステータスメッセージとエラーメッセージを表4-1に示します。

使おうとする機能に関する有用な操作情報を表示するINFOキーもあります。一度押したEXITキーまたはINFOキーをもう一度押すと、メッセージがキャンセルされます。

4.3.3 アナンスエータ

ディスプレイの上部にある5つのアナンスエータは下記の状態を知らせます。

SRQ — Model 7001がIEEE-488バスを通じてサービスを要求すると、それを知らせる。サービス要求イネーブルレジスタをプログラムすることにより、どういう条件でSRQを発生させるかをコントロールすることができます（第5節参照）。

REM — Model 7001がIEEE-488バスを通じて使用されるときにリモート状態にあることを知らせる。Model 7001は、バスRENラインが真のときにリスンにアドレッシングすることによってリモート状態にすることができます。

TALK — Model 7001がIEEE-488バス上のアクティブトーカーであることを知らせる。Model 7001は、一次アドレスを基に導出された正しいバストークコマンドを送信することによってトーカーアクティブ状態にすることができます。

LSTN — Model 7001がアクティブIEEE-488リスナになると、それを知らせる。Model 7001は、リスンにアドレッシングすることによってアクティブリスナ状態にすることができます。

ARM — Model 7001がアイドル状態から抜け出ると、それを知らせる。Model 7001がアイドル状態から抜け出ていると、スキャンは実行できません。

表4-1

エラーメッセージとステータスメッセージ

コード番号	内容
+522	“Slot 2 identification error”(EE)
+521	“Slot 1 identification error”(EE)
+510	“Saved state error”(EE)
+173	“Waiting in arm layer 2”(SE)
+172	“Waiting in arm layer 1”(SE)
+171	“Waiting in trigger Layer”(SE)
+161	“Program running”(SE)
+126	“Device calculating”(SE)
+125	“Device measuring”(SE)
+124	“Device sweeping”(SE)
+123	“Device ranging”(SE)
+122	“Device settling”(SE)
+121	“Device calibrating”(SE)
+101	“Operation Complete”(SE)
0	“No error”(SE)
-100	“Command Error”(EE)
-101	“Invalid Character”(EE)
-102	“Syntax Error”(EE)
-103	“Invarid Separator”(EE)
-104	“Data Type Error”(EE)
-105	“GET not allowed”(EE)
-108	“Parameter not allowed”(EE)
-109	“Missing Parameter”(EE)
-110	“Command Header Error”(EE)
-111	“Command Header Separator Error”(EE)
-112	“Program mnemonic too long”(EE)
-113	“Undefined header”(EE)
-114	“Header suffix out of range”(EE)
-120	“Numeric data error”(EE)
-121	“Invarid character in number”(EE)
-123	“Exponent too large”(EE)
-124	“Too many digits in number”(EE)
-128	“Numeric data not allowed”(EE)
-140	“Character data error”(EE)
-141	“Invalid character data”(EE)
-144	“Character data too long”(EE)
-148	“Character data not allowed”(EE)
-150	“String data error”(EE)

表4-1

エラーメッセージとステータスメッセージ

コード番号	内容
-151	“Invalid string data”(EE)
-154	“String too long”(EE)
-158	“String data not allowed”(EE)
-160	“Block data error”(EE)
-161	“Invalid block data”(EE)
-168	“Block data not allowed”(EE)
-170	“Expression error”(EE)
-171	“Invalid expression”(EE)
-178	“Expression data not allowed”(EE)
-200	“Execution error”(EE)
-201	“Invalid while in local”(EE)
-202	“Settings lost due to rfl”(EE)
-210	“Trigger error”(EE)
-211	“Trigger ignored”(EE)
-212	“Arm ignored”(EE)
-213	“Init ignored”(EE)
-214	“Trigger deadlock”(EE)
-215	“Arm deadlock”(EE)
-220	“Parameter Error”(EE)
-221	“Settings conflict”(EE)
-222	“Parameter data out of range”(EE)
-223	“Too much data”(EE)
-224	“Illegal parameter value”(EE)
-241	“Hardware missing”(EE)
-260	“Expression Error”(EE)
-281	“Cannot create program”(EE)
-282	“Illegal program name”(EE)
-284	“Program currently running”(EE)
-285	“Program syntax error”(EE)
-330	“Self Test failed”(EE)
-350	“Queue overflow”(EE)
-410	“Query interrupted”(EE)
-420	“Query unterminated”(EE)
-430	“Query deadlocked”(EE)
-440	“Query unterminated after indefinite response”(EE)

EE = エラーイベント

SE = ステータスイベント

4.4 アナログバックプレーン

Model 7001は、一方のスロットに装着されているModel 701Xシリーズカードの行またはバンクを他方のスロットに

装着されている別のModel 701Xシリーズカードの行またはバンクに接続させる3ポール・アナログバックプレーンを装備しています。メインフレームの3ポール・アナログバックプレーンを図4-4に示します。

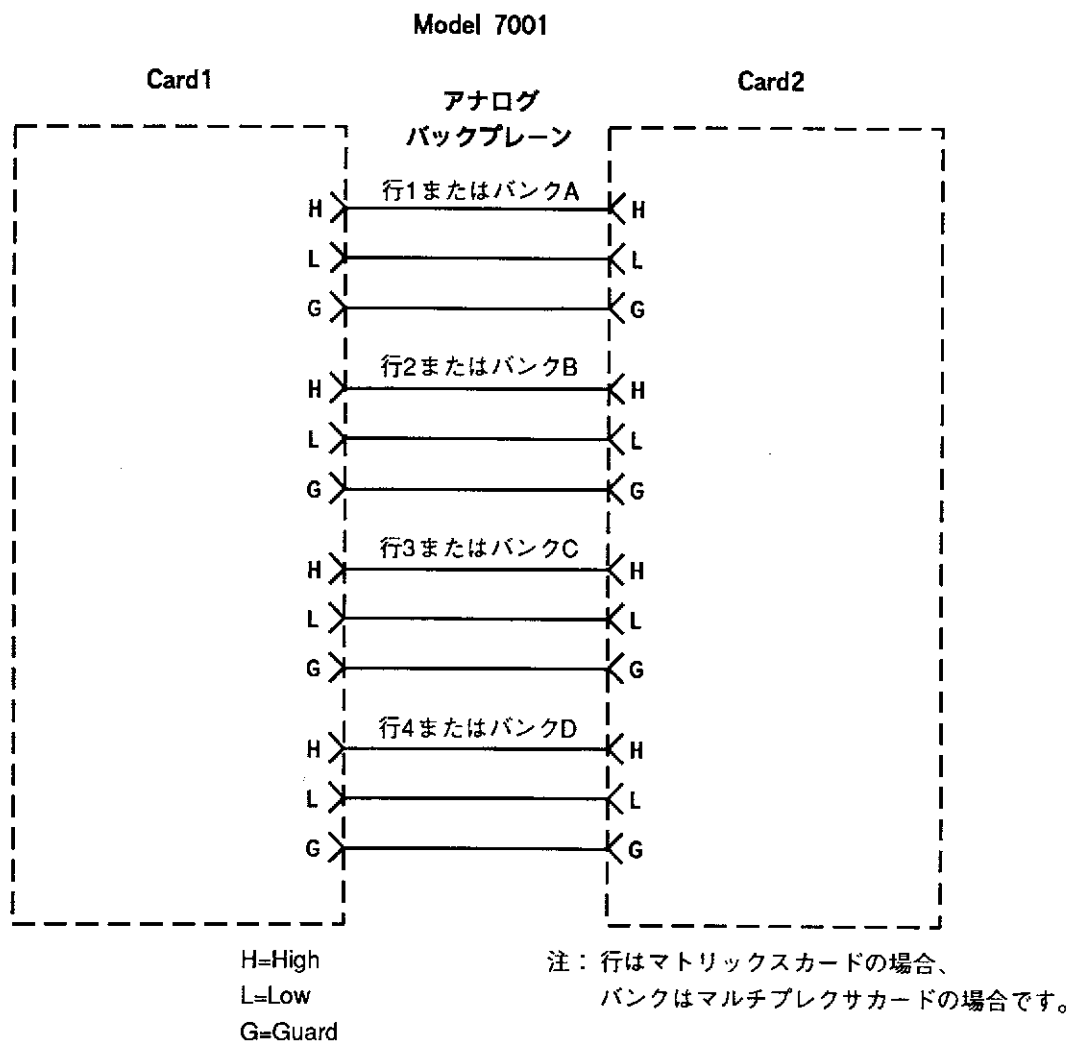


図4-4

Model 7001アナログバックプレーン

たとえば、2枚のModel 7012マトリックスカードのバックプレーンジャンパが取り付けられた場合、これら2枚のカードは、メインフレーム内に装着するだけで互いに接続されます。これで4×10マトリックスが1つ構成されます。一方または両方のマトリックスカード上のバックプレーンジャンパが取り外された場合、2枚のカードは互いに電氣的に隔絶され、4×10マトリックスが4つ構成されることになります。Model 7012マトリックスカードがどのようにしてアナログバックプレーンに接続されるかを図4-5に示します。なお、このマトリックスカードは2ボール (HighとLow) だけしかありません。アナログバックプレーンの第3のボール (Guard) は使用されません。

Model 7001のアナログバックプレーンには必ずしも全部のModel 701Xシリーズカードを接続できるわけではありませんから、注意してください。Model 7013リレースイッチカードは、個別の20のIN/OUTリレーチャンネルで構成されています。各チャンネルは互いに電氣的に隔絶されています。メインフレームの一方のスロットに装着されたModel 7013は、他方のスロットに装着された他のどのカードとも電氣的に隔絶されます。

注

Model 7001では、701Xシリーズ以外のカード (Model 7052など) についてはアナログバックプレーンは使えません。メインフレームの一方のスロットに装着された701Xシリーズ以外のカードは、他方のスロットに装着された他のどのカードとも電氣的に隔絶されます。701Xシリーズ以外のカードを他方のスロットに装着されているカードに接続するには、それらを配線接続するしかありません。

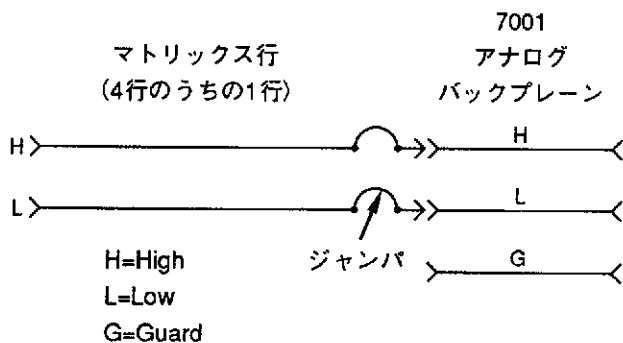


図4-5 バックプレーンへのマトリックス行の接続

4.5 メインフレームプログラミング

以下に、チャンネルを閉じるためとスキャンを実行するためにフロントパネルからModel 7001をプログラムする方法について詳しく説明します。

4.5.1 チャンネルアサインメント

メインフレームをプログラムするのに使われるチャンネルアサインメントは、スイッチングカードのタイプとカードが装着されるスロットに基づいています。

非マトリックスタイプのカード:

Model 7011マルチプレクサカードなどの非マトリックスタイプのカードは、図4-3Aに示すように、単純に40のチャンネル (1~40) で構成されています。Model 7001をプログラムするときは、カードが装着されるスロットを指定する必要があります。そこで、スロット番号とカードチャンネル番号を組み合わせることにより、メインフレームのCHANNELアサインメントが与えられます。スロットとカードチャンネルは感嘆符 (!) によって区切られます。以下に示す例では、“CHANNEL” はメインフレームのプログラミングチャンネルアサインメントを指し、“Channel” はスイッチングカードのチャンネル番号を指します。

CHANNEL !11=Slot 1,Channel 1
 CHANNEL !120=Slot 1,Channel 20
 CHANNEL 2!2=Slot 2,Channel 2
 CHANNEL 2!36=Slot 2,Channel 36

非マトリックスタイプのスイッチングカードの場合の全てのメインフレームチャンネルアサインメントを図4-6に示します。

1	2	3	4	5	6	7	8	9	10
1!1	1!2	1!3	1!4	1!5	1!6	1!7	1!8	1!9	1!10
1!11	1!12	1!13	1!14	1!15	1!16	1!17	1!18	1!19	1!20
1!21	1!22	1!23	1!24	1!25	1!26	1!27	1!28	1!29	1!30
1!31	1!32	1!33	1!34	1!35	1!36	1!37	1!38	1!39	1!40

A. スロット1 (Card 1)

1	2	3	4	5	6	7	8	9	10
2!1	2!2	2!3	2!4	2!5	2!6	2!7	2!8	2!9	2!10
2!11	2!12	2!13	2!14	2!15	2!16	2!17	2!18	2!19	2!20
2!21	2!22	2!23	2!24	2!25	2!26	2!27	2!28	2!29	2!30
2!31	2!32	2!33	2!34	2!35	2!36	2!37	2!38	2!39	2!40

B. スロット2 (Card 2)

例：1!18 = スロット1、チャンネル18
 2!36 = スロット2、チャンネル36

図4-6

チャンネルアサインメント (非マトリックスタイプのカード)

マトリックスカード：

Model 7012などのマトリックスカードのチャンネルは、図4-3Bに示すように、行列のクロスポイントとして構成されています。Model 7001をプログラムするときは、カードが装着されるスロットを指定する必要があります。そこで、スロット番号とクロスポイント座標（行列）を組み合わせることにより、メインフレームのCHANNELアサインメントが与えられます。スロットと行と列は感嘆符（!）によって区切られます。以下に示す例では、“CHANNEL”はメインフレームのプログラミングチャンネルアサインメントを指します。

CHANNEL 1!1!1=Slot 1,Row 1,Column 1
 CHANNEL 2!3!6=Slot 2,Row 3,Column 6

マトリックスカードの場合の全てのメインフレームチャンネルアサインメントを図4-7に示します。

		列									
		1	2	3	4	5	6	7	8	9	10
行	1	1!1!1	1!1!2	1!1!3	1!1!4	1!1!5	1!1!6	1!1!7	1!1!8	1!1!9	1!1!10
	2	1!2!1	1!2!2	1!2!3	1!2!4	1!2!5	1!2!6	1!2!7	1!2!8	1!2!9	1!2!10
	3	1!3!1	1!3!2	1!3!3	1!3!4	1!3!5	1!3!6	1!3!7	1!3!8	1!3!9	1!3!10
	4	1!4!1	1!4!2	1!4!3	1!4!4	1!4!5	1!4!6	1!4!7	1!4!8	1!4!9	1!4!10

A. スロット1 (Card 1)

		列									
		1	2	3	4	5	6	7	8	9	10
行	2	2!1!1	2!1!2	2!1!3	2!1!4	2!1!5	2!1!6	2!1!7	2!1!8	2!1!9	2!1!10
	3	2!2!1	2!2!2	2!2!3	2!2!4	2!2!5	2!2!6	2!2!7	2!2!8	2!2!9	2!2!10
	4	2!3!1	2!3!2	2!3!3	2!3!4	2!3!5	2!3!6	2!3!7	2!3!8	2!3!9	2!3!10
	5	2!4!1	2!4!2	2!4!3	2!4!4	2!4!5	2!4!6	2!4!7	2!4!8	2!4!9	2!4!10

B. スロット2 (Card 2)

例：1!2!4 = スロット1、行2、列4
 2!3!6 = スロット2、行3、列6

図4-7
 チャンネルアサインメント (マトリックスカード)

4.5.2 チャンネルリストとスキャンリスト

Model 7001では2つの基本操作を実行することができます。すなわち、チャンネルのリストを開閉すること、チャンネルのリストをスキャンすることができます。以下に、これらの操作を実行するためのチャンネルの入力方法について説明します。なお、基本操作については、4.5.3項 (チャンネルの開閉) および4.5.4項 (チャンネルのスキャン) で説明します。

各操作ごとにそれぞれ固有のリストがあります。基本的な開閉操作の場合、チャンネルはチャンネルリストに入力されます。スキャン操作の場合は、チャンネルはスキャンリストに入力されます。

原則として、EXITキーを押すことでチャンネルステータスディスプレイを選択し (図4-2のように表示されていない場合)、SCAN LISTキーを押すことでチャンネルリストとスキャンリストをトグル切替えます。

プログラムされたCHANNELはスロット番号とカード番号で構成されます。スロットとカードチャンネルは感嘆符 (!) によって区切られます。マトリックスカードの場合、カードチャンネルは行列で構成され、行と列も感嘆符 (!) によって区切られます。定義されたリスト中の個々のCHANNELはカンマ (,) によって区切られます。CHANNELのレンジはハイフン (-) によって区切られます。メモリにストアされているチャンネルパターンは、文字 "M" とそれに続くメモリロケーション番号によって指定されます。リストはカンマ (,) で終わります。

基本的な開閉操作の場合、チャンネルリスト中のチャンネルの順序は重要ではありません。リスト中の全チャンネルが同時に閉じるか開くかするからです。スキャン操作の場合は、チャンネルの順序は重要な意味をもちます。チャンネルは、スキャンリスト中に示されている順にスキャンされます。

以下のステップを実行することにより、チャンネルリストまたはスキャンリストを作成します。

ステップ1 表示モードを選択する

1. 基本的な開閉操作のためにチャンネルのリストを入力するときは、Model 7001がチャンネルリスト表示状態（図4-1A参照）にあり、下記のメッセージプロンプトが表示されていることを確認してください。

SELECT CHANNELS

Model 7001は電源を投入するとこの状態になります。チャンネルリストとスキャンリストはSCAN LISTキーでトグル切替えます。

2. スキャン操作のためにチャンネルのリストを入力するときは、スキャンリストを表示させます。

SCAN CHANNELS

注：現在、使いたくないリストが表示されている場合は、CLEAR LISTキーを押し、リスト中の全チャンネルの表示をクリアします。

ステップ2 スロット番号とカード番号を入力する

1. スロット番号の入力 — 目的のチャンネルがCard 1（スロット1）にある場合は、キーボードの“1”キーを押します。目的のチャンネルがCard 2（スロット2）にある場合は、キーボードの“2”キーを押します。スロットとチャンネルの間のセパレータは、スロット番号をキーインすると自動的に入力されます。たとえば、スロット1を選択した場合、入力したデータは次のように表示されます。

SELECT CHANNELS 1!
SCAN CHANNELS 1!

2. チャンネルの入力：

非マトリックスタイプのカード — 非マトリックスタイプのカードの場合、キーボードを使ってカードのチャンネル番号を入力します。たとえば、チャンネル20を入力した場合（前回、スロット1を選択した場合）、入力したチャンネルデータは次のように表示されます。

SELECT CHANNELS 1!20
SCAN CHANNELS 1!20

マトリックスカード — マトリックスカードの場合、キーボードを使って、まず行番号（1~4）を入力し、次に列番号を入力します。行と列の間のセパレータは、行番号を入力すると自動的に入力されます。たとえば、行3、列6を入力した場合（前回、スロット2を選択した場合）、入力したチャンネルデータは次のように表示されます。

SELECT CHANNELS 2!3!6
SCAN CHANNELS 2!3!6

注

Model 7001は、最大100チャンネルパターンをM1からM100までのメモリロケーションにストアすることができます。1つのチャンネルパターンにおける閉チャンネルは、キーボードの“M”キーを使ってそのメモリロケーションをキーインするだけで入力することができます。たとえば、特定のスロットに対応する10の閉チャンネルがメモリロケーションM1に収められているものとして、キーボードからM1をキーインするだけでそのチャンネルパターンを入力できます。入力したチャンネルデータは次のように表示されます。

SELECT CHANNELS M
SCAN CHANNELS M1

チャンネルパターンをメモリに入力する方法は4.5.5項で説明します。

この時点では、入力したチャンネルは、左カーソルキー（<）を押すことによってキャンセルできます。

ステップ3 チャンネルセパレータまたはターミネータを入力する

複数のチャンネルの入力にはセパレータが必要です。また、最後に入力したチャンネルの後にはターミネータを使わなければなりません。

カンマ (,) は、別々のチャンネルを区切るのに使われ、リストを終了させる役目もします。ステップ2の説明に従って1つのチャンネルを入力した後、右カーソルキー (▷) またはENTERキーを押し、カンマセパレータを入力します。それから別のチャンネルをキーインすることができます。リスト中の最後のチャンネルをキーインした後、ENTERキーまたは右カーソルキー (▷) を使ってリストを終了させます。カンマターミネータを入力することで、左カーソルキー (◁) を押したときに、最後に入力したチャンネルがキャンセルされるのを防ぐことができます。下例は適当なフォーマットを示すものです。

```
SELECT CHANNELS 1!1,1!2,1!3,1!4,1!5,1!6,1!7,1!8,1!9,1!10,M1,
SCAN CHANNELS 1!1,1!2,1!3,1!4,1!5,1!6,1!7,1!8,1!9,1!10,M1,
```

この表示は、チャンネル1!1~1!10と、メモリロケーション1 (M1) にストアされている閉チャンネルが入力されたことを表わしています。

ハイフン (-) はチャンネルのレンジを指定するのに使われます。ステップ2の説明に従って最初のチャンネルを入力した後、キーボードの“-”キーを押してハイフンを入力し、次に最後のチャンネルをキーインします。下例は適当なフォーマットを示すものです。

```
SELECT CHANNELS 1!1-1!10,M1
SCAN CHANNELS 1!1-1!10,M1
```

この表示は、チャンネル1!1~1!10と、メモリロケーション1 (M1) にストアされている閉チャンネルが入力されたことを表わしています。

ステップ4 ステップ2および3の基本手順を繰り返して必要な全チャンネルを入力する

ステップ5 入力されたリストを必要に応じてエディットする

リストに変更を加えるのにエディットキーを使えます。これらのキーを使うことで、チャンネルを変更、削除し、あるいはリスト中の任意の個所にチャンネルを挿入することができます。エディットを行なうときは、チャンネルレンジ (すなわち、1!1-1!6) およびメモリロケーション (すなわち、M1) はシングルチャンネル入力として扱われます。

1. チャンネルの変更 — 以下のステップを実行することによりチャンネルを変更します。
 - A. カーソルキーを使って、変更すべきチャンネルの始めにカーソルを位置付けます。
 - B. 新しいチャンネルをキーインし、右カーソルキー (▷) またはENTERキーを押してカンマを加えます。

2. チャンネルの削除 — 以下のステップを実行することによりチャンネルを削除します。
 - A. 削除すべきチャンネルの始めにカーソルを位置付けます。
 - B. DELETEキーを押してチャンネルを削除します。
 3. チャンネルの挿入 — 以下のステップを実行することによりリスト中の任意の個所にチャンネルを挿入します。
 - A. リスト中の目的の個所にカーソルを位置付けます。チャンネルは、選択したチャンネルとその前のチャンネルの間に挿入されます。
 - B. INSERTキーを押します。下記のメッセージが表示されます。
- INSERTING ENTRY
- C. 新しいチャンネルをキーインし、右カーソルキー (▷) またはENTERキーを押してカンマを加えます。
 - D. INSERTキーをもう一度押し、挿入ファンクションをキャンセルします。
4. 全チャンネルの削除 — CLEAR LISTキーを押すことによってリスト全体を削除することができます。

チャンネルリストおよびスキャンリスト入力キーの要約

キーボード:

数字キー — スロット番号およびチャンネル番号 (マトリックスカードの場合は行と列) を入力する。

“M” キーと数字キー — ストアされているチャンネルのメモリロケーション (すなわち、M1) を入力する。

“-” — チャンネルのレンジ (すなわち、1!1-1!9) を定義する。

“▷” — チャンネルセパレータ (,) を入力する。

“◁と▷” — リストを終了させた後、カーソルキーを使ってエディットすべきチャンネルを選択する。

ENTER: チャンネルセパレータ (,) およびリストターミネータ (,) (すなわち、1!1,) を入力する。

INSERT: リスト中の選択された個所に新しいチャンネルを入力する。

DELETE: 選択されたチャンネルを削除する。

CLEAR LIST: 表示されているリストをクリアする。

注

1. スキャンリストはModel 7001の電源を切っても失われません。前回定義されたスキャンリストは、次にModel 7001の電源を投入したときも使えます。

2. カードタイプの指定を701Xシリーズ以外のカードに変更するときは、スキャンリストがカードタイプに適合しないことがありますから注意してください。たとえば、スキャンリストがマトリックスチャンネル（すなわち、11111）で構成されている場合、非マトリックスタイプのカードには不適合となります。このような場合には、CLEAR LISTキーを押し、新しいスキャンリストを作成してください。
3. チャンネルリストはModel 7001の電源を切ると失われます。ただし、メモリにストアされているチャンネルパターンはいつでもリコールできます。チャンネルリストは復元されることはありませんが、チャンネルステータスディスプレイはセーブされているパターンに更新されます（4.5.5項参照）。

4.5.3 チャンネルの開閉

Model 7001の基本機能の1つに、ユーザによって指定された1つまたはそれ以上のチャンネルを閉じる（または開く）機能があります。指定された全チャンネルが同時に閉じるか開くかします。ただし、シングルチャンネルモードがイネーブルされている場合は例外です（4.8.4項参照）。シングルチャンネルモードがイネーブルされている場合は、一番若い番号のスロットの一番若い番号のチャンネルが閉じます。

この操作は、Model 7001がチャンネルリスト表示状態のときに実行されます。Model 7001は、電源を投入するとチャンネルリスト表示状態になります。

チャンネルを開閉する手順を実行するには、コントロールされるチャンネルの入力の仕方を知っていなければなりません。原則として、チャンネルリストはキーボードを使って入力します。スロットおよびチャンネル（マトリックスカードの場合は行と列）を入力するには、数字キーを使います。チャンネルセパレータおよびリストターミネータとして使われるカンマ（,）を加えるには、右カーソルキー（▷）（またはENTERキー）を使います。チャンネルのレンジを定義するにはハイフンキー（-）を使います。メモリにストアされているチャンネルパターンの閉チャンネルを入力するには、“M”キーを使います。詳細は4.5.2項を参照してください。

以下のステップを実行することによりチャンネルを開閉します。

注

開閉操作を実行する前に、ご使用のスイッチングカードが正しく設定されていることを確認してください。装着されたカードについて各種設定を行なうには、CARD CONFIG MENUを使います。詳細は4.7項を参照してください。

ステップ1 Model 7001をチャンネルリスト表示状態にする

Model 7001がチャンネルリスト表示状態になっていない場合は、下記のメッセージプロンプトが表示されるまでEXITキーおよび/またはSCAN LISTキーを押します。

```
SELECT CHANNELS
```

ステップ2 チャンネルリストを入力する

4.5.2項の説明に従って開閉すべきチャンネルを入力します。下例は適当なフォーマットを示すものです。

```
SELECT CHANNELS 1!1,1!2,2!1-2!5,M1
```

このチャンネルリストには、スロット1のチャンネル1および2、スロット2のチャンネル1～5、メモリロケーション1（M1）にストアされているチャンネルパターンが含まれます。

ステップ3 リスト中のチャンネルを閉じる（または開く）

以下の手順で開閉操作を実行します。

1. リスト中のチャンネルを閉じるには、CLOSEキーを押します。チャンネルステータスディスプレイ上の“ダッシュ”は、閉じたチャンネルを表わします。
2. 現在閉じているリスト中のチャンネルを開くには、OPENキーを押します。リスト中のチャンネルだけが開きます。閉じてはいてもリスト中に含まれないチャンネルは、OPENキーを押しても開きません。リスト中に含まれないチャンネルを開くには、OPEN ALLを押さなければなりません。

注：OPENキーとCLOSEキーは、通常のチャンネル表示状態のときにしか使えません。OPEN ALLキーは、現在何が表示されているかに関係なく、いつでも使えます。

開閉キーの要約：

CLOSE — チャンネルリスト中に定義されているチャンネルを閉じる。

OPEN — チャンネルリスト中に定義されているチャンネルを開く。

OPEN ALL — 閉じている全チャンネルを開く。

4.5.4 チャンネルのスキャン

Model 7001は、指定されたチャンネルリストをスキャンすることができます。チャンネルがスキャンされる順序は、スキャンリスト中に示されている順序によって決まりません。

スキャンプロセス（トリガモデル）

以下に、スキャンプロセスのフロントパネルコントロールについて説明します。図5-8の流れ図（トリガモデル）は、フロントパネルからのスキャンオペレーションをまとめたものです。

アイドル

Model 7001は、トリガモデルのレイヤのどれか1つの中で動作していない限り、アイドル状態にあるとみなされます。Model 7001がアイドル状態のときは、フロントパネルのARMインジケータは消えています。アイドル状態のときは、Model 7001はスキャンを実行できません。

フロントパネルからSTEPキーを押すことによってModel 7001はアイドル状態から抜け出します。アイドル状態ではないとき（ARMインジケータが点灯しているとき）、スキャンファンクションはイネーブルされているとみなされます。

スキャンは、OPEN ALLキーを押すことによっていつでも打ち切ることができます。スキャンを打ち切ると、Model 7001はアイドル状態に戻ります。

Model 7001がアイドル状態から抜け出ると、オペレーションはトリガモデルのアームレイヤに入ります。

アームレイヤ

注

アームレイヤは、SETUP ARM CONTROLメニュー（4.8.3参照）からプログラムします。

原則として、Model 7001では、オペレーションを次のレイヤ（スキャンレイヤ）に進ませるにはアームイベントが必要です。Immediateアームスペーシングを選択した場合には、Model 7001がアイドル状態から抜け出ると、直ちにオペレーションが次のレイヤに進みます。RESETおよび出荷時デフォルトでは、アームスペーシングはImmediateに設定されます。他のアームスペーシングイベントのどれか1つが選択をした場合には、Model 7001は該当するイベントが発生するまで待ちます。

Manualアームスペーシングを選択した場合には、フロントパネルのSTEPキーが押されるまで待ちます。 GPIBアームスペーシングを選択した場合には、バストリガ（GETまたは*TRG）が受信されるまで待ちます。Externalアームスペーシングを選択した場合には、入力トリガ（リヤパネル上のEXTERNAL TRIGGERコネクタ経由）が受信されるまで待ちます。TrigLink（Trigger Link）アームスペーシングを選択した場合には、入力トリガが（TRIGGER LINKを介して）受信されるまで待ちます。Holdアームスペーシングを選択した場合には、どのアームスペーシングイベントに対しても応答しません。

流れ図から分かるように、プログラムされたスペーシングイベントに対してオペレーションをループアラウンドさせるパスがあります。このパスはソースパイパスと呼ばれます。ソースパイパスがイネーブルされ（Arm Trigger ControlがSourceに設定され）、ExternalまたはTrigLinkが選択された場合、オペレーションは最初にアームレイヤを通るときにスペーシングイベントをループアラウンドします。別のアーム（アームカウント>1）にプログラムされている場合は、パイパスループはイネーブルされていても無効です。オペレーションがIdle（アイドル）に入ると、ソースパイパスループはリセットされます。

ソースパイパスをイネーブルすると、このレイヤの出力トリガもイネーブルされます。オペレーションがスキャンレイヤからアームレイヤに戻ると、出力トリガパルスが発生します。TrigLinkアームスペーシングを選択した場合には、出力トリガパルスは、プログラムされているTRIGGER LINK出力ライン上で使用可能です。Externalアームスペーシングを選択した場合には、出力トリガパルスは、CHANNEL READYコネクタで使用可能です。

このレイヤの出力トリガは、ソースバイパスがディスエーブルされると (Arm Trigger ControlがAcceptorに設定されると)、ディスエーブルされます。RESETおよび出荷時デフォルトでは、Arm Trigger ControlはAcceptorに設定されません。

Model 7001は付加アームのプログラムが可能です。アームカウントは、無限または有限値 (1~9999) に設定することができます。RESETおよび出荷時デフォルトでは、アームカウントは1に設定されます。

オペレーションは、アームレイヤから出た後、スキャンレイヤに入ります。

スキャンレイヤ

注

スキャンレイヤは、SCAN CONTROLメニューからプログラムします (4.8.2項参照)。

スキャンレイヤ内のオペレーションはアームレイヤ内のオペレーションと同様ですが、スペーシングがアームイベントではなくスキャンイベントによってコントロールされます。このレイヤでは、スキャンスペーシングをコントロールするのにタイマを使えます。Timerスキャンスペーシングを選択した場合、オペレーションは、最初にスキャンレイヤを通過すると直ちにチャンネルレイヤに入ります。付加スキャン (1~9999または無限) はそれぞれ、プログラムされたタイマ時間間隔が経過するまで行なわれません。タイマは、1msecから99999.999秒までの時間間隔に設定することができます。

ソースバイパスおよび出力トリガは、アームレイヤでのソースバイパスおよび出力トリガと同様に動作します。ソースバイパス (イネーブルされている場合) は、プログラムされた全スキャンが完了すると、リセットされます (無効になります)。

RESETおよび出荷時デフォルトでは、スキャンスペーシングはImmediate、スキャンカウントは無限、Scan Trigger ControlはAcceptorに設定されます。

オペレーションは、スキャンレイヤから出た後、チャンネルレイヤに入ります。

チャンネルレイヤ

注

チャンネルレイヤは、CHANNEL CONTROLメニューからプログラムします (4.8.1項参照)。

チャンネルレイヤ内のオペレーションはスキャンレイヤ内のオペレーションと同様ですが、スペーシングがスキャンイベントではなくチャンネルイベントによってコントロールされ、出力が常にイネーブルされています。

Timerチャンネルスペーシングを選択した場合には、直ちに最初のチャンネルがスキャンされます。次に、プログラムされたタイマ時間間隔 (1msec~99999.999秒) によって決められたレートで付加チャンネルがそれぞれスキャンされます。Immediateチャンネルスペーシングを選択した場合には、直ちにチャンネルがスキャンされます。スキャンレートは、主に、内部的に設定されたリレーセットリング時間とユーザがプログラムしたディレイ (4.7.4項参照) によって決まります。Model 701Xシリーズのカードの場合、内部セットリング時間は6msecです。すなわち、リレーが閉じてから3msecとリレーが開いてから3msecです。このリレーセットリング時間に、ユーザがプログラムしたディレイが加わります。

各チャンネルごとのデバイスアクション (チャンネルスキャン) は次の通りです。

1. スキャンされる最初のチャンネルではない場合、前のチャンネルを開き、内部セットリングディレイがタイムアウトになるのを待つ。
2. 次のチャンネルを閉じ、内部セットリングディレイがタイムアウトになるのを待つ。
3. ユーザがプログラムしたディレイ (0~99999.999秒) がタイムアウトになるのを待つ。

ソースバイパス (イネーブルされている場合) は、プログラムされた数のチャンネルがスキャンされると、リセットされます (有効になります)。

チャンネルレイヤの出力トリガは常にイネーブルされており、各チャンネルがスキャンされた後 (デバイスアクション後) に発生します。TrigLink以外の、どのチャンネルスペーシングを選択した場合でも、TTL出力トリガパルスはCHANNEL READYコネクタで使用可能です。TrigLinkチャンネルスペーシングの場合は、出力トリガは次のように動作します。

1. 非同期Trigger Linkモードを選択した場合、TTL出力トリガパルスはプログラムされたTrigger Link出力ライン上で使用可能です。
2. 半同期Trigger Linkモードを選択し、ソースバイパスをディスエーブルした場合（Channel Trigger ControlをAcceptorに設定した場合）、Trigger Linkラインはリリースされます（Highになります）。
3. 半同期Trigger Linkモードを選択し、ソースバイパスをイネーブルした場合（Channel Trigger ControlをSourceに設定した場合）、Trigger LinkラインはLowにされ、それからリリースされます。

Trigger Linkの詳細は4.11項を参照してください。

RESETおよび出荷時デフォルトでは、チャンネルスペーシングはManual、チャンネルカウントはスキャンリスト長、Channel Trigger ControlはSourceに設定されます。

スキャンリスト

チャンネルをスキャンするには、スキャンリストへのチャンネルの入力の仕方を知っていなければなりません。原則として、チャンネルリストはキーパッドを使って入力します。スロットおよびチャンネル（マトリックスカードの場合は行と列）を入力するには、数字キーを使います。チャンネルセパレータまたはリストターミネータとして使われるカンマ（,）を加えるには、右カーソルキー（▷）（またはENTERキー）を使います。チャンネルのレンジを定義するにはハイフンキー（-）を使います。メモリにストアされているチャンネルパターンを入力するには、“M”キーを使います。詳細は4.5.2項を参照してください。

チャンネルパターンをスキャンするとき、パターン中の閉スイッチはスキャン中を通じて閉じたままです。テストシステムでバイアス電圧を加えるときはチャンネルパターンを使うと便利です。

スキャン手順

以下のステップを実行することによりチャンネルをスキャンします。

注

スキャン操作を実行する前に、ご使用のスイッチングカードが正しく設定されていることを確認してください。装着されたカードについて各種設定を行なうには、CARD CONFIGメニューを使います。詳細は4.7項を参照してください。

ステップ1 OPEN ALLキーを押す

スキャンコンフィギュレーションの前に、OPEN ALLキーを押すことによってModel 7001をアイドル状態にしなければなりません。これで、現在進行中のスキャンが停止し、全チャンネルが開き、Model 7001がアイドル状態になります。

ステップ2 スキャンリストを作成する

Model 7001がアイドル状態で、4.5.2項の説明に従ってチャンネルを入力します。下例は適当なフォーマットを示すものです。

```
SCAN CHANNELS 1!1,1!2,2!1-2!5,M1,
```

このスキャンリストには、スロット2のチャンネル1～5、およびメモリロケーション1にストアされている閉チャンネルが含まれます。チャンネルは、このリスト中に示されている順にスキャンされます。M1はシングルチャンネルとして扱われます。たとえば、M1が2つの閉チャンネルで構成されている場合、これら2つのチャンネルはM1がスキャンされているとき閉じたままです。

スキャンが終了しても、リスト中の最後のチャンネルは閉じたままです。最後のチャンネルを開きたいときは、閉チャンネルが入っていないメモリロケーションをスキャンリスト中の最後の入力として指定します。たとえば、上記のスキャンリストで、M1に閉チャンネルが入っていなければ、シングルスキャンが終了するとチャンネル2!5が開きます。

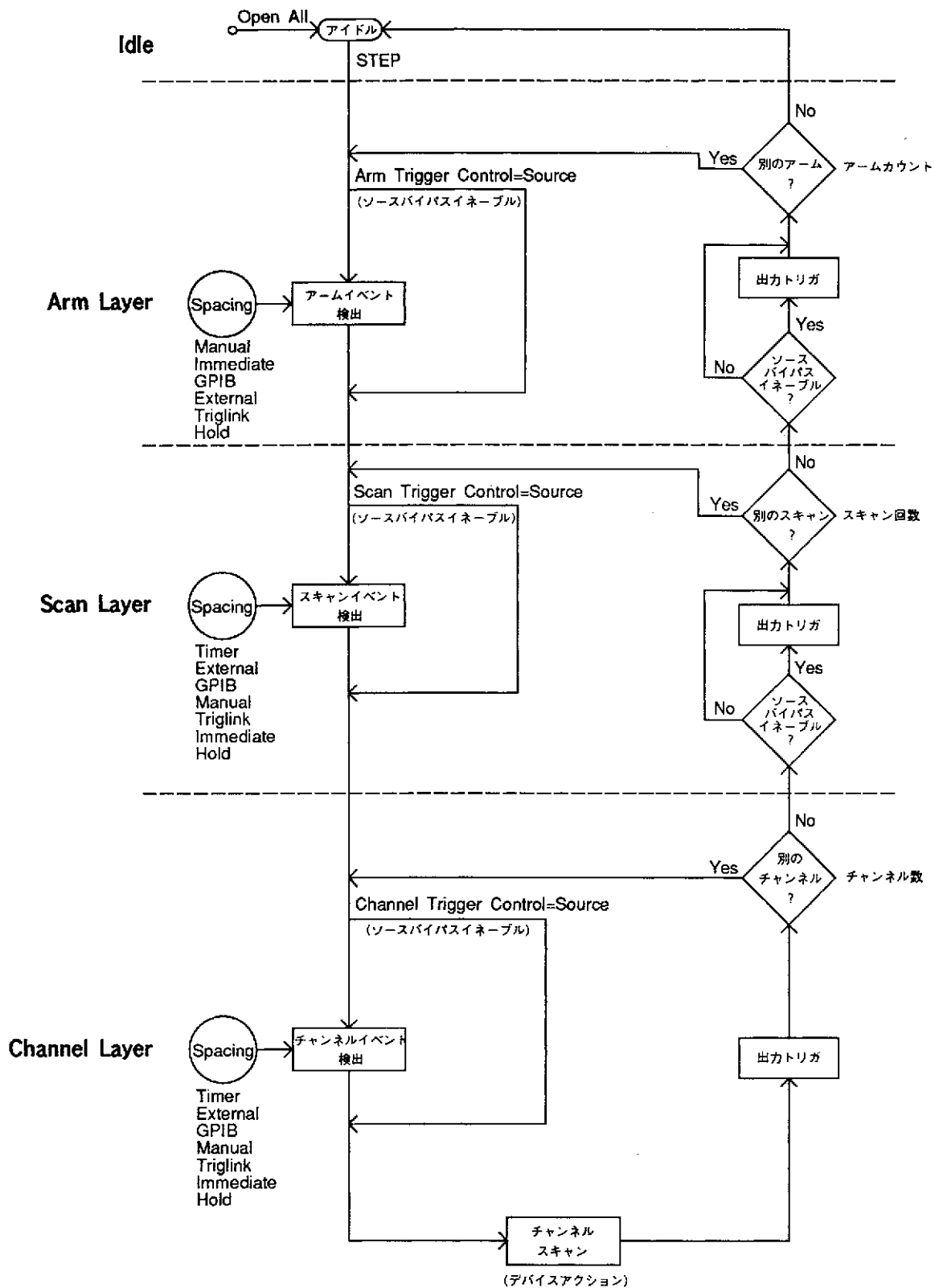


図4-8 フロントパネルスキャンオペレーション

ステップ3 スキャンのチャンネルレイヤを設定する

チャンネルレイヤは、CONFIGURE SCANメニューのCHAN CONTROLメニューからプログラムします。CHANNEL CONTROLメニューの使い方については、4.8.1項および表4-6に詳しく記載してあります。

以下の手順は、チャンネルレイヤを設定するためのステップを示すものです。

1. CONFIGURE SCANメニューからCHAN-CONTROLを選択します。
2. CHANNEL CONTROLメニューからCHANNEL-SPACINGを選択します。
3. SELECT CHAN SPACINGメニューから、チャンネルスキャンプロセスのコントロール用として下記のイベントのどれか1つ（またはIMMEDIATE）を選択します。
 - A. TIMER — これは、チャンネル間のタイマ時間間隔（1msec～99999.999秒）を設定するのに使います。
 - B. EXTERNAL — これを選択した場合、外部トリガによってスキャンスペーシングがコントロールされます。リヤパネル上のEXTERNAL TRIGGERコネクタにトリガが加えられるたびに、次にスキャンされるチャンネルが選択されます。
 - C. GPIB — これを選択した場合、バストリガ（GETまたは*TRG）によってチャンネルスペーシングがコントロールされます。Model 7001がバストリガを受信するたびに、次にスキャンされるチャンネルが選択されます。
 - D. MANUAL — これを選択した場合、フロントパネルのSTEPキーによってチャンネルスペーシングがコントロールされます。STEPキーを押すたびに、次にスキャンされるチャンネルが選択されます。
 - E. TRIGLINK — これを選択した場合、Trigger Linkを通じて受信されたトリガによってチャンネルスペーシングがコントロールされます。トリガが受信されるたびに、次にスキャンされるチャンネルが選択されます。
 - F. IMMEDIATE — これを選択した場合、チャンネルスペーシングをコントロールするのにイベントは使われません。リレーセットリングのデレイとユーザによってプログラムされた任意選択のDELAY（4.7項参照）だけがチャンネルスペーシングに影響します。
4. CHANNEL CONTROLメニューからNUMBER OF CHANSを選択します。
5. NUMBER OF CHANNELSメニューから、下記の項目のどれか1つを選択し、スキャンするチャンネルの数を指定します。

- A. USE SCANLIST LENGTH — これを選択した場合、プログラムされた各スキャン（スキャンカウント）ごとに、スキャンリスト中の全チャンネルのシングルスキャンが行なわれます。
- B. CHAN-COUNT — これを選択した場合、CHANNEL COUNTについて下記の2つのオプションを使えます。
 - a. INFINITE — これを選択した場合、チャンネルスキャンは無限に繰り返されます。
 - b. ENTER-CHAN-COUNT — これは、スキャンするチャンネルの数（1～9999）を定義するのに使います。スキャンリスト長を超えるチャンネルカウントに対しては、チャンネルスキャンはラップアラウンドし、そのカウントに達するまで続きます。

ステップ4 スキャンのスキャンレイヤを設定する

スキャンレイヤは、CONFIGURE SCANメニューのSCAN CONTROLメニューからプログラムします。SCAN CONTROLメニューの使い方については、4.8.2項で詳しく説明します。

以下の手順は、スキャンレイヤを設定するためのステップを示すものです。

1. CONFIGURE SCANメニューからSCAN CONTROLを選択します。
2. SCAN CONTROLメニューからSCAN SPACINGを選択します。
3. SELECT SCAN SPACINGメニューから、スキャンスペーシングのコントロール用として下記のイベントのどれか1つ（またはIMMEDIATE）を選択します。
 - A. TIMER — これは、スキャン間のタイマ時間間隔（1msec～99999.999秒）を設定するのに使います。
 - B. EXTERNAL — これを選択した場合、外部トリガによってスキャンスペーシングがコントロールされます。リヤパネル上のEXTERNAL TRIGGERコネクタにトリガが加えられると、オペレーションがチャンネルレイヤに進みます。
 - C. GPIB — これを選択した場合、バストリガ（GETまたは*TRG）によってスキャンスペーシングがコントロールされます。Model 7001がバストリガを受信すると、オペレーションがチャンネルレイヤに進みます。
 - D. MANUAL — これを選択した場合、フロントパネルのSTEPキーによってスキャンスペーシングがコントロールされます。STEPキーを押すたびに、オペレーションがチャンネルレイヤに進みます。

- E. **TRIGLINK** — これを選択した場合、Trigger Linkを通じて受信されたトリガによってスキャンスペーシングがコントロールされます。トリガが受信されると、オペレーションがチャンネルレイヤに進みます。
 - F. **IMMEDIATE** — これを選択した場合、スキャンスペーシングをコントロールするのにイベントは使われません。オペレーションは即時にチャンネルレイヤに進みます。
4. **SCAN CONTROL**メニューから**NUMBER-OF-SCANS**を選択します。
 5. **NUMBER OF SCANS**メニューから下記の項目のどれか1つを選択し、スキャンの実行回数を指定します。
 - A. **INFINITE** — これを選択した場合、スキャンは無限に繰り返されます。
 - B. **ENTER-SCAN-COUNT** — これは、スキャンの実行回数 (1~9999) を定義するのに使います。

ステップ5 スキャンのアームレイヤを設定する

アームレイヤは、**CONFIGURE SCAN**メニューの**ARM CONTROL**メニューからプログラムします。**ARM CONTROL**メニューの使い方については、4.8.3項で詳しく説明します。

以下の手順は、アームレイヤを設定するためのステップを示すものです。

1. **CONFIGURE SCAN**メニューから**ARM CONTROL**を選択します。
2. **SETUP ARM CONTROL**メニューから**ARM SPACING**を選択します。
3. **SELECT ARM SPACING**メニューから、アーム間の時間間隔のコントロール用として下記のイベントのどれか1つ (または**IMMEDIATE**) を選択します。
 - A. **MANUAL** — これを選択した場合、フロントパネルの**STEP**キーによってアームスペーシングがコントロールされます。**STEP**キーを押すと、オペレーションがスキャンレイヤに進みます。
 - B. **IMMEDIATE** — これを選択した場合、アームスペーシングをコントロールするのにイベントは使われません。オペレーションは即時にスキャンレイヤに進みます。
 - C. **GPIB** — これを選択した場合、バストリガ (**GET**または***TRG**) によってアームスペーシングがコントロールされます。Model 7001がバストリガを受信すると、オペレーションがスキャンレイヤに進みます。

- D. **EXTERNAL** — これを選択した場合、外部トリガによってアームスペーシングがコントロールされます。リヤパネル上の**EXTERNAL TRIGGER**コネクタにトリガが加えられると、オペレーションがスキャンレイヤに進みます。
- E. **TRIGLINK** — これを選択した場合、Trigger Linkを通じて受信されたトリガによってアームスペーシングがコントロールされます。トリガが受信されると、オペレーションがスキャンレイヤに進みます。

4. **SETUP ARM CONTROL**メニューから**ARM COUNT**を選択します。
5. **ARM COUNT**メニューから下記の項目のどれか1つを選択し、アームの実行回数を指定します。
 - A. **INFINITE** — これを選択した場合、アーム回数は無限です。
 - B. **ENTER-ARM-COUNT** — これは、アームの実行回数 (1~9999) を定義するのに使います。

ステップ6 スキャンを開始する

注

スキャンを見るには、Model 7001がチャンネルステータス表示状態に戻るまで**EXIT**キーを押すことによって**CONFIGURE SCAN**メニューから抜け出ます。必要であれば、**SCAN LIST**キーを押し、スキャンされるチャンネルを表示させます。

この手順のステップ1でModel 7001をアイドル状態にしました。このアイドル状態では、スキャンは実行できません。**STEP**キーを押すことによりModel 7001をアイドル状態から抜け出させます。

Model 7001がアイドル状態から抜け出たら、スキャンはインネープルされたとみなされます。スキャンは、3つのレイヤをどのようにプログラムするかそのプログラムの仕方に従って進行します。**Immediate**スペーシングを選択した場合、オペレーションは直ちに次のレイヤに入ります。スペーシングを特定のイベントにプログラムした場合は、そのイベントが発生するまで次のレイヤには入りません。

スキャン例

以下の例は、2チャンネル・スキャンリストを2回スキャンするものとして示します。これらの例では、全てのイベントを**STEP**キーで置き換えることができます。たとえば、Model 7001が**EXTERNAL**トリガイイベントを待つ場合、代わりに**STEP**キーを押してイベントを発生させることができます。

例1:

ARM SPACING = IMMEDIATE
SCAN SPACING = TIMER、EXTERNAL、GPIB、MANUAL
またはTRIGLINK
CHANNEL SPACING = TIMER、EXTERNAL、GPIB、
MANUALまたはTRIGLINK

1. OPEN ALLキーを押して全チャンネルを開き、Model 7001をアイドル状態にします。
2. STEPキーを押し、Model 7001をアイドル状態から抜け出させます (ARMインジケータ点灯)。
3. 最初のスキャンスパーシングイベントによりオペレーションがチャンネルレイヤに入ります。
4. 最初のチャンネルスパーシングイベントによりスキャンリスト中の最初のチャンネルがスキャンされます。
5. 次のチャンネルスパーシングイベントにより2番目のチャンネルがスキャンされます。
6. 次のスキャンスパーシングイベントによりオペレーションがチャンネルレイヤに戻ります。
7. 次のチャンネルスパーシングイベントにより最初のチャンネルがスキャンされます。
8. 次のチャンネルスパーシングイベントにより2番目のチャンネルがスキャンされ、スキャンがディスアーム (アイドル化) されます。

例2:

ARM SPACING = IMMEDIATE
SCAN SPACING = IMMEDIATE
CHANNEL SPACING = TIMER、EXTERNAL、GPIB、
MANUALまたはTRIGLINK

1. OPEN ALLキーを押して全チャンネルを開き、Model 7001をアイドル状態にします。
2. STEPキーを押し、Model 7001をアイドル状態から抜け出させ、オペレーションをチャンネルレイヤに進ませます。
3. 最初のチャンネルスパーシングイベントにより最初のチャンネルがスキャンされます。
4. 次のチャンネルスパーシングイベントにより2番目のチャンネルがスキャンされます。
5. IMMEDIATEスキャンスパーシングが選択されているので、オペレーションが自動的にチャンネルレイヤに戻ります。
6. 次のチャンネルスパーシングイベントにより最初のチャンネルがスキャンされます。
7. 次のチャンネルスパーシングイベントにより2番目のチャンネルがスキャンされ、スキャンがディスアーム (アイドル化) されます。

例3:

ARM SPACING = MANUAL、GPIB、EXTERNALまたは
TRIGLINK
SCAN SPACING = TIMER、EXTERNAL、GPIB、MANUAL
またはTRIGLINK
CHANNEL SPACING = TIMER、EXTERNAL、GPIB、
MANUALまたはTRIGLINK

1. OPEN ALLキーを押して全チャンネルを開き、Model 7001をアイドル状態にします。
2. STEPキーを押し、Model 7001をアイドル状態から抜け出させます。
3. アームスパーシングイベントによりオペレーションがスキャンレイヤに入ります。
4. 最初のスキャンスパーシングイベントによりオペレーションがチャンネルレイヤに入ります。
5. 最初のチャンネルスパーシングイベントによりスキャンリスト中の最初のチャンネルがスキャンされます。
6. 次のチャンネルスパーシングイベントにより2番目のチャンネルがスキャンされます。
7. 次のスキャンスパーシングイベントによりオペレーションがチャンネルレイヤに戻ります。
8. 次のチャンネルスパーシングイベントにより最初のチャンネルがスキャンされます。
9. 次のチャンネルスパーシングイベントにより2番目のチャンネルがスキャンされ、スキャナがディスアーム (アイドル化) されます。

例4:

ARM SPACING = MANUAL、GPIB、EXTERNALまたは
TRIGLINK
SCAN SPACING = IMMEDIATE
CHANNEL SPACING = TIMER、EXTERNAL、GPIB、
MANUALまたはTRIGLINK

1. OPEN ALLキーを押して全チャンネルを開き、Model 7001をアイドル状態にします。
2. STEPキーを押し、Model 7001をアイドル状態から抜け出させます。
3. アームスパーシングイベントによりオペレーションがスキャンレイヤに入ります。
4. IMMEDIATEスキャンスパーシングが選択されているので、直ちにオペレーションがチャンネルレイヤに入ります。
5. 最初のチャンネルスパーシングイベントによりスキャンリスト中の最初のチャンネルがスキャンされます。
6. 次のチャンネルスパーシングイベントにより2番目のチャンネルが選択されます。

7. IMMEDIATEスキャンスペーシングが選択されているので、オペレーションが自動的にチャンネルレイヤに戻ります。
8. 次のチャンネルスペーシングイベントにより最初のチャンネルが選択されます。
9. 次のチャンネルスペーシングイベントにより2番目のチャンネルが選択され、スキャンがディスプレイ（アイドル化）されます。

スキャン注記事項：

1. スキャン中、STEPキーはいつでも使えます。このキーを使ってスペーシングイベントを発生させることができます。たとえば、TIMERスペーシングイベントにプログラムされている場合、STEPキーを使って次のチャンネルを選択することもできます。
2. Forbidden Channelsはチャンネルリスト中に入力できません。
3. Single Channelがイネーブルされている場合は、メモリロケーション（すなわち、M1）にストアされている一番若い番号のチャンネルだけが、選択されると閉じます。
4. スキャン中、Card Pairを使えます。スキャンリスト中のチャンネルとそれらに対応する他のスロットのチャンネルがスキャンされます。

スキャンキーの要約：

OPEN ALL — スキャンを停止し、全チャンネルを開き、Model 7001をアイドル状態にするのに使います。

STEP — Model 7001をアイドル状態から抜け出させるのに使います。また、マニュアルでスキャンをステップスルーするのに使います。

4.5.5 チャンネルパターンのストア (STOREおよびRECALL)

表示されているチャンネルパターンは、メモリロケーションM1～M100にストアすることができます。チャンネルパターンとは、単に、開チャンネルと閉チャンネルのパターンのことです。

チャンネルパターンのストア

以下のステップを実行することにより、チャンネルパターンをメモリロケーションにストアします。

ステップ1 Model 7001をチャンネルリスト表示状態にする

SELECT CHANNELS

ステップ2 不要のチャンネルで現在閉じているものがあれば、OPEN ALLキーを押してそれらを開く

ステップ3 閉じたいチャンネルを入力する

チャンネルリストの入力については4.5.2項に説明してあります。下例は適当なフォーマットを示すものです。

SELECT CHANNELS 1!1,1!2,2!4,

このチャンネルリストには、スロット1のチャンネル1、スロット2のチャンネル4が含まれます。

ステップ4 チャンネルを閉じる

CLOSEを押すことにより、リスト中の指定されたチャンネルを閉じます。

ステップ5 表示されているチャンネルパターンをストアする。

以下のステップを実行することによりチャンネルパターンをストアします。

1. STOREキーを押します。下記の定型メッセージが表示されます。

STORE CHANS AT #001

このメッセージは、チャンネルパターンがメモリロケーションM1にストアされることを表わします。

注

チャンネルパターンをストアしないでStoreモードから抜け出るには、EXITキーを押します。Model 7001がチャンネルリスト表示状態に戻ります。

2. 表示されているメモリロケーションにチャンネルパターンをストアしたい場合は、ステップ3に進んでください。それ以外の場合は、カーソルキー（◀および▶）と数字キーを使って、目的のメモリロケーション（001～100）をキーインします。
3. 表示されているメモリロケーションにチャンネルパターンをストアするには、ENTERキーを押します。Model 7001がチャンネルステータス表示状態に戻ります。

チャンネルパターンのリコール

チャンネルパターンをリコールする方法は2つあります。一方の方法は、RECALLを使ってメモリからチャンネルパターンに直接アクセスするものです。各チャンネルの実際の状態を表わす開/閉チャンネルパターンがディスプレイ上に表示されます。

もう一方の方法は、チャンネルリストおよび/またはスキャンリスト中のチャンネルパターンのメモリロケーション（すなわち、M1）を指定するものです。そのメモリロケーションを閉じるか、またはスキャンすると、チャンネルパターン中の閉チャンネルは閉じます。

RECALLを使ってチャンネルパターンをリコールする

以下のステップを実行することにより、RECALLを使ってチャンネルパターンをストアします。

1. チャンネルリストまたはスキャンリスト表示状態で、RECALLキーを押すと、下記の定型メッセージが表示されます。

RECALL CHANS AT #001

このメッセージは、メモリロケーションM1からチャンネルパターンがリコールされることを表わします。

注

チャンネルパターンをリコールしないでRecallモードから抜け出るには、EXITキーを押します。Model 7001がチャンネルリスト表示状態に戻ります。

2. 表示されているメモリロケーションからチャンネルパターンをリコールしたい場合は、ステップ3に進んでください。それ以外の場合は、カーソルキー（<および>）と数字キーを使って目的のメモリロケーション（001～100）をキーインします。
3. 表示されているメモリロケーションからチャンネルパターンをリコールするには、ENTERキーを押します。Model 7001がチャンネルステータス表示状態に戻り、リコールされたチャンネルパターンを表示します。

注意

選択されたチャンネルパターンに表示が切り替わると、直ちに、対応するカード上のリレーも切り替わります。したがって、チャンネルパターンをリコールする前に、カードリレーを付勢したいかどうか確認してください。

チャンネルリストおよび/またはスキャンリストを介してチャンネルパターンをリコールする

チャンネルパターンは、チャンネルリストおよび/またはスキャンリスト中のメモリロケーションを指定することによってリコールすることができます。たとえば、閉チャンネル2!11および2!36を持っているチャンネルパターンがメモリロケーションM10にストアされているとします。また、M10が次のようなチャンネルリストに含まれているとします。

SELECT CHANNELS 1!1, 1!4, M10,

このチャンネルリストが表示されている状態でCLOSEを押すと、チャンネル1!1、1!4、2!11、2!36が閉じます。

今度は、M10が次のようなスキャンリストに含まれているとします。

SCAN CHANNELS 1!1, 1!4, M10, 1!6,

M10を選択すると、チャンネル2!11と2!36は同時に閉じ、チャンネル1!6をスキャンするときも閉じたままです。

4.6 MENU

Model 7001を設定するための各種の機器操作は、フロントパネルMENUを使って行ないます。MENU構造を表4-2に示します。

MENU使用上の一般規則

1. MAIN MENUを表示させるには、Model 7001を通常のチャンネル表示状態にし、MENUキーを押します。既にメニュー構造に入っている場合は、MAIN MENUの項目が表示されるまでEXITキーを押すことによって表示させることができます（次の規則を参照してください）。

表4-2
MENU構造

メニュー項目	説明
SAVESETUP SAVE RESTORE PWRON FACTORY DEFAULT USER SETUP RESET	セットアップメニュー： セットアップをメモリロケーションにセーブする（10セットアップまで）。 7001をメモリロケーションにストアされているセットアップに戻す。 電源投入メニュー： 電源を投入すると、出荷時デフォルトのセットアップになる。 電源を投入すると、メモリロケーションにストアされているセットアップになる。 7001を元の電源投入時セットアップに戻す。
GPIB ADDRESS STATUS	GPIBセットアップメニュー： IEEE-488アドレスをチェック/変更する。 IEEE-488ステータスバイトを表示する。
DIGITAL-I/O OUTPUT-STATUS OUTPUT-POLARITY INPUT-STATUS	デジタルI/Oメニュー： デジタル出力チャンネルの状態をチェック/変更する。 デジタル出力ポートの極性チェック/変更する。 デジタル入力チャンネルを読み取る。
TEST BUILT-IN TESTS AUTO MANUAL DISPLAY TESTS KEYS PATTERNS	セルフテストメニュー： デジタルボードをテストする： 全テストを自動実行する。 実行するテストを選択する。 表示ボードをテストする： フロントパネルキーの動作確認を行なう。 ディスプレイの動作確認を行なう。
LANGUAGE ENGLISH GERMAN FRENCH	7001言語を選択する。 英語のメッセージを表示する。 独語のメッセージを表示する。 仏語のメッセージを表示する。
GENERAL SERIAL# STATUS-MESSAGES	一般メニュー： シリアルナンバー、SCPIバージョン、ソフトウェア改訂水準を表示する。 ステータスメッセージモードをイネーブル/ディスエーブルする。

MAIN MENUの項目は次のように表示されます。

```

MAIN MENU
SAVESETUP  GPIB      DIGITAL-I/O▶
◀TEST      LANGUAGE GENERAL
    
```

2. EXITキーを押すと、ディスプレイは前のメニューレベルに戻ります。また、EXITが実行された場合、キーインされたパラメータ変更は無視されます。MAIN MENUが表示されているとき、EXITキーまたはMENUキーを押すと、MENUがディスプレイされます。
3. カーソルの位置は、点滅しているメニュー項目またはパラメータによって表わされます。カーソルをある項目から次の項目に移動させるには、カーソルキー (◀および▶) を使います。
4. 表示されている矢印 (◀または▶) は、情報あるいは選択できるメニュー項目がまだほかにあることを表わします。“▶”が表示されているときは、カーソルキーを使って付加メッセージを表示させます。一方、“◀”が表示されているときは、カーソルキー◀を使います。
5. 数値パラメータをキーインするには、変更すべき桁上にカーソルを位置付け、キーボード上の該当するキーを押します。数値をキーインした後、カーソルは右隣の桁に移動します。
6. パラメータの変更を行なうときは、必ず、ENTERキーを押します。
7. 無効のパラメータを入力 (ENTER) すると、エラーとなり、その入力は無視されます。
8. INFOキーはメニュー構造中のどこでも使えます。このキーを使って操作に関する適宜な情報メッセージを表示させます。情報メッセージをキャンセルするには、EXITまたはINFOキーをもう一度押します。

4.6.1 SAVESETUP

SAVESETUPメニューは以下の操作に使います。

- 現在のセットアップを後のリコールのためにメモリにセーブする。最大10セットアップをメモリにセーブすることができます。
- Model 7001をメモリにセーブされているセットアップに戻す。
- 電源投入 (PWRON) 時のコンフィギュレーションをセットアップする。Model 7001は、電源を投入したとき、出荷時デフォルトのセットアップをとるように設定することもできますし、あるいは指定されたメモリロケーションにストアされているセットアップをとるように設定することもできます。

- Model 7001を元の電源投入時セットアップにリセットする (表4-3参照)。

以下のステップを実行することにより、SAVESETUPメニューを表示させます。

1. MAIN MENUを表示させます (4.6項の「一般規則」の#1参照)。
2. カーソルキー (◀および▶) を使ってカーソルをSAVESETUP上に位置付け、ENTERキーを押し、下記のメニューを表示させます。

SETUP MENU

SAVE RESTORE PWRON RESET

SAVE — このメニュー項目は、現在のセットアップをメモリ内の指定されたメモリロケーションにセーブするのに使います。SAVEを選択するには、カーソルをSAVE上に合わせ、ENTERキーを押します。下記の定型メッセージが表示されます。

SAVE SETUP #0(0-9)

1. 現在のセットアップを表示されているメモリロケーションにセーブするには、ENTERキーを押します。Model 7001がSETUP MENUに戻ります。
2. 現在のセットアップを別のメモリロケーションにセーブするには、値 (0~9) をキーインし、ENTERキーを押します。Model 7001がSETUP MENUに戻ります。

RESTORE — このメニュー項目は、Model 7001を、メモリ内の指定されたメモリロケーションストアされているセットアップに戻すのに使います。RESTOREを選択するには、カーソルをRESTORE上に合わせ、ENTERキーを押します。下記の定型メッセージが表示されます。

RESTORE SETUP #0

1. Model 7001を、表示されているメモリロケーションにストアされているセットアップに戻すには、ENTERキーを押します。Model 7001がSETUP MENUに戻ります。
2. Model 7001を、別のメモリロケーションにストアされているセットアップに戻すには、メモリロケーション (0~9) をキーインし、ENTERキーを押します。Model 7001がSETUP MENUに戻ります。

PWRON — このメニュー項目は、Model 7001の電源投入時のセットアップを選択するのに使います。PWRONメニューを選択するには、カーソルをPWRON上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

SET POWER-ON DEFAULT FACTORY-DEFAULT USE-SETUP

FACTORY DEFAULT — これを選択した場合、Model 7001は、次回、電源を投入したときに出荷時デフォルトの状態に戻ります（表4-3参照）。FACTORY-DEFAULTメニュー項目を選択するには、カーソルをFACTORY-DEFAULT上に合わせ、ENTERキーを押します。Model 7001がSETUP MENUに戻ります。

USER SETUP — これを選択した場合、Model 7001は、電源を投入すると、指定されたメモリロケーションにセーブされているセットアップをとります。USER-SETUPを選択するには、カーソルをUSER-SETUP上に合わせ、ENTERキーを押します。下記の定型メッセージが表示されます。

PWRON DEFAULT=#0

1. 電源投入時、表示されているメモリロケーションにストアされているセットアップをとるようするには、ENTERキーを押します。Model 7001がSETUP MENUに戻ります。
2. 電源投入時、別のメモリロケーションにストアされているセットアップをとるようするには、メモリロケーション（0～9）をキーインし、ENTERキーを押します。Model 7001がSETUP MENUに戻ります。

RESET — このメニュー項目は、Model 7001をRESETデフォルトセットアップにリセットするのに使います（表4-3参照）。RESETを選択するには、カーソルをRESET上に合わせ、ENTERキーを押します。下記のメッセージが表示されます。

RESETTING INSTRUMENT

ENTERキーを押すと、下記のメッセージが表示されます。

RESET COMPLETE

ENTERキーを押すと、SETUP MENUに戻ります。

4.6.2 GPIB

GPIBメニューは以下の操作に使います。

- IEEE-488バスのアドレスをチェックおよび/または変更する。
- IEEE-488バスのステータスバイトを表示する。

以下のステップを実行することにより、GPIBメニューを表示させます。

1. MAIN MENUを表示させます（4.6項の「一般規則」の#1参照）。
2. カーソルキー（◀および▶）を使ってカーソルをGPIB上に位置付け、ENTERキーを押し、下記のメニューを表示させます。

GPIB SETUP MENU ADDRESS STATUS

ADDRESS — このメニュー項目は、IEEE-488アドレスをチェックおよび/または変更するのに使います。工場ではアドレスは7に設定されますが、0から30までの任意の値に変更できます。ADDRESSを選択するには、カーソルをADDRESS上に合わせ、ENTERキーを押します。現在、アドレスが7に設定されている場合、下記のメッセージが表示されます。

ADDRESS=07(0-30)

1. 表示されているアドレスのままにしておくのであれば、ENTERまたはEXITキーを押します。Model 7001がGPIB SETUP MENUに戻ります。
2. アドレスを変更するには、有効な値（0～30）をキーインし、ENTERキーを押します。Model 7001がGPIB SETUP MENUに戻ります。

STATUS — このメニュー項目は、IEEE-488ステータスバイトを表示させるのに使います。ステータスバイトについては第5節を参照してください。STATUSを選択するには、カーソルをSTATUS上に位置付け、ENTERキーを押します。たとえば、ステータスビットの全バイトがゼロにリセットされた場合、次のように表示されます。

SHOW STATUS BYTE

EVA=0 QSB=0 MAV=0 ESB=0 ▶

◀ MSS=0 OSB=0

ステータスバイトを見終わったら、ENTERまたはEXITキーを押します。Model 7001がGPIB STATUS MENUに戻ります。

表4-3
デフォルトの状態

項目	出荷時デフォルト	RESET
チャンネルステータス	全チャンネル開	全チャンネル開
チャンネルリスト	クリア	無変更
スキャンリスト	クリア	無変更
GPIBアドレス	7	無変更
デジタルI/O出力レベル	High	High
言語	英語	無変更
ステータスメッセージ	Off	Off
ボール数	2ボール (両スロットとも)	無変更
カードペア	Off	Off
ディレイ	0 sec	0 sec
チャンネルスペーシング	マニュアル (STEPキー)	マニュアル (STEPキー)
チャンネル数	スキャンリスト長	スキャンリスト長
チャンネルトリガコントロール	Source	Source
スキャンスペーシング	Immediate	Immediate
スキャン回数	無限	無限
スキヤントリガコントロール	Acceptor	Acceptor
アームスペーシング	Immediate	Immediate
アームカウント	1	1
アームトリガコントロール	Acceptor	Acceptor
シングルチャンネル	Off	Off
制限チャンネル	なし	無変更

注：“無変更”は、現在設定されている状態に対してRESETが影響を及ぼさないことを表わします。

4.6.3 DIGITAL-I/O

DIGITAL-I/Oメニューは以下の操作に使用します。

- 4デジタル出力チャンネルのOUTPUT-STATUS (オンまたはオフ) をチェックおよび/または変更する。
- デジタル出力ポートのOUTPUT-POLARITYをチェックおよび/または変更する。極性は、アクティブハイ (オン = +5V) にすることもアクティブロー (オン = 0V) にすることもできます。

- シングルデジタル入力チャンネルのINPUT-STATUS (オンまたはオフ) をチェックする。

注

デジタル出力ポートでは0Vおよび+5Vのデフォルトレベルを使用します。ただし、5Vレベルに代えて最大30Vのユーザ供給電圧レベルを使用することができます (4.9.1項参照)。

以下のステップを実行することにより、DIGITAL-I/O MENUを表示させます。

1. MAIN MENUを表示させます (4.6項の「一般規則」の #1参照)。
2. カーソルキー (◀および▶) を使ってカーソルをDIGITAL-I/O上に合わせ、ENTERキーを押して下記のメニューを表示させます。

```
DIGITAL-I/O MENU
OUTPUT-STATUS OUTPUT-POLARITY▶
◀ INPUT-STATUS
```

OUTPUT-STATUS — このメニュー項目は、4デジタル出力チャンネルの状態 (オンまたはオフ) をチェックおよび/または変更するのに使います。“オン”チャンネルの実際のレベルは極性によって決まり、極性は次のメニュー項目 (OUTPUT-POLARITY) によってコントロールされます。OUTPUT-STATUSを選択するには、カーソルをOUTPUT-STATUS上に合わせ、ENTERキーを押します。たとえば、全デジタル出力チャンネルがオフであれば、下記のメッセージが表示されます。

```
DIGITAL OUTPUT LINES
1=OFF 2=OFF 3=OFF 4=OFF
```

1. 表示されている出力チャンネル状態のままにしておくのであれば、ENTERまたはEXITキーを押します。Model 7001がDIGITAL-I/O MENUに戻ります。
2. 1つまたはそれ以上の出力チャンネルの状態を変更するには、以下のステップを実行します。
 - A. 変更したいチャンネルにカーソルを合わせます。
 - B. キーボード上の任意のキーを押します。これらのキーを押すと、“ON”と“OF”が切り替わります。
 - C. 変更したい各チャンネルごとにステップAおよびBを繰り返します。
 - D. 目的の状態が表示されたら、ENTERキーを押します。Model 7001がDIGITAL-I/O MENUに戻ります。

OUTPUT-POLARITY — このメニュー項目は、デジタル出力ポートの極性を変更するのに使います。各チャンネルの極性は、アクティブハイ (ON = +5V) にすることもアクティブロー (ON = 0V) にすることもできます。OUTPUT-POLARITYを選択するには、カーソルをOUTPUT-POLARITY上に合わせ、ENTERキーを押します。下記のメッセージが表示されます。

```
SELECT OUTPUT LINE
TTL1 TTL2 TTL3 TTL4
```

1. カーソルを該当するチャンネル上に合わせ、ENTERキーを押します。選択したチャンネルの極性を表わすメッセージが表示されます。たとえば、

```
TTL1 LOGIC SENSE
ACTIVE-HIGH ACTIVE-LOW
```

カーソルの位置でチャンネルの極性が示されます。

2. 現在選択されている極性のままにしておくのであれば、ENTERまたはEXITキーを押します。ディスプレイがSELECT OUTPUT LINE MENUに戻ります。
3. 極性を変更するには、カーソルキー (◀または▶) を使って、もう一方のメニュー項目上にカーソルを合わせ、ENTERキーを押します。Model 7001がSELECT OUTPUT LINE MENUに戻ります。
4. 終わったら、EXITキーを押してDIGITAL-I/O MENUに戻ります。

INPUT-STATUS — このメニュー項目は、デジタル入力チャンネル (オンまたはオフ) を読み取るのに使います。極性はアクティブハイ (ON = 5V) に固定されています。

INPUT-STATUSを選択するには、カーソルをINPUT-STATUS上に合わせ、ENTERキーを押します。たとえば、デジタル入力チャンネルが“オン”になっていれば、下記のメッセージが表示されます。

```
DIGITAL INPUT LINE
Status=ON
```

ENTERまたはEXITキーを押してDIGITAL-I/O MENUに戻ります。

4.6.4 TEST

TESTメニューは、Model 7001に関する問題を隔離するための診断ツールとして使います。これらのテスト手順については第7節を参照してください。

4.6.5 LANGUAGE

LANGUAGEメニューは、表示メッセージの言語を変更するのに使います。選択できる言語は次の通りです。

- ENGLISHの表示メッセージ
- GERMANの表示メッセージ
- FRENCHの表示メッセージ

以下のステップを実行することにより、メニュー表示メッセージの言語を変更します。

1. MAIN MENUを表示させます (4.6項の「一般規則」の#1参照)。
2. カーソルキー (◀および▶) を使ってカーソルをLANGUAGE上に合わせ、ENTERキーを押します。下記の言語メニュー項目が表示されます。

```
CHOOSE MENU LANGUAGE
ENGLISH GERMAN FRENCH
```

カーソルは現在選択されている言語上に位置します。現在、ENGLISH以外の言語が選択されている場合、メニューはその言語で表示されます。

3. 現在選択されている言語のままにしておくのであれば、EXITまたはENTERキーを押します。ディスプレイがMAIN MENUに戻ります。
4. 他の言語のどれか1つを選択するには、カーソルを目的の言語メニュー項目上に合わせ、ENTERを押します。ディスプレイがMAIN MENUに戻ります。

注：選択した言語は電源投入時のデフォルトになります。

4.6.6 GENERAL

GENERAL MENUは以下の操作に使います。

- Model 7001のシリアルナンバー。
- SCPIバージョンコントロールナンバー。

- Model 7001にインストールされているソフトウェアの改訂水準。
- ステータスメッセージモードの選択。

以下のステップを実行することにより、GENERAL MENUを表示させます。

1. MAIN MENUを表示させます (4.6項の「一般規則」の#1参照)。
2. カーソルキー (◀および▶) を使ってカーソルをGENERAL上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

```
GENERAL MENU
SERIAL# STATUS-MESSAGES
```

SERIAL # — Model 7001のシリアルナンバー、SCPIバージョンおよびソフトウェア改訂水準を表示させるには、カーソルをSERIAL #上に合わせ、ENTERキーを押します。下記の情報が表示されます。

```
SERIAL #XXXXXX
SCPI version #XXX.X software rev AXX/AYY
```

AXXはメインCPUのソフトウェア改訂水準、AYYはディスプレイCPUのソフトウェア改訂水準です。

終わったら、ENTERまたはEXITキーを押してGENERAL MENUに戻ります。

STATUS MESSAGES — このメニュー項目は、ステータスメッセージモードをイネーブルまたはディスエーブルするのに使います。イネーブルされると、実行される操作を確認するためにステータスメッセージが表示されます。カーソルをSTATUS-MESSAGES上に合わせ、ENTERキーを押します。下記のメッセージが表示されます。

```
STATUS MESSAGES
OFF ON
```

カーソルを目的の状態 (オンまたはオフ) 上に合わせ、ENTERキーを押します。ディスプレイがGENERAL MENUに戻ります。

4.7 CARD CONFIG

メインフレームのスロットを設定するための各種の操作は、CARD CONFIG MENUから行ないます。メニュー構造を表4-4に示します。

CARD CONFIG MENUの使用上の一般規則:

1. CARD CONFIG MENUを表示させるには、Model 7001をチャンネルステータス表示状態にし、CARD CONFIGURATIONキーを押します。既にメニュー構造に入っている場合は、CARD CONFIG MENUの項目が表示されるのでEXITキーを押すことによって表示させることができます(次の規則を参照してください)。CARD CONFIG MENUの項目は次のように表示されます。

```

CARD CONFIG MENU
TYPE      #-OF-POLES  CARD-PAIR▶
◀DELAY  READ-I/O-CARD
    
```

2. EXITキーを押すと、ディスプレイが前のメニューレベルに戻ります。また、EXITが実行された場合、キーインされたパラメータ変更は無視されます。

CARD CONFIG MENUが表示されているとき、EXITキーを押すと、カードコンフィギュレーションがディスエーブルされます。

3. カーソルの位置は、点滅しているメニュー項目またはパラメータによって表わされます。カーソルをある項目から次の項目に移動させるには、カーソルキー (◀および▶) を使います。
4. 表示されている矢印 (◀または▶) は、情報あるいは選択できるメニュー項目がまだほかにあることを表わします。“▶”が表示されているときは、カーソルキーを使って付加メッセージを表示させます。一方、“◀”が表示されているときは、カーソルキーを使います。
5. 数値パラメータをキーインするには、変更すべき桁上にカーソルを位置付け、キーボード上の該当するキーを押します。数値をキーインした後、カーソルは右隣の桁に移動します。
6. パラメータの変更を行なうときは、必ず、ENTERキーを押します。
7. 無効のパラメータを入力 (ENTER) すると、エラーとなり、その入力は無視されます。
8. INFOキーはメニュー構造中のどこでも使えます。このキーを使って操作に関する適宜な情報メッセージを表示させます。情報メッセージをキャンセルするには、EXITまたはINFOキーをもう一度押します。

表4-4
CARD CONFIG MENUの構造

メニュー項目	説明
TYPE SLOT-1 SLOT-2	カードタイプを設定する： スロット1のカードの型番を指定する。 スロット2のカードの型番を指定する。
#-OF-POLES SLOT-1 SLOT-2	ポール数を選択する： スロット1のカードのポールモードを選択する。 スロット2のカードのポールモードを選択する。
CARD-PAIR	Card Pairをイネーブルまたはディスエーブルする。
DELAY SLOT-1 SLOT-2	ディレイを設定する： スロット1のカードのディレイを設定する。 スロット2のカードのディレイを設定する。
READ-I/O-CARD	I/Oカードを識別する。

4.7.1 TYPE

TYPE機能は、メインフレーム内に装着されている701Xシリーズ以外の各カードの型番を指定するのに使います。このメニュー項目を使うことで、スロットをシミュレータとして動作するように指定することができます。スロットに値9990を指定すれば、メインフレームは40チャンネルマルチプレクサカードが装着されているかのように動作します。スロットに値9991を指定すれば、メインフレームは4x10マトリックスカードが装着されているかのように動作します。このようなシミュレータにより、メインフレーム内にカードが装着されていない場合でもテストプログラムを開発することができます。

TYPEを選択すると、各スロットごとの指定メッセージが表示されます。これらのメッセージは、電源投入シーケンス中にModel 7001メインフレームによって決められたもので、4.2.3項に説明してあります。原則として、Model 7001

メインフレームは、装着されているModel 701Xシリーズのカードについては（電源投入時に）検出します。他のタイプのカードはいずれも検出されませんから、ユーザが該当する型番をスロットに指定しなければなりません。このユーザ指定型番は、電源を切ってもメインフレームによって“記憶”されます。ただし、メインフレームが701Xシリーズのカードを検出した場合は、ユーザによって指定された型番は無視し、その701Xシリーズのカードの型番をスロットに指定します。TYPEの表示メッセージの解釈の仕方については表4-5を参照してください。

メインフレームが正しく動作するためには、使用するスロットに正しい型番を指定しなければなりません。（701Xシリーズのカードの場合は自動的に指定され、それ以外の場合はユーザがマニュアルで指定する）。型番を指定することで、リレーのセットリング時間が自動的に決定されます。

表4-5
TYPE指定

指定メッセージ	解釈*
SLOT-1 CARD: 701X**	スロット1に701Xシリーズのカードが装着されている。
SLOT-1 CARD: NONE	スロット1にカードが装着されていないか、またはスロット1が指定されていない。
SLOT-1 CARD: 7YYY***	ユーザによってスロット1に701Xシリーズ以外のカードが指定されている。カードがスロットに装着されている場合も装着されていない場合もある。また、別の701Xシリーズ以外のカードが装着されている場合もある。
SLOT-1 CARD: 9990	ユーザによってスロット1が40チャンネルマルチプレクサをシミュレートするように指定されている。
SLOT-1 CARD: 9991	ユーザによってスロット1が4x10マトリックスをシミュレートするように指定されている。701Xシリーズ以外のカードが装着されている場合もある。
SLOT-2 CARD: 701X**	スロット2に701Xシリーズのカードが装着されている。
SLOT-2 CARD: NONE	スロット2にカードが装着されていないか、または指定外のカードが装着されている。
SLOT-2 CARD: 7YYY***	ユーザによってスロット2に701Xシリーズ以外の型番が指定されている。カードがスロットに装着されていない場合もある。また、別の701Xシリーズ以外のカードが装着されている場合もある。
SLOT-2 CARD: 9990	ユーザによってスロット2がカード1x10マルチプレクサをシミュレートするように指定されている。701Xシリーズ以外のカードが装着されている場合もある。
SLOT-2 CARD: 9991	ユーザによってスロット2が4x10マトリックスをシミュレートするように指定されている。701Xシリーズ以外のカードが装着されている場合もある。

* メインフレーム内にカードを装着するか、またはそこからカードを取り出すと、必ず、パワーサイクリングが行なわれるものとする。

** Model 7011、7012または7013などのModel 701Xシリーズカード。

*** Model 7052~7402。

以下のステップを実行することにより、カードのTYPEを指定します。

1. CARD CONFIG MENUを表示させます (4.7項の「一般規則」の#1参照)。
2. カーソルをTYPE上に合わせ、ENTERキーを押します。下記のメッセージが表示されます。

SET CARD TYPE
SLOT-1 SLOT-2

SLOT-1 — スロット1を指定するには、カーソルをSLOT-1上に合わせ、ENTERキーを押します。現在のスロット指定を示すメッセージが表示されます。たとえば、前回、スロット1にModel 7052を指定した場合、下記のメッセージが表示されます。

SLOT-1 CARD:7052

注：スロット1にModel 701Xシリーズのカードが指定されている場合、そのカードがスロットに実際に装着されているということですから、指定を変更することはできません。SET CARD TYPEメニューに戻るには、EXITキーを押します。

1. 下記のガイドラインに従ってスロット1の正しい指定を決めます。
 - A. スロット1が空で、使わないのであれば、“NONE”と指定します。そうすると、スロット1は使われませんから、スロット1のフロントパネルチャンネルステータスディスプレイはオフになります。
 - B. スロット1に装着されている701Xシリーズ以外のカード (Model 7052~7402) を使用する場合は、スロット1にその型番を指定します。たとえば、スロット1にModel 7062が装着されている場合、スロット1を“7062”と指定します。
 - C. スロット1を40チャンネルマルチプレクサのシミュレータに使うのであれば、スロット1を“9990”と指定します。4x10マトリックスをシミュレートする場合は、スロット1を“9991”と指定します。なお、スロット1をシミュレータとして使っている間、装着されたままの701Xシリーズ以外のカードはコントロールされます。
2. カーソルキー (◀および▶) を使って正しいスロット指定を表示させます。原則として、◀カーソルキーを押すと、表示されている型番が+1され、▶カーソルキーを押すと、型番が-1されます。
3. 正しいスロット指定が表示されたら、ENTERキーを押します。ディスプレイがSET CARD TYPEメニューに戻ります。

SLOT-2 — スロット2を指定するには、カーソルをSLOT-2上に合わせ、ENTERキーを押します。現在のスロット指定を示すメッセージが表示されます。たとえば、前回、スロット2にModel 7052を指定した場合、下記のメッセージが表示されます。

SLOT-2 CARD:7052

注：スロット2にModel 701Xシリーズのカードが指定されている場合、そのカードがスロットに実際に装着されているということですから、指定を変更することはできません。SET CARD TYPEメニューに戻るには、EXITキーを押します。

1. 下記のガイドラインに従ってスロット2の正しい指定を決めます。
 - A. スロット2が空で、使わないのであれば、“NONE”と指定します。そうすると、スロット2は使われませんから、スロット2のフロントパネルチャンネルステータスディスプレイはオフになります。
 - B. スロット2に装着されている701Xシリーズ以外のカード (Model 7052~7402) を使用する場合は、スロット2にその型番を指定します。たとえば、スロット2にModel 7062が装着されている場合、スロット2を“7062”と指定します。
 - C. スロット2を40チャンネルマルチプレクサのシミュレータに使うのであれば、スロット2を“9990”と指定します。4x10マトリックスをシミュレートする場合は、スロット2を“9991”と指定します。なお、スロット2をシミュレータとして使っている間、装着されたままの701Xシリーズ以外のカードはコントロールされます。
2. カーソルキー (◀および▶) を使って正しいスロット指定を表示させます。原則として、◀カーソルキーを押すと、表示されている型番が+1され、▶カーソルキーを押すと、型番が-1されます。
3. 正しいスロット指定が表示されたら、ENTERキーを押します。ディスプレイがSET CARD TYPEメニューに戻ります。

4.7.2 #-OF-POLES

スイッチングカードの中には、ポールコンフィギュレーションを変更できるものがあります。たとえば、Model 7011は、通常は、2ポール、40チャンネルのマルチプレクサカードとして使われますが、バンクAおよびBをバンクCおよびDからアイソレートできますから、4ポールオペレーションに設定することができます。4ポールモードでは、チャンネルは2つずつペアに生まれ、20の4ポールチャンネルを構成します。詳細はModel 7011取扱説明書を参照してください。

付加ポールモード機能をもつカードを示すと、次の通りです。

カード	ポールモード		
	1ポール	2ポール	4ポール
7011	使用不可	40チャンネル	20チャンネル
7056	20チャンネル	10チャンネル	使用不可
7156	20チャンネル	10チャンネル	使用不可

無効のポールモードを選択しようとする、設定矛盾エラーとなり、その入力は無視されます。

以下のステップを実行することにより、#-OF-POLESメニューを表示させます。

1. CARD CONFIG MENUを表示させます (4.7項の「一般規則」の#1参照)。
2. カーソルを#-OF-POLES上に合わせ、ENTERキーを押して下記のメニューを表示させます。

SELECT # OF POLES
SLOT-1 SLOT-2

SLOT-1 — このメニュー項目は、スロット1のカードのポールモードを設定するのに使います。カーソルをSLOT-1上に合わせ、ENTERキーを押します。ポールモードの選択肢が表示されます。現在選択されているポールモードは、カーソル (点滅しているメニュー項目) によって表わされます。

OF POLES SLOT-1
1-POLE 2-POLE 4-POLE

1. 表示されているポールモードのままにしておくのであれば、ENTERまたはEXITキーを押します。ディスプレイがSELECT # OF POLESメニューに戻ります。
2. 表示されているポールモードを変更するには、カーソルを目的のメニュー項目上に合わせ、ENTERキーを押します。ディスプレイがSELECT # OF POLESメニューに戻ります。

SLOT-2 — このメニュー項目は、スロット2のカードのポールモードを設定するのに使います。カーソルをSLOT-2上に合わせ、ENTERキーを押します。ポールモードの選択肢が表示されます。現在選択されているポールモードは、カーソル (点滅しているメニュー項目) によって表わされます。

OF POLES SLOT-2
1-POLE 2-POLE 4-POLE

1. 表示されているポールモードのままにしておくのであれば、ENTERまたはEXITキーを押します。ディスプレイがSELECT # OF POLESメニューに戻ります。
2. 表示されているポールモードを変更するには、カーソルを目的のメニュー項目上に合わせ、ENTERキーを押します。ディスプレイがSELECT # OF POLESメニューに戻ります。

4.7.3 CARD-PAIR

CARD-PAIRは、Model 7001のCARD-PAIR機能をネーブル (オン) またはディスエーブル (オフ) するのに使います。CARD-PAIRをイネーブルすることで、同数のチャンネルを持っている2枚の同タイプのカードをペアにすることができます。イネーブルされると、両方のカードの対応するチャンネルがペアとして動作します。たとえば、スロット1のチャンネル4を閉じると、スロット2のチャンネル4も閉じます。逆に、スロット2のチャンネル5を閉じると、スロット1のチャンネル5も閉じます。この機能は、4線スキャンを行なうときに特に便利です。

以下のステップを実行することにより、CARD-PAIR機能の状態をチェックまたは変更します。

1. CARD CONFIG MENUを表示させます (4.7項の「一般規則」の#1参照)。
2. カーソルをCARD-PAIR上に合わせ、ENTERキーを押します。CARD-PAIR機能の現在の状態は、点滅しているカーソルによって示されます。

SET CARD PAIR
OF ON

3. 表示されているCARD-PAIR状態のままにしておくのであれば、ENTERまたはEXITキーを押します。ディスプレイがCARD CONFIG MENUに戻ります。
4. もう一方の状態を選択するには、カーソルをそのメニュー項目上に合わせ、ENTERキーを押します。ディスプレイがCARD CONFIG MENUに戻ります。

4.7.4 DELAY

DELAY機能は、メインフレーム内に装着されている2枚のカードそれぞれのチャンネルダイレイを設定するのに使います。ダイレイは、00000.001秒 (1msec) から99999.999秒まで設定できます。このダイレイは、内部的に設定されたリレーセットリングダイレイに加わるものです。

このユーザプログラムディレイは、スキャン中におけるチャンネル間のディレイです。最初のディレイは、最初のチャンネル（1つまたは複数）が閉じた後に生じます。言い換えれば、プログラムされたディレイはスキャンの開始時には生じないということです。

以下のステップを実行することにより、ディレイを設定します。

1. CARD CONFIG MENUを表示させます（4.7項の「一般規則」の#1参照）。
2. カーソルをDELAY上に合わせ、ENTERキーを押します。下記のメッセージが表示されます。

SET DELAY FOR:
SLOT-1 SLOT-2

SLOT-1 — スロット1のカードのディレイを設定するには、カーソルをSLOT-1上に合わせ、ENTERキーを押します。現在設定されているディレイが表示されます。たとえば、スロット1のカードを1秒のディレイに設定した場合、下記のメッセージが表示されます。

(1)DELAY=00001.000

1. ディレイを変更するには、カーソルキーとキーボードを使って目的の値をキーインします。
2. 目的のディレイが表示されたら、ENTERキーを押します。ディスプレイがSET DELAY FOR:メニューに戻ります。

SLOT-2 — スロット2のカードのディレイを設定するには、カーソルをSLOT-2上に合わせ、ENTERキーを押します。現在設定されているディレイが表示されます。たとえば、スロット2のカードを10秒のディレイに設定した場合、下記のメッセージが表示されます。

(2)DELAY=00010.000

1. ディレイを変更するには、カーソルキーとキーボードを使って目的の値をキーインします。
2. 目的のディレイが表示されたら、ENTERキーを押します。ディスプレイがSET DELAY FOR:メニューに戻ります。

4.7.5 READ-I/O-CARD

このメニュー項目は、Model 7001にI/Oカードが装着されているかどうかを確認するのに使います。

1. CARD CONFIG MENUを表示させます（4.7項の「一般規則」の#1参照）。
2. カーソルをREAD-I/O-CARD上に合わせ、ENTERキーを押し、I/Oカードの型番を表示させます。I/Oカードが装着されていない場合は、下記のメッセージが表示されます。

No I/O Cards

3. EXITを押し、CARD CONFIG MENUに戻ります。

4.8 SCAN CONFIG

スキャンを設定するための各種の操作は、CONFIGURE SCANメニューから行ないます。メニュー構造を表4-6に示します。

表4-6
CONFIGURE SCANメニュー構造

メニュー項目	説明
CHAN-CONTROL	チャンネルレイヤを設定する：
CHANNEL-SPACING	チャンネルスペーシングを選択する：
TIMER	タイマにより各チャンネルを選択する。
EXTERNAL	外部トリガにより各チャンネルを選択する。
GPIB	バストリガにより各チャンネルを選択する。
MANUAL	STEPキーにより各チャンネルを選択する。
TRIGLINK	Trigger Linkトリガにより各チャンネルを選択する。
ASYNCHRONOUS	入力/出力トリガに別々のラインを使用する。
SEMI-SYNCHRONOUS	入力/出力トリガに同一のラインを使用する。
IMMEDIATE	チャンネルを無限にスキャンする。
HOLD	スキャンをチャンネルレイヤでホールドする。
NUMBER-OF-CHANS	スキャンするチャンネルの数を定義する：
USE-SCANLIST-LENGTH	カウント = スキャンリスト中のチャンネル入力数。
CHAN-COUNT	ユーザ定義カウント：
INFINITE	チャンネルスキャンを無限に繰り返す。
ENTER-CHAN-COUNT	カウント = ユーザ定義値 (1~9999)。
CONTROL	トリガモードを選択する：
SOURCE	ソースバイパスをイネーブルする。
ACCEPTOR	ソースバイパスをディスエーブルする。

表4-6
CONFIGURE SCANメニュー構造

メニュー項目	説明
SCAN-CONTROL SCAN-SPACING TIMER EXTERNAL GPIB MANUAL TRIGLINK IMMEDIATE HOLD NUMBER-OF-SCANS INFINITE ENTER-SCAN-COUNT CONTROL SOURCE ACCEPTOR	スキャンレイヤを設定する： スキャンスペーシングを選択する： タイマによりスキャンスペーシングをコントロールする。 外部トリガによりスキャンスペーシングをコントロールする。 バストリガによりスキャンスペーシングをコントロールする。 STEPキーによりスキャンスペーシングをコントロールする。 Trigger Linkトリガによりスキャンスペーシングをコントロールする。 オペレーションを即時にチャンネルレイヤに進ませる。 スキャンをスキャンレイヤでホールドする。 スキャンの実行回数を定義する： スキャンを無限に繰り返す。 カウント = ユーザ定義値 (1~9999)。 トリガコントロールモードを選択する： ソースバイパスをイネーブルする。 ソースバイパスをディスエーブルする。
ARM-CONTROL ARM SPACING MANUAL IMMEDIATE GPIB EXTERNAL TRIGLINK HOLD ARM-COUNT INFINITE ENTER-ARM-COUNT TRIGGER-CONTROL SOURCE ACCEPTOR	アームレイヤを設定する。 アームスペーシングコントロールを選択する： STEPキーによりスキャナをアームする。 スキャナを即時にアームする。 バストリガによりスキャナをアームする。 外部トリガによりスキャナをアームする。 Trigger Linkトリガによりスキャナをアームする。 スキャンをアームレイヤでホールドする。 スキャナのアーム回数を定義する： スキャナを連続的に再アームする。 ユーザ定義カウント値 (1~9999)。 トリガコントロールモードを選択する： ソースバイパスをイネーブルする。 ソースバイパスをディスエーブルする。
CHAN-RESTRICTIONS SINGLE-CHAN RESTRICTED-CHANNELS	チャンネル制限を定義する： シングルチャンネルモードをイネーブル/ディスエーブルする。 閉じることができないチャンネルを指定する。

CONFIGURE SCANの使用上の一般規則：

1. CONFIGURE SCANメニューを表示させるには、Model 7001をチャンネルステータス表示状態にし、SCAN CONFIGURATIONキーを押します。既にメニュー構造に入っている場合は、CONFIGURE SCANメニュー項目が表示されるまでEXITキーを押すことによって表示させることができます。CONFIGURE SCANのメニュー項目は次のように表示されます。

```

CONFIGURE SCAN
CHAN-CONTROL   SCAN-CONTROL▶
◀ARM-CONTROL  CHAN-RESTRICTIONS
    
```

2. EXITキーを押すと、ディスプレイは前のメニューレベルに戻ります。また、EXITが実行された場合、キーインされたパラメータ変更は無視されます。CONFIGURE SCANメニューが表示されているとき、EXITキーを押すと、SCAN CONFIGURATIONがディスエーブルされます。
3. カーソルの位置は、点滅しているメニュー項目またはパラメータによって表わされます。カーソルをある項目から次の項目に移動させるには、カーソルキー（◀および▶）を使います。
4. 表示されている矢印（◀または▶）は、情報あるいは選択できるメニュー項目がまだほかにあることを表わします。“▶”が表示されているときは、▶カーソルキーを使って付加メッセージを表示させます。一方、“◀”が表示されているときは、◀カーソルキーを使を使います。
5. 数値パラメータをキーインするには、変更すべき桁上にカーソルを位置付け、キーボード上の該当するキーを押します。数値をキーインした後、カーソルは右隣の桁に移動します。
6. パラメータの変更を行なうときは、必ず、ENTERキーを押します。
7. 無効のパラメータを入力（ENTER）すると、エラーとなり、その入力は無視されます。
8. INFOキーはメニュー構造中のどこでも使えます。このキーを使って操作に関する適宜な情報メッセージを表示させます。情報メッセージをキャンセルするには、EXITまたはINFOキーをもう一度押します。

4.8.1 CHAN-CONTROL

チャンネルコントロールは以下の操作に使います。

- スキャンにおけるチャンネル閉鎖の時間間隔（CHANNEL SPACING）をコントロールするイベントを選択する。
- スキャンするNUMBER OF CHANS（チャンネル数）を指定する。
- ソースバイパスをイネーブルする。

以下のステップを実行することにより、CHANNEL CONTROLメニューを表示させます。

1. CONFIGURE SCANメニューを表示させます（4.8項の「一般規則」の#1参照）。
2. カーソルキー（◀および▶）を使ってカーソルをCHAN-CONTROL上に合わせ、ENTERキーを押して下記のメニューを表示させます。

```

CHANNEL CONTROL
CHANNEL-SPACING  NUMBER-OF CHANS▶
◀CONTROL
    
```

CHANNEL SPACING — このメニュー項目は、スキャンにおけるチャンネル閉鎖の時間間隔をコントロールするイベントを選択するのに使います。このメニュー項目を選択するには、CHANNEL CONTROLメニューが表示されている状態で、カーソルをCHANNEL-SPACING上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

```

SELECT CHAN SPACING
TIMER           EXTERNAL GPIB MANUAL▶
◀TRIGLINK IMMEDIATE HOLD
    
```

TIMER — チャンネルスペーシングをコントロールするのにタイマを使います。タイマを使って、スキャンにおけるチャンネル閉鎖の時間間隔を設定します。タイマは、0から99999.999秒まで0.001秒（1msec）単位で設定できます。最初の時間間隔は、スキャンされる最初のチャンネルが閉じた後に始まります。タイマを使うには以下のステップを実行します。

1. SELECT CHAN SPACINGメニューが表示されている状態で、カーソルをTIMER上に合わせ、ENTERキーを押します。現在設定されている時間間隔（秒単位）を示すメッセージが表示されます。たとえば、タイマが1msecに設定されている場合、下記のメッセージが表示されます。

```
INTERVAL=00000.001
```

2. 表示されている時間間隔のままにしておくのであれば、EXITまたはENTERキーを押します。ディスプレイがCHANNEL CONTROLメニューに戻ります。

3. 別の時間間隔を設定するには、キーパッドを使って値（秒単位）をキーインし、ENTERキーを押します。ディスプレイがCHANNEL CONTROLメニューに戻ります。

注：タイマが選択されると、フロントパネルのSTEPキー（MANUAL参照）がアクティブになります。STEPキーを押すたびに、次にスキャンされるチャンネルが選択されます。これを使うことで、時間間隔が比較的長いスキャンを迅速にステップスルーすることができます。

EXTERNAL — これを選択した場合、外部トリガによってチャンネルスペーシングがコントロールされます。Model 7001にトリガが加えられるたびに、現在のチャンネルが開き、次のチャンネルが閉じます。外部トリガは、リヤパネル上の“EXTERNAL TRIGGER” BNCコネクタに加えられます。外部トリガモードでは、Model 7001を他の機器へのトリガソースとして使えます。チャンネルが閉じ、安定した後、リヤパネル上の“CHANNEL READY” BNCコネクタにはトリガパルスが加えられます。このトリガパルスは、たとえば、読取りのためにDMMをトリガするのに使えます。外部トリガについては、4.10項で詳しく説明します。

SELECT CHAN SPACINGメニューから外部トリガを選択するには、カーソルをEXTERNAL上に合わせ、ENTERキーを押します。ディスプレイがCHANNEL CONTROLメニューに戻ります。

注：外部トリガを選択すると、フロントパネルのSTEPキー（MANUAL参照）がアクティブになります。STEPキーを押すたびに、次にスキャンされるチャンネルが選択されます。

GPIB — これを選択した場合、バストリガによってチャンネルスペーシングがコントロールされます。バスを通じてトリガ（GETまたは*TRG）が送信されるたびに、現在のチャンネルが開き、次のチャンネルが閉じます。バストリガの詳細は第5節を参照してください。

SELECT CHAN SPACINGメニューからバストリガを選択するには、カーソルをGPIB上に合わせ、ENTERキーを押します。ディスプレイがCHANNEL CONTROLメニューに戻ります。

注：バストリガを選択すると、フロントパネルのSTEPキー（MANUAL参照）がアクティブになります。STEPキーを押すたびに、次にスキャンされるチャンネルが選択されます。

MANUAL — これを選択した場合、フロントパネルのSTEPキーによってチャンネルスペーシングがコントロールされます。STEPキーを押すたびに、現在のチャンネルが開き、次のチャンネルが閉じます。

CHANNEL SPACINGメニューからマニュアルトリガを選択するには、カーソルをMANUAL上に合わせ、ENTERキーを押します。ディスプレイがCHANNEL CONTROLメニューに戻ります。

注：フロントパネルのSTEPキーは、TIMER、EXTERNAL、GPIBまたはTRIGGER LINKを選択したときもアクティブになります。

TRIGLINK — これを選択した場合、Model 7001のTrigger Linkによってチャンネルスペーシングがコントロールされます。Trigger Linkは機能強化されたトリガシステムで、最大6ラインにより他の機器に対してトリガパルスを送受信するものです。Model 7001にトリガが加えられるたびに、現在のチャンネルが開き、次のチャンネルが閉じます。Trigger Linkの使い方の詳細は4.11項を参照してください。

注：Trigger Linkを選択すると、フロントパネルのSTEPキー（MANUAL参照）がアクティブになります。STEPキーを押すたびに、次にスキャンされるチャンネルが選択されます。

SELECT CHAN SPACINGメニューからTrigger Linkを選択するには、カーソルをTRIGLINK上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

```
SET TRIGGERLINK MODE
ASYNCHRONOUS      SEMI-SYNCHRONOUS
```

ASYNCHRONOUS — 入力トリガと出力トリガを別々のラインで伝送させる必要があるトリガコンフィギュレーションには、非同期トリガリンクを使います。以下のステップを実行することにより、非同期モードを選択し、また、Model 7001の入出力ラインを選択します。

1. SET TRIGGER LINK MODEメニューが表示されている状態で、カーソルをASYNCHRONOUS上に合わせ、ENTERキーを押します。下記のメッセージが表示されます。

```
SELECT INPUT LINE
#1 #2 #3 #4 #5 #6
```

カーソルの位置で、現在選択されている入力ラインが示されます。

2. Model 7001のトリガ入力ラインを選択するには、カーソルを目的のライン番号上に合わせ、ENTERキーを押します。

下記のメッセージが表示されます。

SELECT OUTPUT LINE

#1 #2 #3 #4 #5 #6

カーソルの位置で、現在選択されている出力ラインが表示されます。

3. Model 7001のトリガ出力ラインを選択するには、カーソルを別のライン番号上に合わせ、ENTERキーを押します。ディスプレイがSET TRIGGERLINK MODEメニューに戻ります。なお、入力と出力の両方に同一のラインを使うことはできません。

SEMI-SYNCHRONOUS — このモードでは、Model 7001の入力トリガと出力トリガは同一のラインに指定されます。以下のステップを実行することにより、半同期モードを選択し、また、トリガラインを選択します。

1. SET TRIGGERLINK MODEメニューが表示されている状態で、カーソルをSEMI-SYNCHRONOUS上に合わせ、ENTERキーを押します。下記のメッセージが表示されます。

SET SEMI-SYNC LINE

#1 #2 #3 #4 #5 #6

カーソルの位置で、現在選択されているトリガラインが表示されます。

2. Model 7001のトリガラインを選択するには、カーソルを目的のライン番号上に合わせ、ENTERキーを押します。ディスプレイがSET TRIGGERLINK MODEメニューに戻ります。

IMMEDIATE — これを選択した場合、チャンネルスペーシングは、イベント (TIMERトリガ、EXTERNALトリガなど) によってはコントロールされません。スキヤンが開始されると、Model 7001は、内部セットリング時間とユーザプログラムダイレイ (4.7項参照) が許す速さで迅速にステップスルーします。

SELECT CHAN SPACINGメニューから即時トリガを選択するには、カーソルをIMMEDIATE上に合わせ、ENTERキーを押します。ディスプレイがCHANNEL CONTROLメニューに戻ります。

HOLD — HOLDを選択すると、チャンネルスペーシングが抑制されます。そうすると、スキヤンは停止し、別のチャンネルスペーシングを選択することによってHOLDをキャンセルするまで再開されません。

SET CHAN SPACINGメニューからトリガホールドを選択するには、カーソルをHOLD上に合わせ、ENTERキーを押します。ディスプレイがCHANNEL CONTROLメニューに戻ります。

NUMBER OF CHANS — このメニュー項目は、スキヤンするチャンネルの数を定義するのに使います。このメニュー項目を選択するには、CHANNEL CONTROLメニューが表示されている状態で、カーソルをNUMBER-OF-CHANS上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

NUMBER OF CHANNELS

USE-SCANLIST-LENGTH CHAN-COUNT

USE-SCANLIST-LENGTH — これを選択した場合、スキヤンされるチャンネルの数は、スキヤンリスト中に定義されているチャンネルの数によって決まります。たとえば、スキヤンリストがチャンネル111、112、113、114で構成されていれば、スキヤンされるチャンネルの数は4です。NUMBER OF CHANNELSメニューからスキヤンリスト長カウントを選択するには、カーソルをUSE-SCANLIST-LENGTH上に合わせ、ENTERキーを押します。ディスプレイがCHANNEL CONTROLメニューに戻ります。

CHAN-COUNT — これを選択することで、スキヤンするチャンネルの数をユーザが決めます。ユーザプログラムカウントは、スキヤンリスト中のチャンネル数より小さくても、同等でも、大きくても構いません。たとえば、スキヤンリストが4チャンネル (111、112、113、114) で構成されている場合、ユーザは12のカウントをプログラムすることができます。このカウント値がプログラムされた場合、Model 7001はチャンネルスキヤンを3回繰り返します。チャンネルを (スキヤンというよりはむしろチャンネルを) 繰り返すことは、スキヤンレイヤでのオペレーション (SCAN CONTROL) のダイレイがなくなるという利点があります。全12チャンネル間のチャンネルスペーシングダイレイは同じです。

NUMBER OF CHANNELSメニューからCHAN-COUNTを選択するには、カーソルをCHAN-COUNT上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

CHANNEL COUNT

INFINITE ENTER-CHAN-COUNT

INFINITE — これはスキヤンを連続的に繰り返すのに使います。このメニュー項目を選択するには、CHANNEL COUNTメニューが表示されている状態で、カーソルをINFINITE上に合わせ、ENTERキーを押します。ディスプレイがCHANNEL CONTROLメニューに戻ります。

ENTER-CHAN-COUNT — これはスキャンするチャンネルの数を定義するのに使います。以下のステップを実行することにより、有限カウントを定義します。

1. CHANNEL COUNTメニューが表示されている状態で、カーソルをENTER-CHAN-COUNT上に合わせ、ENTERキーを押します。現在のカウントを示すメッセージが表示されます。

CHANNEL COUNT=0010

このメッセージは、現在のカウントが10に設定されていることを表わします。

2. 別のカウント (1~9999) にプログラムするには、キーパッドを使ってカウント値を入力します。
3. 目的のカウント値が表示されたら、ENTERキーを押します。ディスプレイがCHANNEL CONTROLメニューに戻ります。

CONTROL — このメニュー項目は、ソースバイパスをイネーブルまたはディスエーブルするのに使います。ソースバイパスは、チャンネルスペーシングイベントを最初のチャンネルレイヤ通過時にバイパスさせるためのものです。このメニュー項目を選択するには、CHANNEL CONTROLメニューが表示されている状態で、カーソルをCONTROL上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

TRIGGER CONTROL
SOURCE ACCEPTOR

SOURCE — これを選択すると、ソースバイパスがイネーブルされます。チャンネルスペーシングイベントは、最初のチャンネルレイヤ通過時にバイパスされます。これにより、スキャンされる最初のチャンネルは、プログラムされたイベントを待つ必要なく閉じることができます。

ACCEPTOR — これを選択すると、ソースバイパスがディスエーブルされます。

4.8.2 SCAN CONTROL

スキャンコントロールは以下の操作に使います。

- スキャンスペーシングをコントロールするイベントを選択する。
- NUMBER-OF-SCANS (スキャン実行回数) を指定する。
- ソースバイパスをイネーブルまたはディスエーブルする。

以下のステップを実行することにより、SCAN CONTROLメニューを表示させます。

1. CONFIGURE SCANメニューを表示させる (4.8項の「一般規則」の#1参照)。
2. カーソルキー (<および>) を使ってカーソルをSCAN-CONTROL上に合わせ、ENTERキーを押して下記のメニューを表示させます。

SCAN CONTROL
SCAN-SPACING NUMBER-OF SCANS ►
◀ CONTROL

SCAN-SPACING — このメニュー項目は、スキャンの時間間隔をコントロールするイベントを選択するのに使います。このメニュー項目を選択するには、SCAN CONTROLメニューが表示されている状態で、カーソルをSCAN-SPACING上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

SELECT SCAN SPACING
TIMER EXTERNAL GPIB MANUAL ►
◀ TRIGLINK IMMEDIATE HOLD

TIMER — タイマ機能によりスキャンスペーシングをコントロールします。タイマはスキャンの時間間隔を設定するのに使います。タイマは、0から99999.999秒まで0.001秒 (1msec) 単位で設定できます。スキャンがトリガされて開始された後、次のスキャンはプログラムされた時間間隔が経過すると開始されます。ただし、プログラムされた時間間隔が1回のスキャンに要する時間より短い場合は、次のスキャンは前のスキャンが終了するまで開始されません。

注：タイマが選択されると、フロントパネルのSTEPキーがアクティブになります。スキャンの終了後、STEPキーを押すと、次のスキャンが開始されます (この場合、Model 7001がもう1回スキャンを行なうようにプログラムされているものとします。NUMBER OF SCANSを参照してください)。

以下のステップを実行することにより、タイマを使ってスキャンの時間間隔を設定します。

1. SELECT SCAN SPACINGメニューが表示されている状態で、カーソルをTIMER上に合わせ、ENTERキーを押します。現在設定されている時間間隔 (秒単位) を示すメッセージが表示されます。

INTERVAL=00000.001

2. 表示されている時間間隔のままにしておくのであれば、EXITまたはENTERキーを押します。ディスプレイがSELECT SCAN CONTROLメニューに戻ります。

3. 別の時間間隔を設定するには、キーパッドを使って値（秒単位）をキーインし、ENTERキーを押します。ディスプレイがSCAN CONTROLメニューに戻ります。

EXTERNAL — これを選択した場合、外部トリガによってスキャンスペーシングがコントロールされます。Model 7001にトリガが加えられると、オペレーションがチャンネルレイヤに入ります。外部トリガは、リヤパネル上の“EXTERNAL TRIGGER” BNCコネクタに加えられます。外部トリガの詳細は4.10項を参照してください。

注：外部トリガを選択すると、フロントパネルのSTEPキー（MANUAL参照）がアクティブになります。スキャンの終了後、STEPキーを押すと、次のスキャンが開始されます（この場合、Model 7001がもう1回スキャンを行なうようにプログラムされているものとします。NUMBER OF SCANSを参照してください）。

SELECT SCAN SPACINGメニューから外部トリガを選択するには、カーソルをEXTERNAL上に合わせ、ENTERキーを押します。ディスプレイがSCAN CONTROLメニューに戻ります。

GPIB — これを選択した場合、バストリガによってスキャンスペーシングがコントロールされます。Model 7001がバストリガ（GETまたは*TRG）を受信すると、オペレーションがチャンネルレイヤに入ります。バストリガの詳細は第5節を参照してください。

注：バストリガを選択すると、フロントパネルのSTEPキー（MANUAL参照）がアクティブになります。スキャンの終了後、STEPキーを押すと、次のスキャンが開始されます（この場合、Model 7001がもう1回スキャンを行なうようにプログラムされているものとします。NUMBER OF SCANSを参照してください）。

SELECT SCAN SPACINGメニューからバストリガを選択するには、カーソルをGPIB上に合わせ、ENTERキーを押します。ディスプレイがSCAN CONTROLメニューに戻ります。

MANUAL — これを選択した場合、フロントパネルのSTEPキーによってスキャンスペーシングがコントロールされます。STEPキーを押すと、オペレーションがチャンネルレイヤに入ります。

注：フロントパネルのSTEPキーは、TIMER、EXTERNAL、GPIBまたはTRIGGER LINKを選択したときもアクティブになります。

SELECT SCAN SPACINGメニューからマニュアルトリガを選択するには、カーソルをMANUAL上に合わせ、ENTERキーを押します。ディスプレイがSCAN CONTROLメニューに戻ります。

TRIGLINK — これを選択した場合、Model 7001のTrigger Linkによってスキャンスペーシングがコントロールされます。Trigger Linkは機能強化されたトリガシステムで、最大6ラインにより他の機器に対してトリガパルスを送受信するものです。Model 7001がTrigger Linkを通じてトリガを受信すると、オペレーションがチャンネルレイヤに入ります。Trigger Linkの使い方の詳細は4.11項を参照してください。

注：Trigger Linkを選択すると、フロントパネルのSTEPキー（MANUAL参照）がアクティブになります。スキャンの終了後、STEPキーを押すと、次のスキャンが開始されます（この場合、Model 7001がもう1回スキャンを行なうようにプログラムされているものとします。NUMBER OF SCANSを参照してください）。

SELECT SCAN SPACINGメニューからTrigger Linkを選択するには、カーソルをTRIGLINK上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

```
SELECT INPUT LINE
#1 #2 #3 #4 #5 #6
```

カーソルの位置で、現在選択されている入力ラインが示されます。

Model 7001のトリガ入力ラインを選択するには、カーソルを目的のライン番号上に合わせ、ENTERキーを押します。下記のメッセージが表示されます。

```
SELECT OUTPUT LINE
#1 #2 #3 #4 #5 #6
```

カーソルの位置で、現在選択されている入力ラインが示されます。

Model 7001のトリガ出力ラインを選択するには、カーソルを別のライン番号上に合わせ、ENTERキーを押します。なお、入力と出力の両方に同一のラインを使うことはできません。

IMMEDIATE — IMMEDIATEを選択した場合、オペレーションは即時にチャンネルレイヤに入ります。SELECT SCAN SPACINGメニューから即時スキャンを選択するには、カーソルをIMMEDIATE上に合わせ、ENTERキーを押します。ディスプレイがSCAN CONTROLメニューに戻ります。

HOLD — HOLDを選択すると、スキャンスペーシングが抑制されます。そうすると、オペレーションは、他のスキャンスペーシングのどれか1つを選択することによってHOLDをキャンセルするまで再開されません。SELECT SCAN SPACINGメニューからHOLDを選択するには、カーソルをHOLD上に合わせ、ENTERキーを押します。ディスプレイがSCAN CONTROLメニューに戻ります。

NUMBER-OF-SCANS — このメニュー項目は、Model 7001によって実行されるスキャンの回数を定義するのに使います。このメニュー項目を選択するには、SCAN CONTROLメニューが表示されている状態で、カーソルをNUMBER-OF-SCANS上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

NUMBER OF SCANS
INFINITE ENTER-SCAN-COUNT

INFINITE — これはスキャンを連続的に繰り返すのに使います。NUMBER OF SCANSメニューから連続スキャンを選択するには、カーソルをINFINITE上に合わせ、ENTERキーを押します。ディスプレイがSCAN CONTROLメニューに戻ります。

ENTER-SCAN-COUNT — これを選択することで、スキャンの実行回数をユーザが決めます。Model 7001に対して1~9999回のスキャンを行なうようにプログラムすることができます。たとえば、10のスキャンカウントを入力すれば、プログラムされたスキャンが10回行なわれます。以下のステップを実行することにより、スキャンカウントを入力します。

1. NUMBER OF SCANSメニューが表示されている状態で、カーソルをENTER-SCAN-COUNT上に合わせ、ENTERキーを押します。現在のスキャンカウントを示すメッセージが表示されます。

SCAN COUNT=0001

このスキャンカウントは、Model 7001がスキャンを1回実行することを表わします。値が“0000”であれば、スキャンカウントが無限に設定されているということです。

2. 別のスキャンカウント (1~9999) にプログラムするには、キーパッドを使ってカウント値をキーインします。
3. 目的のカウント値が表示されたら、ENTERキーを押します。ディスプレイがSCAN CONTROLメニューに戻ります。

CONTROL — このメニュー項目は、ソースバイパスをイネーブルまたはディスエーブルするのに使います。

ソースバイパスは、スキャンスペーシングイベントを最初のスキャンレイヤ通過時にバイパスさせるためのものです。このメニュー項目を選択するには、SCAN CONTROLメニューが表示されている状態で、カーソルをCONTROL上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

TRIGGER CONTROL
SOURCE ACCEPTOR

SOURCE — これを選択すると、ソースバイパスがイネーブルされます。スキャンスペーシングイベントは、最初のスキャンレイヤ通過時にバイパスされます。これにより、オペレーションは、プログラムされたイベントを待つ必要なくチャンネルレイヤに入ることができます。

ACCEPTOR — これを選択すると、ソースバイパスがディスエーブルされます。

4.8.3 ARM CONTROL

アームコントロールは以下の操作に使います。

- スキャナのアームイベント (ARM SPACING) を選択する。
- スキャナのアーム回数 (ARM COUNT) を指定する。
- ソースバイパスをイネーブルまたはディスエーブルする。

以下のステップを実行することにより、ARM CONTROLメニューを表示させます。

1. CONFIGURE SCANメニューを表示させます (4.8項の「一般規則」の#1参照)。
2. カーソルキー (<および>) を使ってカーソルをARM CONTROL上に合わせ、ENTERキーを押して下記のメニューを表示させます。

SETUP ARM CONTROL
ARM-SPACING ARM-COUNT ▶
◀ TRIGGER-CONTROL

ARM-SPACING — このメニュー項目は、アームスペーシングをコントロールするイベントを選択するのに使います。このメニュー項目を選択するには、カーソルをARM-SPACING上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

ARM SPACING
MANUAL IMMEDIATE GPIB ▶
◀ EXTERNAL TRIGLINK HOLD

MANUAL — これを選択した場合、フロントパネルのSTEPキーによってアームスペーシングがコントロールされます。STEPキーを押すと、オペレーションがスキャンレイヤに入ります。

注：フロントパネルのSTEPキーは、TIMER、EXTERNAL、GPIBまたはTRIGGER LINKを選択したときもアクティブになります。

SELECT ARM SPACINGメニューからマニュアルトリガ（STEPキー）を選択するには、カーソルをMANUAL上に合わせ、ENTERキーを押します。ディスプレイがSETUP ARM CONTROLメニューに戻ります。

IMMEDIATE — これを選択した場合、オペレーションは即時にスキャンレイヤに入ります。

SELECT ARM SPACINGメニューから即時トリガを選択するには、カーソルをIMMEDIATE上に合わせ、ENTERキーを押します。ディスプレイがSETUP ARM CONTROLメニューに戻ります。

GPIB — これを選択した場合、バストリガによってアームスペーシングがコントロールされます。Model 7001がバストリガ（GETまたは*TRG）を受信すると、オペレーションが即時にスキャンレイヤに入ります。バストリガの詳細は第5節を参照してください。

注：バストリガを選択すると、フロントパネルのSTEPキー（MANUAL参照）がアクティブになります。STEPキーを押すと、オペレーションがスキャンレイヤに入ります。

SELECT ARM SPACINGメニューからバストリガを選択するには、カーソルをGPIB上に合わせ、ENTERキーを押します。ディスプレイがSETUP ARM CONTROLメニューに戻ります。

EXTERNAL — これを選択した場合、外部トリガによってアームスペーシングがコントロールされます。Model 7001にトリガが加えられると、オペレーションがスキャンレイヤに入ります。外部トリガは、リヤパネル上の“EXTERNAL TRIGGER” BNCコネクタに加えられます。外部トリガの詳細は4.10項を参照してください。

注：外部トリガを選択すると、フロントパネルのSTEPキー（MANUAL参照）がアクティブになります。STEPキーを押すと、オペレーションがスキャンレイヤに入ります。

SELECT ARM SPACINGメニューから外部トリガを選択するには、カーソルをEXTERNAL上に合わせ、ENTERキーを押します。ディスプレイがSETUP ARM CONTROLメニューに戻ります。

TRIGLINK — これを選択した場合、Model 7001のTrigger Linkによってアームスペーシングがコントロールされます。Trigger Linkは機能強化されたトリガシステムで、最大6ラインにより他の機器に対してトリガパルスを送受信するものです。Model 7001がTrigger Linkを通じてトリガを受信すると、オペレーションがスキャンレイヤに入ります。Trigger Linkの使い方の詳細は4.11項を参照してください。

注：Trigger Linkを選択すると、フロントパネルのSTEPキー（MANUAL参照）がアクティブになります。STEPキーを押すと、オペレーションがスキャンレイヤに入ります。

SELECT ARM SPACINGメニューからTrigger Linkを選択するには、カーソルをTRIGLINK上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

SELECT INPUT LINE
#1 #2 #3 #4 #5 #6

カーソルの位置で、現在選択されている入力ラインが示されます。

Model 7001のトリガ入力ラインを選択するには、カーソルを目的のライン番号上に合わせ、ENTERキーを押します。下記のメッセージが表示されます。

SELECT OUTPUT LINE
#1 #2 #3 #4 #5 #6

カーソルの位置で、現在選択されている出力ラインが示されます。

Model 7001のトリガ出力ラインを選択するには、カーソルを別のライン番号上に合わせ、ENTERキーを押します。なお、入力と出力の両方に同一のラインを使うことはできません。

HOLD — HOLDを選択すると、アームスペーシングが抑制されます。そうすると、オペレーションは、他のアームスペーシングのどれか1つを選択することによってHOLDをキャンセルするまで再開されません。SELECT ARM SPACINGメニューからHOLDを選択するには、カーソルをHOLD上に合わせ、ENTERキーを押します。ディスプレイがSETUP ARM CONTROLメニューに戻ります。

ARM-COUNT — このメニュー項目は、オペレーションがアームレイヤに戻る回数を定義するのに使います。このメニュー項目を選択するには、SETUP ARM CONTROLメニューが表示されている状態で、カーソルをARM-COUNT上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

ARM COUNT

INFINITE ENTER-ARM-COUNT

INFINITE — これは、オペレーションが連続的にアームレイヤに戻るようにするのに使います。ARM COUNTメニューから連続アームを選択するには、カーソルをINFINITE上に合わせ、ENTERキーを押します。ディスプレイがSETUP ARM CONTROLメニューに戻ります。

ENTER-ARM-COUNT — これを選択することで、オペレーションがアームレイヤに戻る回数をユーザが決めます。Model 7001に対して最大9999回のアームを行なうようにプログラムすることができます。以下のステップを実行することにより、アームカウントを入力します。

1. ARM COUNTメニューが表示されている状態で、カーソルをENTER-ARM-COUNT上に合わせ、ENTERキーを押します。現在のアームカウントを示すメッセージが表示されます。

ARM COUNT=0001

このアームカウントは、スキャナが1回アームされることを表わします。アームカウントが“0000”であれば、無限が選択されているということです。

2. 別のカウント (1~9999) にプログラムするには、キーボードを使ってカウント値をキーインします。
3. 目的のカウント値が表示されたら、ENTERキーを押します。ディスプレイがSETUP ARM CONTROLメニューに戻ります。

TRIGGER CONTROL — このメニュー項目は、ソースバイパスをイネーブルまたはディスエーブルするのに使います。ソースバイパスは、アームスレーシングイベントを最初のアームレイヤ通過時にバイパスさせるためのものです。このメニュー項目を選択するには、SETUP ARM CONTROLメニューが表示されている状態で、カーソルをTRIGGER-CONTROL上に合わせ、ENTERキーを押します。下記のメニューが表示されます。

TRIGGER CONTROL
SOURCE ACCEPTOR

SOURCE — これを選択すると、ソースバイパスがイネーブルされます。アームスレーシングイベントは、最初のアームレイヤ通過時にバイパスされます。これにより、オペレーションは、プログラムされたイベントを待つ必要なくスキャンレイヤに入ることができます。

ACCEPTOR — これを選択すると、ソースバイパスがディスエーブルされます。

4.8.4 チャンネル制限

Model 7001は、次のようなチャンネル制限ができるようにプログラムすることができます。

- SINGLE CHAN (シングルチャンネル) モードをイネーブルし、複数のチャンネルが同時に閉じないようにする。
- RESTRICTED CHANNELSのリストを定義する。このリスト中のチャンネルは閉じることはできません。

以下のステップを実行することにより、CHANNEL RESTRICTIONSメニューを表示させます。

1. CONFIGURE SCANメニューを表示させます (4.8項の「一般規則」の#1参照)。
2. カーソルキー (<および>) を使ってカーソルをCHANNEL RESTRICTIONS上に合わせ、ENTERキーを押して下記のメニューを表示させます。

CHANNEL RESTRICTIONS
SINGLE-CHAN RESTRICTED-CHANNELS

SINGLE-SCAN — このメニュー項目は、シングルチャンネルモードをイネーブルするのに使います。シングルチャンネルモードがイネーブル (ON) されたら、1チャンネルしか閉じることができません。複数のチャンネルを同時に閉じることができません。

以下のステップを実行することにより、シングルチャンネルモードをイネーブルまたはディスエーブルします。

1. CHANNEL RESTRICTIONSメニューが表示されている状態で、カーソルをSINGLE-CHAN上に合わせ、ENTERキーを押します。下記のメッセージが表示されます。

SINGLE CHANNEL MODE
OFF ON

カーソルの位置で、シングルチャンネルモードの現在のステータスが示されます。

- カーソルを目的の状態 (ONまたはOFF) 上に合わせ、ENTERキーを押します。ディスプレイがCHANNEL RESTRICTIONSメニューに戻ります。

操作上の注意事項：

- シングルチャンネルモードでは、チャンネルリストに複数のチャンネルが指定されている場合、CLOSEキーを押すと、一番若い番号のスロットの一番若い番号のチャンネルだけが閉じます。たとえば、次のようなチャンネルリストが定義されているとして、

SELECT CHANNELS 2!1,2!2,1!5

When CLOSE is pressed,Channel 1!5 will close.

CLOSEキーを押すと、チャンネル1!5が閉じます。

- シングルチャンネルモードをイネーブルすると、ストアされている全チャンネルパターンがクリアされます。たとえば、メモリロケーション#001 (M1) に閉チャンネル1!1および1!2がストアされている場合、シングルチャンネルモードをイネーブルすると、そのメモリロケーションがクリアされます。
- 現在、複数のチャンネルが閉じている場合、シングルチャンネルモードはイネーブルできません。

RESTRICTED-CHANNELS — このメニュー項目は、フロントパネルから、またはバスを通じて閉じてはならないチャンネルを指定するのに使います。主に、不注意によりチャンネルが閉じるのを防止するための安全機構として用いられます。誤ってチャンネルを閉じると、機器あるいはDUTが損傷することがあります (たとえば、マトリックステストシステムの電源ショート)。

以下のステップを実行することにより、制限チャンネルを定義します。

- CHANNEL RESTRICTIONSメニューが表示されている状態で、カーソルをRESTRICTED-CHANNELS上に合わせ、ENTERキーを押します。下記のメッセージが表示されます。

RESTRICTED CHANNELS
SELECT CHANNELS

- キーパッドを使ってチャンネルリストを入力し、EXITキーを押します。

4.9 デジタルI/Oポート

Model 7001は、外部デジタル回路をコントロールするのに使用できるデジタルI/Oポートを組み込んでいます。このポートでは、4つの出力チャンネルと1つの入力チャンネルを使えます。

4.9.1 デジタル出力チャンネル

Model 7001には4つのデジタル出力チャンネルがあります。それぞれのオープンコレクタ出力はHigh (+5V) またはLow (0V) に設定でき、100mAまでシンクします。

デジタル出力チャンネルのレベルは、フロントパネルのMENU機能によって設定します。DIGITAL I/O MENUで、各出力チャンネルの出力ステータス (オンまたはオフ) および各チャンネルの極性 (アクティブハイまたはアクティブロー) を設定します。極性をアクティブハイにするときは、“オン” 出力チャンネルを+5Vに設定します。逆に、極性をアクティブローにするときは、“オン” 出力チャンネルを0Vに設定します。出力ポートの設定について詳細は4.6項を参照してください。

内部5V (TTL) レベルに代えてユーザ供給電圧 (最大 30V) を使うこともできます。異なるレベルを使うには、その電圧をI/Oコンセントのピン3に加えるだけです (図4-9参照)。

4.9.2 デジタル入力チャンネル

Model 7001には、TTL入力レベルを読み取るための1つのデジタル入力チャンネルがあります。TTL入力レベルがHighであれば、それは“ON”と読み取られます。

フロントパネルのMENU機能は、デジタル入力チャンネルのレベルを読み取ります。

4.9.3 I/Oポートの接続

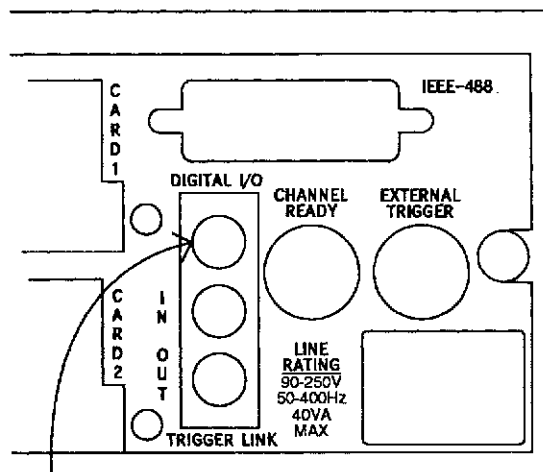
DIGITAL I/Oポートは図4-9に示すようにリヤパネル上にあります。この図には、8ピン・マイクロDINフィメールコンセントのピン配列も示されています。DIGITAL I/OコンセントはTRIGGER LINKコンセントと同じですから、Trigger Linkケーブル (Keithley Model 8501) を接続に使うことができます。

Trigger Linkケーブル2本に切断することにより、非終端の端部を外部デジタル回路にダイレクトにハード配線することができます。

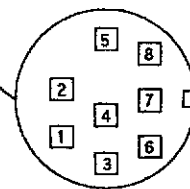
8ピン・マイクロDINフィメールコンセント (リヤパネルをご覧ください。Keithley Model 8501) に外部回路をハード配線することにより、1本のTrigger LinkケーブルをそのままデジタルI/O接続に使うことができます。Trigger LinkケーブルをDIGITAL I/Oポートに使用する場合はピン配列を図4-10に示します。

注意

Trigger LinkとデジタルI/Oポートは同タイプのコネクタを使います。損傷防止のため、Digital I/OをTrigger Linkに接続することはやめてください。また、Digital I/Oに外部回路を接続するときは、誤ってTrigger Linkに接続することのないよう注意してください。



デジタルI/Oコンセント



- 1 = 使用せず
- 2 = デジタル入力
- 3 = 電圧入力 (最大340V)
- 4 = デジタル出力#1
- 5 = デジタル出力#2
- 6 = デジタル出力#3
- 7 = デジタル出力#4
- 8 = コモン

図4-9

デジタルI/Oポート

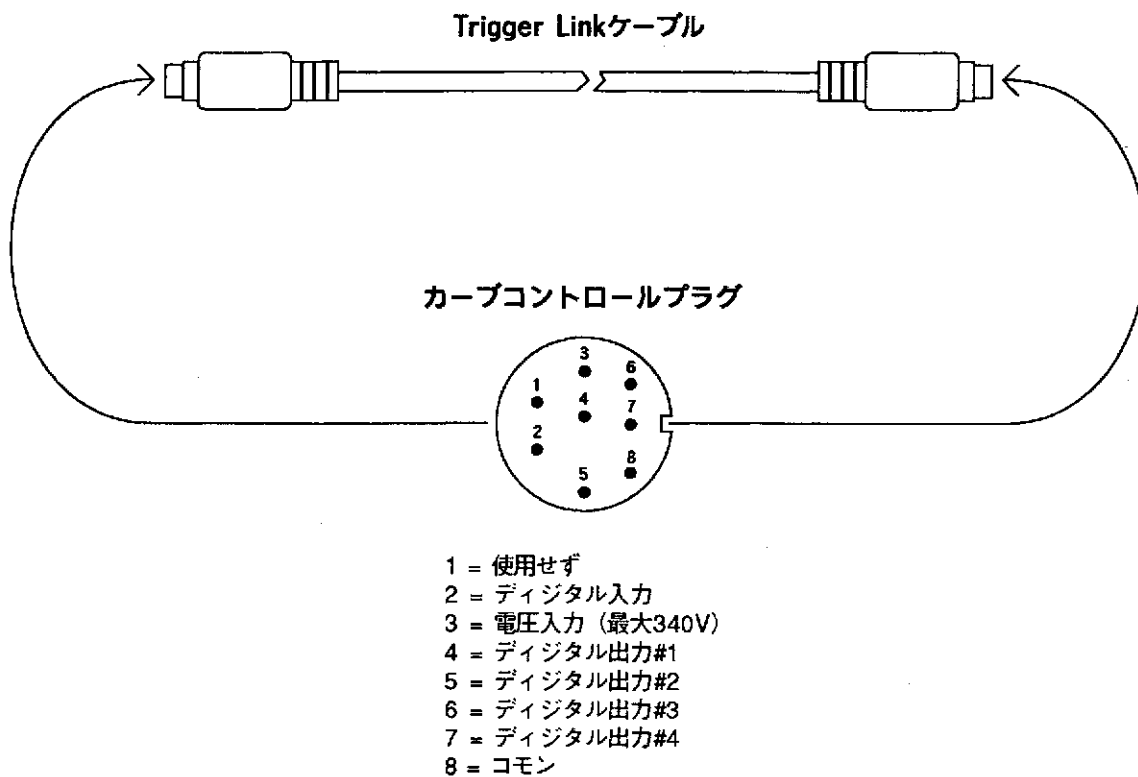


図4-10

Trigger Linkケーブルの使用によるデジタルI/O接続

4.10 外部トリガ

Model 7001は、外部トリガ関係のコネクタとして、リヤパネル上にBNCコネクタを装備しています(図4-11参照)。EXTERNAL TRIGGER入力ジャックでは、他の機器からModel 7001をトリガすることができ、CHANNEL READY出力ジャックでは、Model 7001から他の機器をトリガすることができます。

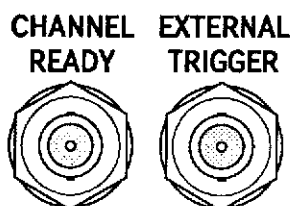


図4-11
外部トリガ用コネクタ (BNC)

4.10.1 外部トリガ

EXTERNAL TRIGGER入力ジャックは、図4-12に示す仕様の、TTLコンパチブルのフォーリングエッジパルスを必要とします。

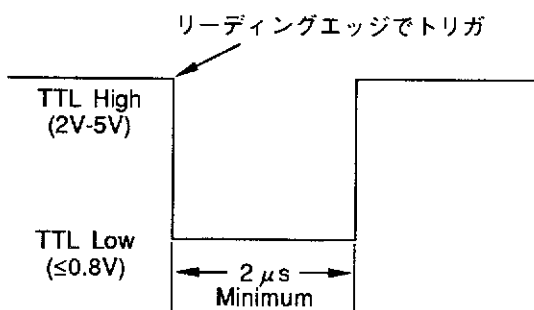


図4-12
外部トリガおよび非同期Trigger Link入力パルス仕様

原則として、外部トリガは、スキャンオペレーションをコントロールするためのイベントとして使うことができます。Model 7001が外部トリガに応答するためには、スキャンオペレーションのレイヤをそれが可能なように設定しなければなりません。3つのレイヤのプログラムの仕方については4.8項に説明してあります。

4.10.2 チャンネルレディ

CHANNEL READY出力ジャックは、他の機器をトリガするのに使うことができるTTLコンパチブルの出力パルスを発生します。このトリガパルスの仕様を図4-13に示します。

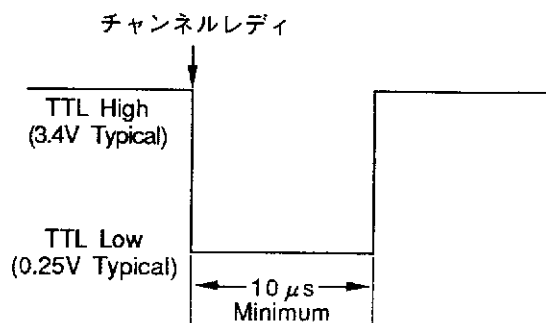


図4-13
チャンネルレディおよび非同期Trigger Link出力パルス仕様

普通、トリガを出力するのは、各チャンネルが閉じ、安定した後にしたいはずですが(セッティング時間には、内部的に設定されたリレーセッティング時間とユーザプログラムデレイが含まれます。4.7項参照)。出力トリガは、チャンネルスペーシングがExternalに設定されている限り、各チャンネルがスキャンされた後に発生します。チャンネルレイヤのプログラムの仕方について詳細は4.8.1項を参照してください。

Model 7001は、スキャンレイヤおよび/またはアームレイヤに入っている状態でトリガを出力することもできます。図4-8は、これらのトリガがトリガモデルのどこで発生するかを示しています。スキャンレイヤのソースバイパスがイネーブルされていて(Control = Source)、スキャンスペーシングがExternalにプログラムされている場合、出力トリガはスキャンレイヤの各リターンパスで発生します。また、アームレイヤのソースバイパスがイネーブルされていて(Control = Source)、アームスペーシングがExternalにプログラムされている場合、出力トリガはアームレイヤの各リターンパスで発生します。スキャンレイヤとアームレイヤのプログラムの仕方について詳細は4.8.3項を参照してください。

4.10.3 外部トリガ例

標準的なテストシステムでは、チャンネルを閉じてから、そのチャンネルに接続されたDUTをKeithley Model 196のようなDDMで測定することができます。このようなテストシステムを図4-14に示します。この図のシステムは、Model 7011マルチプレクサカードを用いて、10のDUTをDMMに対してスイッチングするものです。

このテストシステムの外部トリガの接続を図4-15に示します。Model 7001のChannel Ready (出力) はModel 196のExternal Trigger Inputに接続され、Model 7001のExternal Trigger (入力) はModel 196のVoltmeter Complete Outputに接続されます。

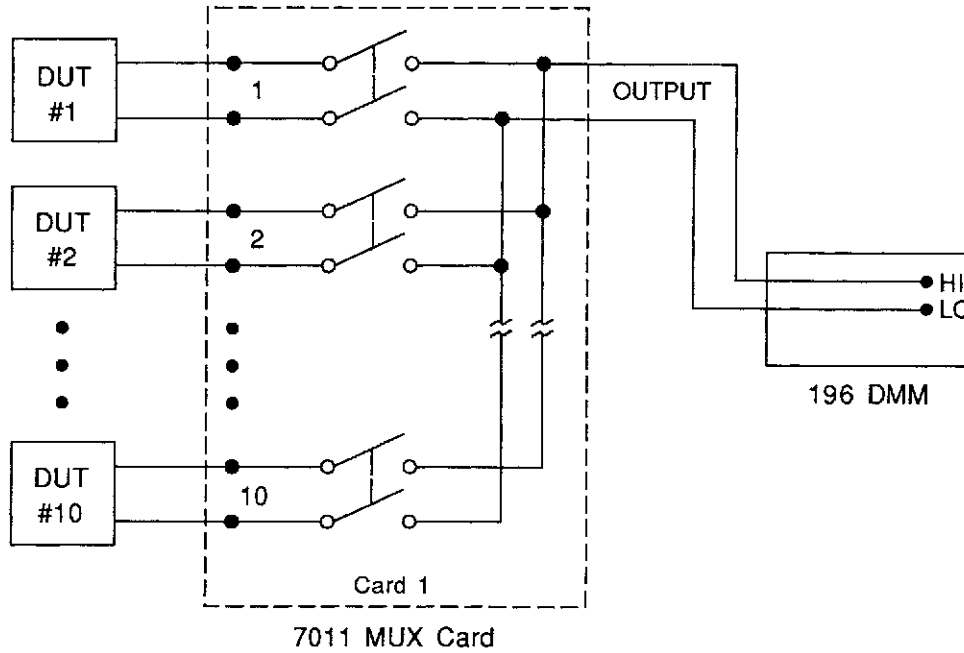


図4-14
DUTテストシステム

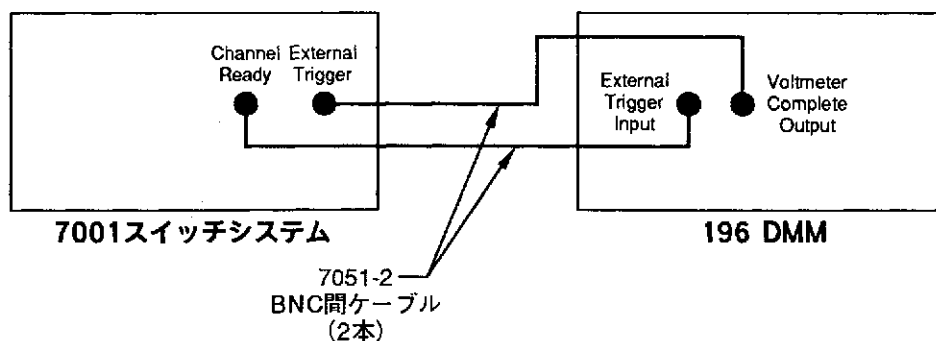


図4-15
外部トリガコネクタ

このテストシステムの場合、Model 196は外部トリガに設定され、スキャンは次のように設定されます。

スキャンリスト = 1!1-1!10

アームレイヤ:

アームスペーシング = Immediate*
アームカウント = 1*
アームトリガコントロール = Acceptor*

スキャンレイヤ:

スキャンスペーシング = Immediate*
スキャン回数 = 1
スキャントリガコントロール = Acceptor*

チャンネルレイヤ:

チャンネルスペーシング = External
チャンネル数 = Use Scan List Length*
チャンネルトリガコントロール = Source*

* 該設定がRESET (および出荷時) デフォルトであることを表わす。

チャンネルトリガコントロールがSourceに設定されていることに注意してください。これを選択した場合、スキャンオペレーションの初めに、最初のチャンネルを閉じるための外部トリガを必要としません。アームスペーシングとスキャンスペーシングはImmediateに設定されていますから、STEPキーを押してスキヤナをアイドル状態から抜け出させると、直ちにスキャンが開始されます。

フロントパネルのSTEPキーを押すと、スキヤナがアームされ、最初のチャンネルを閉じます。チャンネル1!1が安定した後 (チャンネルレディ)、Model 7001のChannel ReadyからModel 196のExternal Trigger Inputへトリガが送られ、DUT#1の測定を開始させます。Model 196は、測定を完了した後、Voltmeter Complete OutputからModel 7001のExternal Triggerへのトリガを出力します。これにより次のチャンネルが閉じます。チャンネル1!2が安定した後、Model 196へトリガが送られ、DUT#2の測定を開始させます。測定が完了すると、Model 7001へトリガが送られ、次のチャンネルが閉じます。このチャンネルが安定した後、DMMへトリガ送られ、測定を開始させます。全10チャンネルがスキャンされ、測定されるまで、このプロセスが継続されます。測定結果をその都度ストアするのに、Model 196のデータ記憶装置を使うこともできます。

4.11 Trigger Link [トリガリンク]

Model 7001は、Trigger Linkを用いた強力な外部トリガ機能を備えています。Trigger Linkは6ラインを有し、最大6つの機器をこのバストリガを通じてコントロールすることができます。Trigger Linkに使われているマイクロ8ピンDINソケットを図4-16に示します。

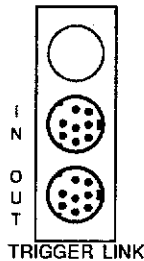


図4-16
Trigger Linkコネクタ

注

リヤパネル上の2つのTrigger Linkコネクタは、実際には互いに並列接続されています。したがって、INラベルとOUTラベルは誤った呼び方です。Trigger Linkを別の機器に接続するときはどちらのコネクタを使っても構いません。

注意

Trigger LinkとデジタルI/Oポートは同タイプのコネクタを使います。損傷防止のため、Digital I/OをTrigger Linkに接続することはやめてください。また、Digital I/Oに外部回路を接続するときは、誤ってTrigger Linkに接続することのないよう注意してください。

通常、Model 7001へのTrigger Link入力トリガは、スキャンオペレーションをコントロールするのに使います。

Model 7001がTrigger Linkコンパチブルのトリガにตอบสนองするためには、スキャンオペレーションのレイヤをそれが可能なようにプログラムしなければなりません。たとえば、Trigger Link入力トリガでチャンネルスキャンプロセスをコントロールしたい場合、チャンネルスペーシングをTRIGLINKトリガイベントにプログラムしなければなりません。現在選択されているチャンネルを測定するために別の機器をトリガするときは、Model 7001からのTrigger Link出力トリガを使うのが普通です。

Trigger Linkには、非同期モードと半同期モードの2つの動作モードがあります。非同期モードでは、入力トリガと出力トリガに別々のラインを使い、半同期モードでは、入力トリガと出力トリガのどちらにも同じラインを使います。

4.11.1 非同期オペレーション

Trigger Linkは、非同期動作モードでも、基本的には、外部トリガの場合（4.10項参照）と同様に動作します。外部トリガの場合と同様、非同期モードでは、入力トリガと出力トリガに別々のラインを使います。また、非同期モードでは、外部トリガの場合と同じTTLコンパチブルパルスを使います。非同期モードの場合の入力トリガ仕様は図4-12、出力トリガ仕様は図4-13にそれぞれ示してあります。

標準的な非同期オペレーションでは、チャンネルレイヤの設定として、チャンネルスペーシングをTRIGLINKに設定し、TriggerlinkモードをAsynchronousに設定します。入力トリガと出力トリガは6ラインのうちどれにも設定できますが、同じラインを使うことはできません。たとえば、入力トリガにライン#1を選択した場合、出力トリガは他の5ライン（#2～#6）のどれか1つを使わなければなりません。チャンネルレイヤでのオペレーション中、Trigger Link入力トリガはそれぞれ、スキャンされるチャンネルを閉じます。Model 7001は、リレーが安定し、ユーザプログラムダイヤ（4.7項参照）がタイムアウトになった後、Trigger Linkトリガを出力します（普通は、測定を行なうためのDMMへ出力）。チャンネルレイヤの設定には、CONFIGURE SCANメニューを使います（4.8.1項参照）。

スキャンレイヤおよび/またはアームレイヤもTRIGGER LINKにプログラムすることができます。この場合、チャンネルスペーシングはTRIGLINKに設定し、また、アームスペーシングもTRIGLINKに設定します。

これらのレイヤでTrigger Linkを使うときは、チャンネルレイヤで入出力ラインを選択したように、ここでも入出力ラインを選択しなければなりません。なお、チャンネルレイヤで選択したのと同じラインを使うことができますから、よく覚えておいてください。

非同期Trigger Link例 #1

標準的なテストシステムでは、チャンネルを閉じてから、そのチャンネルに接続されたDUTをメータで測定することができます。

このようなテストシステムを図4-17に示します。

この図のシステムは、Model 7011マルチプレクサカードを用いて、10のDUTをTrigger Link装備のメータに対してスイッチングするものです。

このテストシステムのTrigger Linkの接続を図4-18に示します。Model 7001のTrigger Linkは、メータのTrigger Linkに接続されます。なお、必要とするTrigger Linkケーブルは1本だけです。

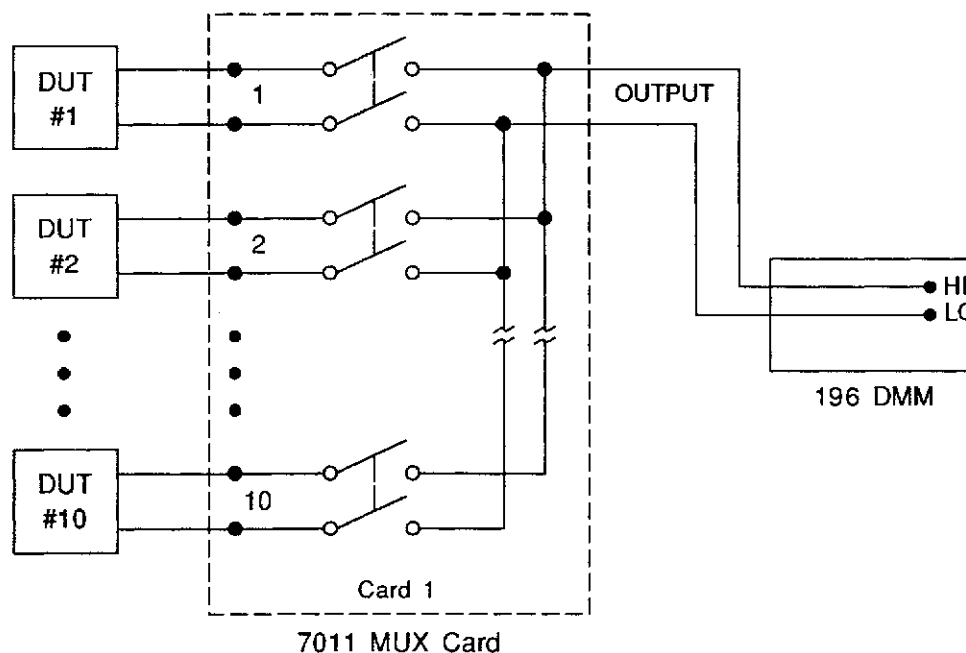


図4-17
DUTテストシステム

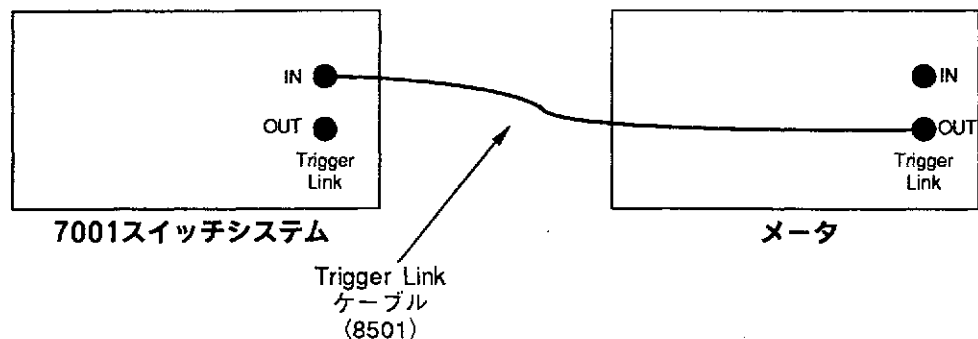


図4-18
トリガリンクコネクタ

この例では、Model 7001とメータは次のように設定されます。

Model 7001 :

スキャンリスト = 111-1!10

アームレイヤ :

アームスペーシング = Immediate*
アームカウント = 1*
アームトリガコントロール = Acceptor*

スキャンレイヤ :

スキャンスペーシング = Immediate*
スキャン回数 = 1
スキャントリガコントロール = Acceptor*

チャンネルレイヤ :

チャンネルスペーシング = Triglink
入力ライン = #2
出力ライン = #1
チャンネルトリガコントロール = Source*
チャンネル数 = Use Scanlist Length*

* 該設定がRESET (および出荷時) デフォルトであることを表わす。

メータ :

アームレイヤ :

アームソース = Immediate
アームカウント = 1
アームトリガコントロール = Acceptor

スキャンレイヤ :

スキャンソース = Immediate
スキャンカウント = 1
スキャントリガコントロール = Acceptor

メジャーレイヤ :

メジャーソース = Triglink
Trigger Linkモード = Asynchronous
入力ライン = #1
出力ライン = #2
メジャートリガコントロール = Acceptor
メジャーカウント = 10

Model 7001のチャンネルトリガコントロールがSourceに設定されていることに注意してください。これを選択した場合、スキャンオペレーションの初めに、最初のチャンネルを閉じるためのTrigger Linkトリガを必要としません。アームスペーシングとスキャンスペーシングはImmediateに設定されていますから、STEPキーを押してスキャナをアイドル状態から抜け出させると、直ちにスキャンが開始されます (この場合、メータもアイドル状態から抜け出ているものとします)。

テストを実行するには、メータ上のTRIGキーを押してメータをアイドル状態から抜け出させ、Model 7001上のSTEPキーを押します。以下に、図4-19に示すオペレーションモデルを参照しながら、オペレーションについて説明します。

Ⓐ TRIGキーを押すと、メータがアイドル状態から抜け出し、メータオペレーションが流れ図中のA点にきます。ここでTrigger Linkトリガを待ちます。アームレイヤとスキャンレイヤがどちらもImmediate Sourceにプログラムされていますから、オペレーションは直ちにメジャーレイヤ (A点) に下ります。

Ⓑ STEPキーを押すと、Model 7001がアイドル状態から抜け出し、オペレーションが流れ図中のB点にきます。アームレイヤとスキャンレイヤがどちらもImmediate Spacingにプログラムされていますから、オペレーションはチャンネルレイヤ (B点) に下ります。

Ⓒ チャンネルトリガコントロールがSourceに設定されていますから、B点ではトリガを待ちません。“Trigger Linkトリガを待つ”をバイパスし、最初のチャンネルを閉じます (C点)。なお、バイパスが有効なのは最初にモデルを通過するときだけです。

Ⓓ リレーのセッティング後、Model 7001はTrigger Linkトリガパルスを出力します (D点)。Model 7001は10チャンネルをスキャンするようにプログラムされていますから、オペレーションはB点までループバックし、ここで入力トリガを待ちます。なお、バイパスはもはや有効ではなくなっています。

ⒺおよびⒻ メータオペレーションがA点でトリガを待っていることを思い出してください。Model 7001からの出力トリガがメータをトリガし、DUT #1の測定を行なわせます (E点)。測定終了後、メータはトリガパルスを出力し (F点)、A点までループバックし、ここで別の入力トリガを待ちます。

メータからModel 7001へトリガが加えられると、次にスキャンされるチャンネルが閉じます。そうすると、メータがトリガされて次のDUTを測定します。全10チャンネルがスキャンされ、測定されるまで、このプロセスが続きます。

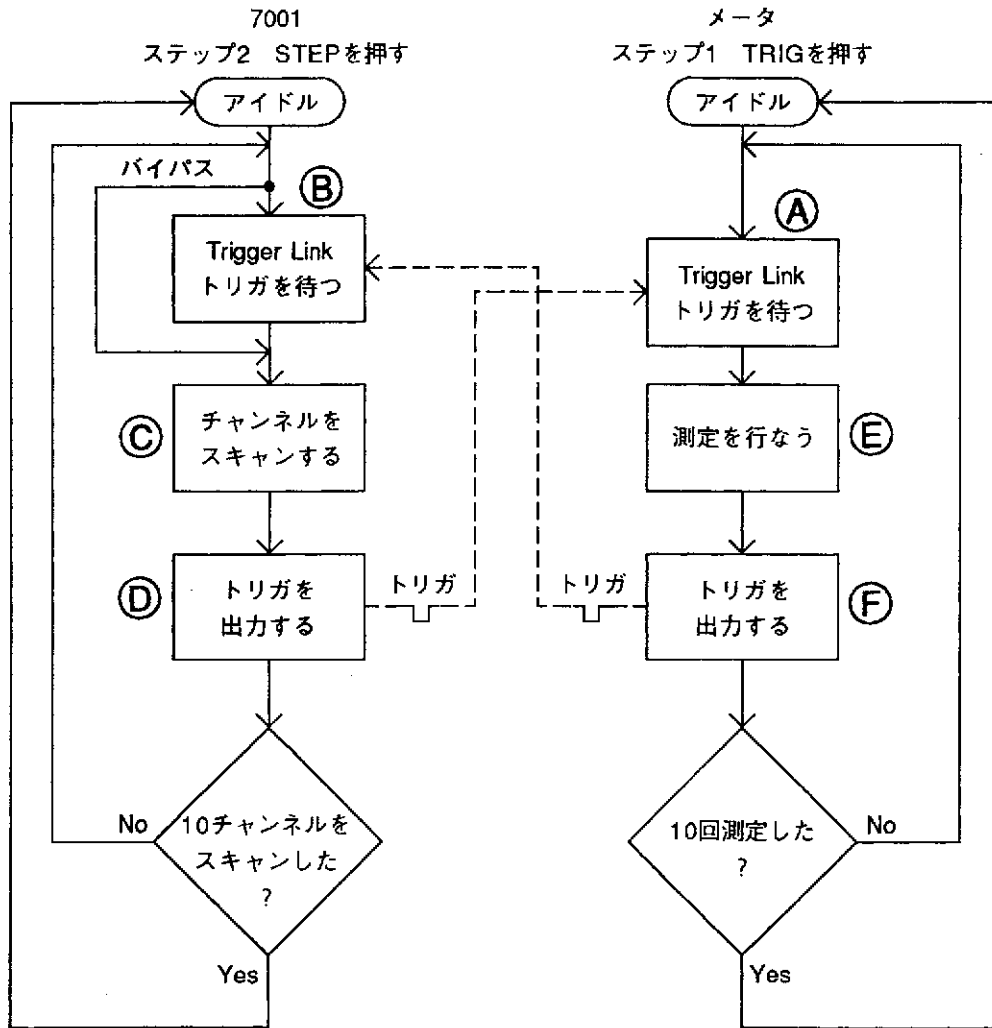


図4-19 非同期Trigger Link例 #1のオペレーションモデル

外部トリガとTrigger Link—

前に説明したように、非同期Trigger Linkのトリガパルスは、外部トリガに使われるトリガパルスと同じです。これらのパルスを同時に使うことを妨げるのは、接続の互換性がないという点です。Trigger Linkはマイクロ8ピンDINコネクタを使い、外部トリガはBNCコネクタを使います。

このような接続の問題は、Model 8502 Trigger Linkアダプタを使用することによって解決できます。このアダプタは、2個のマイクロ8ピンDINコネクタと6個のBNCコネクタを持っています。DINコネクタは、Trigger Linkケーブルを介して、Model 7001上のTrigger Linkコネクタに直接接続します。BNCコネクタは、標準のメールBNC間ケーブルを介して、外部トリガ用BNCコネクタに直接接続します。

アダプタを使ってKeithley Model 196をModel 7001のTrigger Linkに接続する方法を図4-20に示します。前の例（非同期Trigger Linkの例 #1）で、このアダプタを使うことにより、Trigger Link装備のメータをModel 196に置き換えることが可能です。Model 196を外部トリガに設定してModel 7001のSTEPキーを押すと、テストが開始されます。

非同期Trigger Link例 #2

この例では、テストシステム（図4-21）には、個々のDUTをModel 230電圧源によって与えられる異なった2つのバイアスレベルで測定するためのModel 196 DMMが含まれます。電圧源が第一の電圧レベルに設定されると、10チャンネルがスキャンされ、測定されます。次に、電圧源が第二の電圧レベルに設定され、再び10チャンネルがスキャンされ、測定されます。

この例では、Trigger Linkを装備していない機器（Model 196および230）を使いますから、Model 8502 Trigger Linkアダプタを必要とします。接続を図4-22に示します。

この例では、Model 196および230は外部トリガにプログラムされ、また、Model 230は第一の電圧レベルを発生するように設定されます。Model 7001は次のように設定されます。

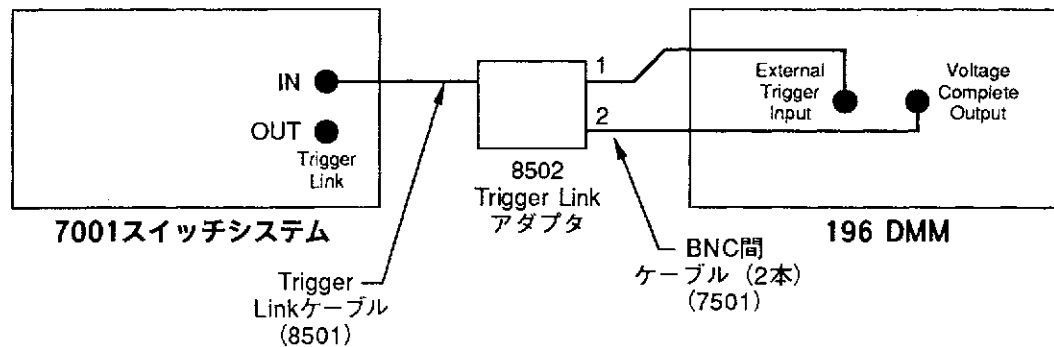


図4-20

Trigger Linkアダプタの使用による接続

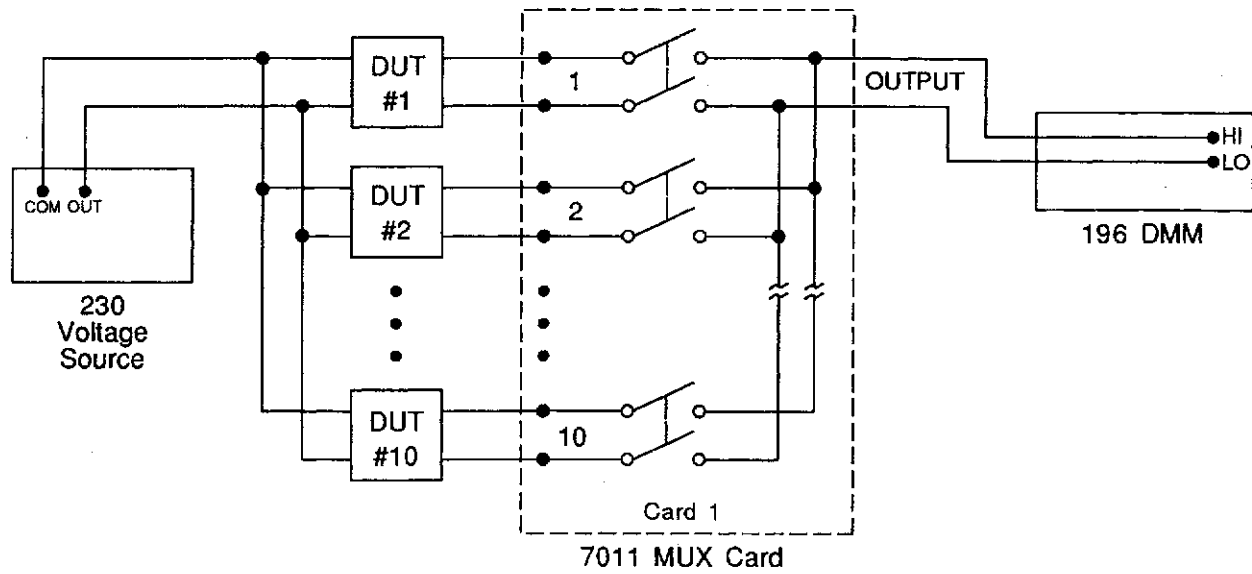


図4-21
DUTテストシステム (非同同期例 #2)

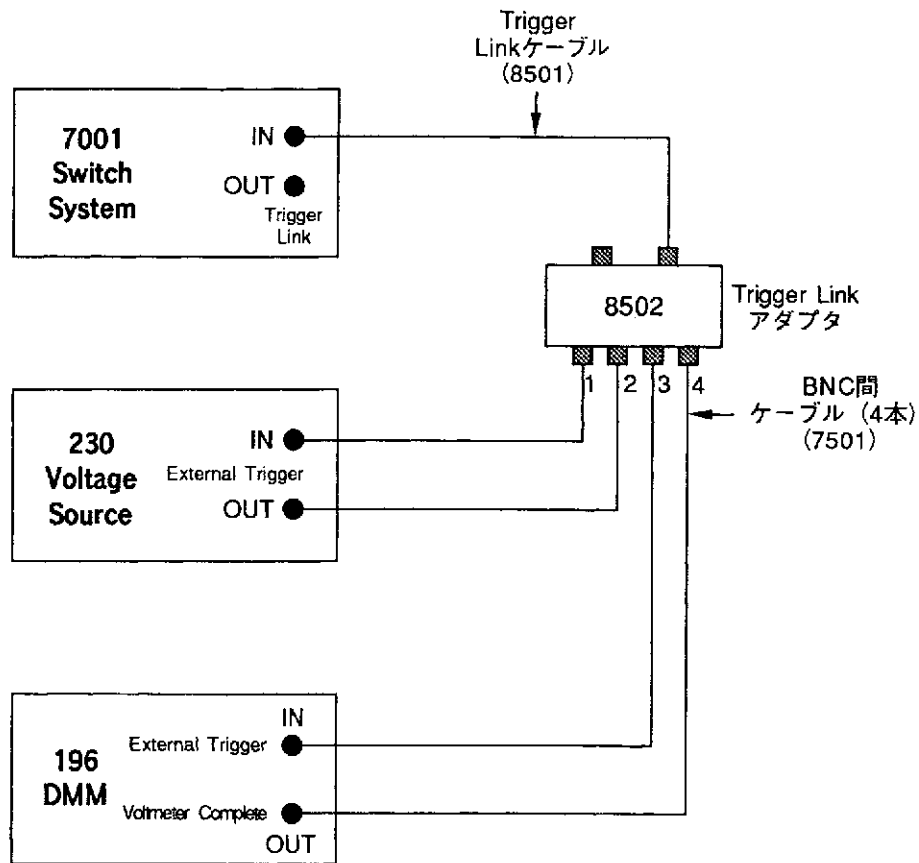


図4-22
Trigger Linkの使用による接続 (非同同期例 #2)

Model 7001 :

スキャンリスト = 1!1-1!10

アームレイヤ :

アームスペーシング = Immediate*
アームトリガコントロール = Acceptor*
アームカウント = 1*

スキャンレイヤ :

スキャンスペーシング = TrigLink
入力ライン = #2
出力ライン = #1
スキャントリガコントロール = Source
スキャン回数 = 2

チャンネルレイヤ :

チャンネルスペーシング = TrigLink
Trigger Linkモード = Asynchronous
入力ライン = #4
出力ライン = #3
チャンネルトリガコントロール = Source*
チャンネル数 = Use Scanlist Length*

* 該設定がRESET (および出荷時) デフォルトであることを表わす。

テストを実行するには、Model 7001上のSTEPキーを押します。以下に、図4-23に示すオペレーションモデルを参照しながら、オペレーションについて説明します。

(A) STEPキーを押すと、Model 7001がアイドル状態から抜け出し、オペレーションが流れ図中のA点にきます。アームレイヤがImmediate Spacingにプログラムされていますから、オペレーションはスキャンレイヤ (A点) に下ります。

(B) スキャントリガコントロールがSourceに設定されていますから、A点ではトリガを待ちません。“Trigger Linkトリガを待つ”をバイパスし、B点に進みます。なお、バイパスが有効なのは最初にモデルを通過するときだけです。

(C) チャンネルトリガコントロールもSourceに設定されていますから、B点ではトリガを待ちません。“Trigger Linkトリガを待つ”をバイパスし、最初のチャンネルを閉じます (C点)。なお、バイパスが有効なのは最初にモデルを通過するときだけです。

(D) リレーのセットリング後、Model 7001はTrigger Linkトリガパルスを出力します (D点)。Model 7001は10チャンネルをスキャンするようにプログラムされていますから、オペレーションはB点までループバックし、ここで入力トリガを待ちます。なお、バイパスはもはや有効ではなくなっています。

(E) Model 7001からのトリガパルスがModel 196をトリガし、DUT #1の測定を行なわせます。測定終了後、DMMはトリガパルスを出力します (E点)。

DMMからModel 7001へトリガが加えられると、次にスキャンされるチャンネルが閉じます。そうすると、メータがトリガされて次のDUTを測定します。全10チャンネルがスキャンされ、測定されるまで、このプロセスが続きます。

(F) 最後のチャンネルがスキャンされ、測定された後、オペレーションはF点に進み、ここでModel 7001がトリガパルスを出力します。Model 7001はスキャンを2回行なうようにプログラムされていますから、オペレーションはA点までループバックし、ここで入力トリガを待ちます。なお、バイパスはもはや有効ではなくなっています。

(G) Model 7001からのトリガパルスがModel 230をトリガし、プログラムされている次の電圧レベルを出力させます。電圧レベルが設定されると、Model 230はトリガパルスを出力します (G点)。

Model 230からModel 7001へトリガパルスが加えられると、オペレーションはB点にきます。ここから新たなスキャンが開始されますから、バイパスは再び有効になります。したがって、オペレーションはC点に下り、ここで最初のチャンネルが再び閉じ、最終的には測定されます。前に説明したように、全10チャンネルがスキャンされ、測定されます。

2回目の最後のチャンネルが閉じ、測定された後、Model 7001はアイドル状態に戻ります。

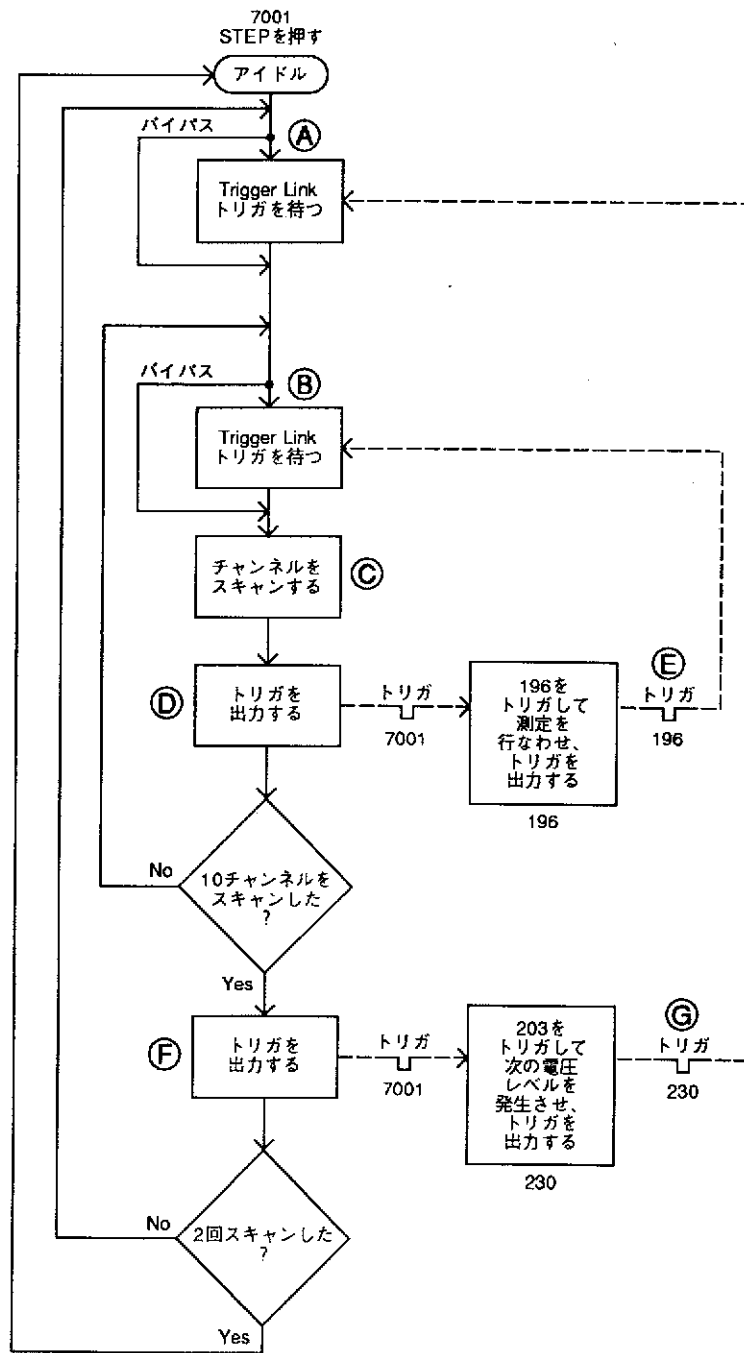


図4-23 非同期Trigger Link例 #2のオペレーションモデル

4.11.2 半同期オペレーション

半同期Trigger Linkモードでは、テストシステムにおける全トリガ（入力および出力）は、シングルラインによってコントロールされます。通常はHigh (+5V) のトリガラインがLow (0V) にされると、トリガは負のエッジで発生します。トリガラインがリリースされると、トリガは正のエッジで発生します（図4-24参照）。このシングルライントリガは、システム内の機器の1つがラインをLowに保持している限り、トリガが抑制されるという利点があります。言い換えれば、システム内の全機器がレディ状態になるまでトリガは発生しないということです。

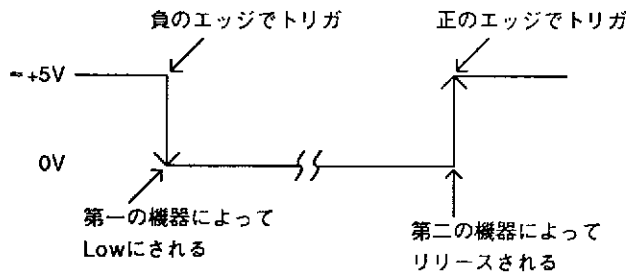


図4-24
半同期Trigger Link仕様

たとえば、図4-25に示すように、メータ（Trigger Link装置）が2台のModel 7001に接続され、半同期オペレーションが行なわれるものとしましょう。3台ともトリガライン#3を使うようにプログラムされています。2台のModel 7001のリレーセットリング時間は、それぞれ、10msecと50msecです。

最初にメータが測定を行なうものとします。読取りが行なわれた後、メータはトリガラインをLowにドライブします。

負のエッジで両方のModel 7001がトリガされ、チャンネルが閉じます。それぞれのModel 7001は、チャンネルを閉じているとき、トリガラインをLowに保持します。スイッチが閉じてから10msec後、第一のModel 7001がトリガラインをリリースします。しかし、第二のModel 7001はまだですからラインをLowに保持し続けます。スイッチが閉じてから50msec後、第二のModel 7001がトリガラインをリリースします。正のエッジでメータがトリガされ、測定が行なわれます。そして、トリガラインがLowに戻り、次のチャンネルが閉じます。全チャンネルがスキャンされ、測定されるまで、このプロセスが繰り返されます。

半同期Trigger Link例

この例でも、非同期Trigger Link例 #1で使ったのと同じテストシステム（図4-17）を使います。たし、トリガは半同期モードで行なわれます。Trigger Linkの接続を図4-26に示します。

Model 7001とメータは次のように設定されます。

Model 7001 :

スキャンリスト = 111-1!10

アームレイヤ :

アームスペーシング = Immediate*

アームカウント = 1*

アームトリガコントロール = Acceptor*

スキャンレイヤ :

スキャンスペーシング = Immediate*

スキャン回数 = 1

スキャントリガコントロール = Acceptor*

チャンネルレイヤ :

チャンネルスペーシング = TrigLink

Trigger Linkモード = Semi-Synchronous

半同期ライン = #1

トリガコントロール = Source*

チャンネル数 = Use Scanlist Length*

* 該設定がRESET（および出荷時）デフォルトであることを表わす。

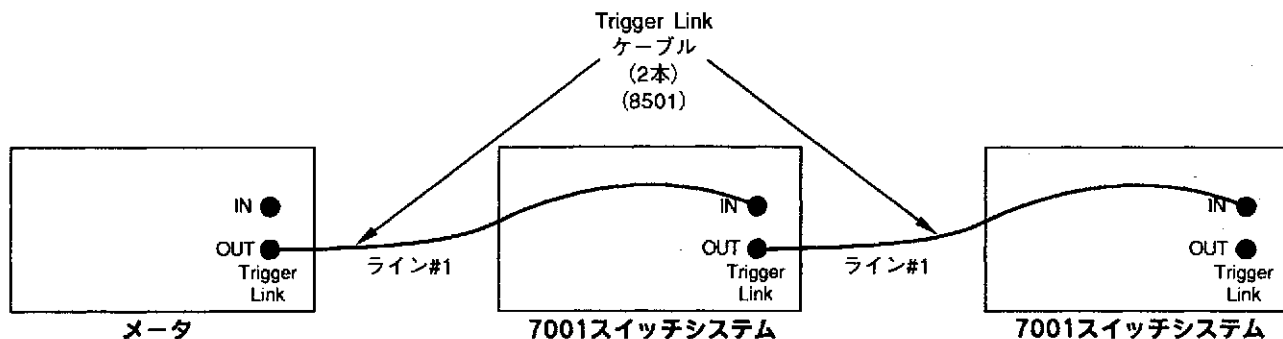


図4-25
標準的な半同期モード接続

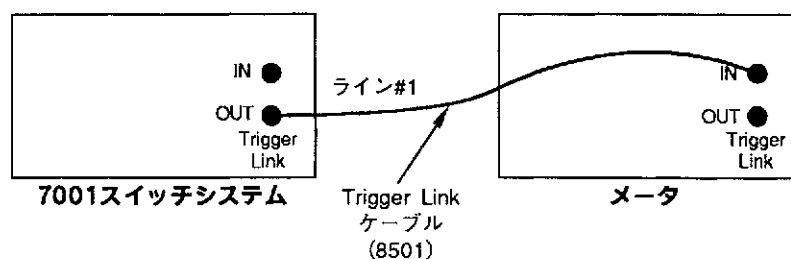


図4-26
Trigger Link接続 (半同期例)

メータ:

アームレイヤ:
 アームソース = Immediate
 アームカウント = 1
 アームトリガコントロール = Acceptor

スキャンレイヤ:
 スキャンソース = Immediate
 スキャン カウント = 1
 スキャントリガコントロール = Acceptor

メジャーレイヤ:
 メジャーソース = TrigLink
 Trigger Linkモード = Semi-Synchronous
 半同期ライン = #1
 トリガコントロール = Acceptor
 メジャーカウント = 10

テストを実行するには、メータ上のTRIGキーを押してメータをアイドル状態から抜け出させ、Model 7001上のSTEPキーを押します。以下に、図4-27に示すオペレーションモデルを参照しながら、オペレーションについて説明します。

Ⓐ TRIGキーを押すと、メータがアイドル状態から抜け出し、メータオペレーションが流れ図中のA点にきます。ここでTrigger Linkトリガを待ちます。アームレイヤとスキャンレイヤがどちらもImmediate Sourceにプログラムされていますから、オペレーションは直ちにメジャーレイヤ (A点) に下ります。

Ⓑ STEPキーを押すと、Model 7001がアイドル状態から抜け出し、オペレーションが流れ図中のB点にきます。アームレイヤとスキャンレイヤがどちらもImmediate Spacingにプログラムされていますから、オペレーションは直ちにチャンネルレイヤ (B点) に下ります。

③ チャンネルトリガコントロールがSourceに設定されていますから、B点ではトリガを待ちません。“Trigger Link トリガを待つ”をバイパスし、最初のチャンネルを閉じます (C点)。なお、バイパスが有効なのは最初にモデルを通過するときだけです。

④ リレーのセットリング後、Model 7001はTrigger LinkトリガラインをLowにします (D点)。Model 7001は10チャンネルをスキャンするようにプログラムされていますから、オペレーションはB点までループバックし、ここで入力トリガを待ちます。なお、バイパスはもはや有効ではありません。

⑤および⑥ メータオペレーションがA点でトリガを待っていることを思い出してください。Model 7001によってトリガラインがLowにされると、負のリーディングエッジでメータがトリガされ、DUT #1が測定されます。なお、メータはトリガラインをLowに保持しています。測定終了後、メータはトリガラインをリリースし (F点)、A点までループバックし、ここで別の入力トリガを待ちます。

メータがトリガラインをリリースすると、正のリーディングエッジでModel 7001がトリガされ、次にスキャンされるチャンネルが閉じます。そうすると、トリガラインがLowになり、メータがトリガされて次のDUTを測定します。全10チャンネルがスキャンされ、測定されるまで、このプロセスが続きます。

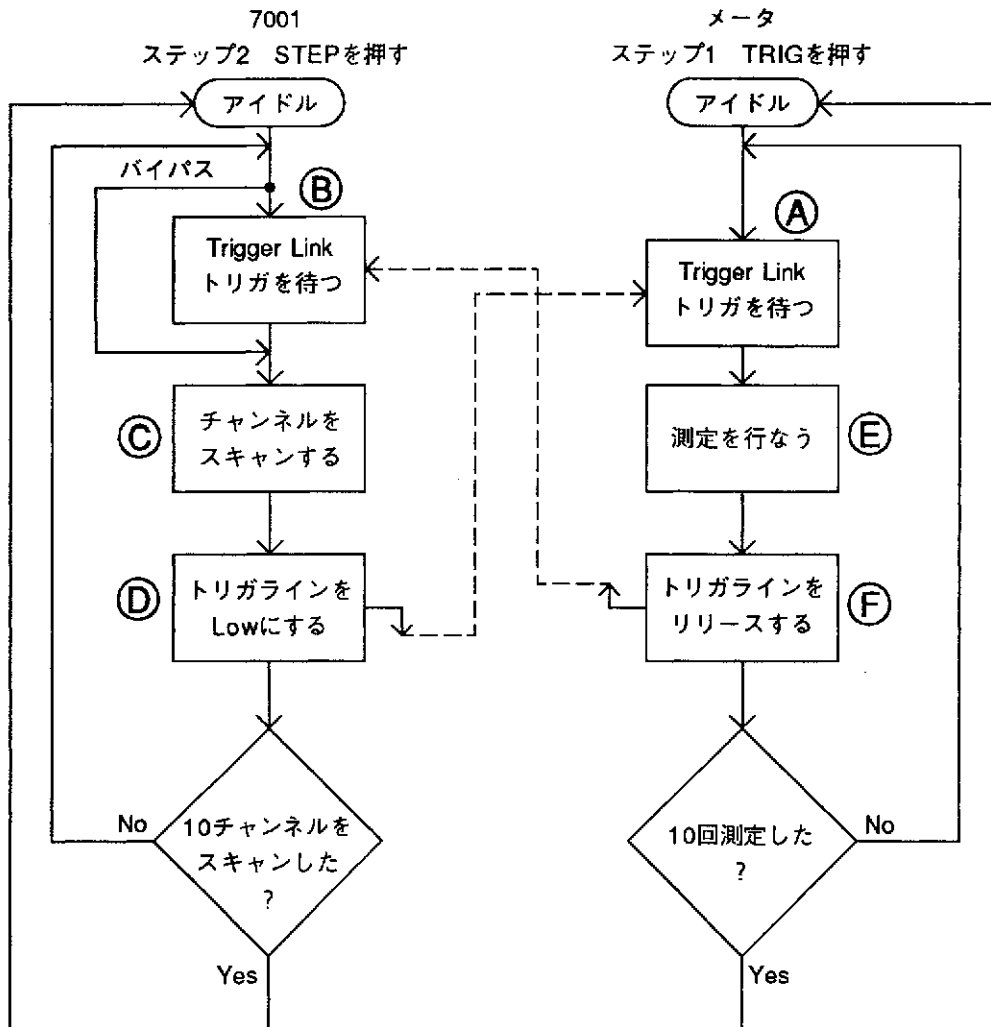


図4-27 半同期Trigger Link例のオペレーションモデル

第5部

IEEE-488について

5.1 概要

本節では、IEEE-488バスを通じてModel 7001をプログラムする場合に参考となる情報について記載します。本節の構成は次の通りです。

5.2 IEEE-488バスの接続：

IEEE-488バスへの機器接続について説明します。

5.3 1次アドレスの選択：

フロントパネルからの1次アドレスの設定の仕方について説明します。

5.4 コントローラプログラミング：

HP BASIC 4.0を用いたプログラミングステートメントを示します。

5.5 IEEE-488オペレーションのフロントパネル関連事項：

ステータスインジケータ、バス操作関係のメッセージ、およびLOCALキーの使い方について説明します。

5.6 ステータス構造：

Model 7001のステータス構造を構成している各種のレジスタおよび待ち行列について説明します。

5.7 トリガモデル：

IEEE-488バスを通じてのスキャンプロセス（トリガモデル）について説明します。

5.8 汎用バスコマンド：

Model 7001をリモート状態にするなどの基本操作に必要な汎用IEEE-488コマンドについて説明します。

5.9 コモンコマンド：

Model 7001が使うIEEE-488.2コモンコマンドについて説明します。

5.10 SCPIコマンドサブシステム：

Model 7001が使うSCPIコマンドについて説明します。

IEEE-488は計装データバスで、そのハードウェアおよびプログラミング規格はもともとは1975年にIEEE（米国電気電子学会）によって採択され、IEEE-488と名付けられたものです。これらの規格は、1978年にIEEE-488-1978に改訂され、さらに、1987年にIEEE-488.1-1987に改訂されました。Model 7001はこれらの規格に準拠しています。

Model 7001は、IEEE-488.2-1987規格およびSCPI 1991

(Standard Commands for Programmable Instruments) 規格にも準拠しています。IEEE-488.2では、機器へあるいは機器からデータを送信するためのシンタックスと、機器がこのデータをどのように解釈するかという点と、機器の状態を記録するためにどのようなレジスタが必要かという点について定めています。この規格のコマンド（コモンコマンド）のサブセットは、ステータスレジスタの読取り、測定トリガ、装置のリセットなどの基本操作を行なうのに使われます。

SCPI規格ではコマンド言語プロトコルについて定めています。SCPI規格はIEEE-488.2より一歩進んだもので、機器のあらゆるプログラマブルアスペクトをコントロールするためのコマンドの標準セットが定義されています。また、これらのコマンドのパラメータがとるべきフォーマットと、機器によって返送される値のフォーマットも定義されています。

5.2 IEEE-488バスの接続

Model 7001は、標準IEEE-488コネクタ（図5-1に例示）を備えたケーブルを介してIEEE-488バスに接続できます。このコネクタは、1台の機器に対して多数の並列接続を行なえるようにスタックすることができます。各コネクタには2個のねじがあって、確実な接続が可能になっています。現行の規格ではメートルねじを使うよう定められており、薄黒い色でメートルねじと判別されます。以前のバージョンでは違ったねじが用いられ、銀色のものでした。Model 7001はメートルねじを使うように設計されていますから、このタイプのコネクタは使用しないでください。

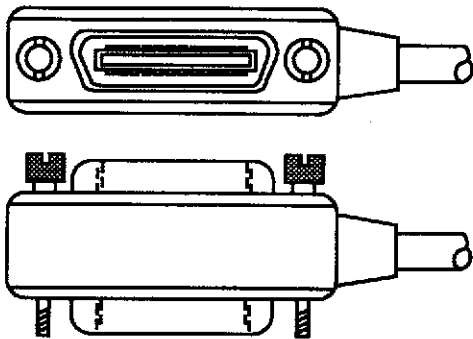


図5-1
IEEE-488コネクタ

マルチユニットテストシステムの場合の標準的な接続を図5-2に示します。コネクタは、理論上は、1台の機器に対して何個でもスタックできますが、機械的損傷を防止するため、1台の機器に対してスタックするコネクタは3個以下としてください。

注

電磁放射による干渉を最小限に抑えるため、必ず、シールドされたIEEE-488ケーブルを使用してください。Model 7007-1および7007-2用のIEEE-488シールドケーブルは東陽テクニカにお求めください。

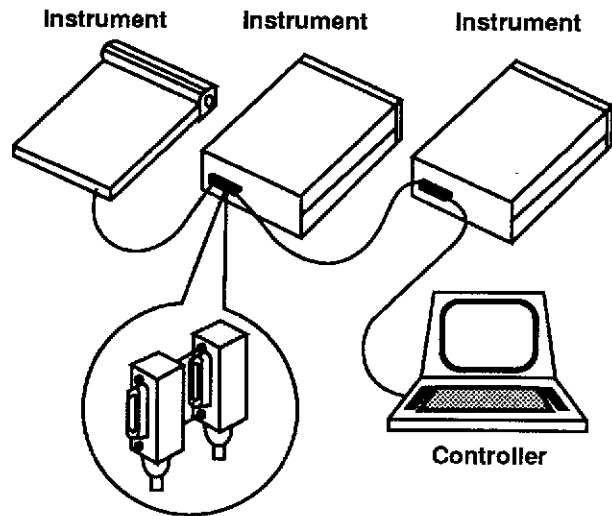


図5-2
IEEE-488の接続

Model 7001をIEEE-488に接続する手順は以下の通りです。

1. ケーブルコネクタをリヤパネル上のコネクタと整合します。コネクタは、決まった向きでしかはまらないようになっています。Model 7001上のIEEE-488コネクタの位置を図5-3に示します。
2. ねじをしっかりと締めます。ただし、締めすぎないように注意してください。
3. 必要あれば、他の機器からの別のコネクタを付加します。
4. ケーブルの他端がコントローラに正しく接続されていることを確認します。ほとんどのコントローラはIEEE-488型のコネクタを装備していますが、異なったタイプの接続ケーブルを必要とするものもいくつかあります。正しい接続の仕方についてはコントローラの取扱説明書を参照してください。

注

IEEE-488バスは、コントローラも含めて最大15台までの装置にしか使えません。最大ケーブル長は、20mあるいは装置数×2mのうちのどちらか短い方とします。これらの制限事項を守らないと、バスの動作不良が起こることがあります。

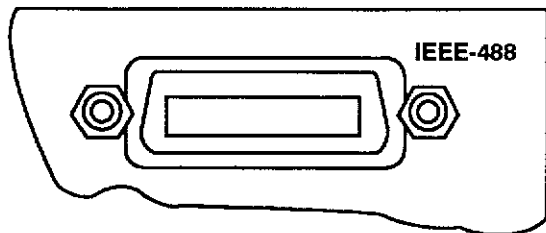


図5-3
IEEE-488コネクタの位置

表5-1および図5-4を参考としてカスタムケーブルを作ることができます。表5-1にはバスの接点アサインメント、図5-4には接点コンフィギュレーションをそれぞれ示してあります。

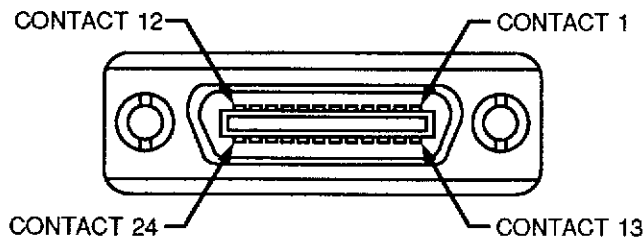


図5-4
接点アサインメント

注意

IEEE-488コモンはデジタルコモンに接続されます。デジタルコモンとアースグラウンドの間の最大電圧は0Vです。

表5-1
IEEE接点アサインメント

接点番号	IEEE-488指定	タイプ
1	DIO1	Data
2	DIO2	Data
3	DIO3	Data
4	DIO4	Data
5	EOI (24)*	Management
6	DAV	Handshake
7	NRFD	Handshake
8	NDAC	Handshake
9	IFC	Management
10	SRQ	Management
11	ATN	Management
12	SHIELD	Ground
13	DIO5	Data
14	DIO6	Data
15	DIO7	Data
16	DIO8	Data
17	REN (24)*	Management
18	Gnd, (6)*	Ground
19	Gnd, (7)*	Ground
20	Gnd, (8)*	Ground
21	Gnd, (9)*	Ground
22	Gnd, (10)*	Ground
23	Gnd, (11)*	Ground
24	Gnd, LOGIC	Ground

* 括弧内の番号は、それぞれ該当する接点番号のシグナルグラウンドリターンを表わす。接点24ではEOIおよびRENシグナルラインが戻る。

5.3 一次アドレスの選択

Model 7001は、リスンコマンドを受信してからでなければ、アドレッシングされたコマンドに回答しません。また、トークコマンドを受信してからでなければ、データを送信しません。Model 7001は、一次アドレスを7に設定して出荷されます。

一次アドレスは、他の機器との矛盾が避けられる限り、0から30までの任意の値に設定できます。なお、コントローラにも一次アドレスが与えられますから、そのアドレスは使わないでください。ほとんどの場合、コントローラのアドレスは0または21ですが、詳しくはコントローラの取扱説明書を参照してください。一次アドレスとしてどの値を選ぶにしても、コントローラのプログラム言語の一部として指定された値と一致することを確認してください。

現在の一次アドレスをチェックするか、または新たな値に変更するには、以下の手順を実行します。

1. MENUキーを押してMAIN MENUを表示させます。
2. カーソルキー（◀および▶）を使ってカーソルをGPIB上に位置付け、ENTERキーを押します。GPIB SETUP MENUが表示されます。
3. カーソルキーを使ってカーソルをADDRESS上に位置付け、ENTERキーを押します。Model 7001の現在の一次アドレスが表示されます。たとえば、Model 7001の一次アドレスが7に設定されてる場合、下記のメッセージが表示されます。

ADDRESS=07 (0-30)

4. 表示されているアドレスのままにしておくのであれば、EXITキーを3回押します。Model 7001がチャンネルステータス表示状態に戻ります。
5. 一次アドレスを変更するには、キーボードを使って新たなアドレス値（0～30）をキーインし、ENTERキーを押します。EXITキーを2回押してチャンネルステータス表示状態に戻ります。

注

バス上の各装置は、それぞれ固有の一次アドレスを持っていない限りなりません。この順守事項を守らないと、バスの動作不良が起こると考えられます。

5.4 コントローラプログラミング

本項に示すプログラム命令は、ヒューレットパッカード BASICバージョン4.0で書かれた例によるものです。この言語を選んだのは、IEEE-488バスのコントロールに対して汎用性があるためです。本項では、Model 7001の操作に不可欠のステートメントを示します。

注

Capital Equipment Corporation (CEC)、I/O TecおよびNational IEEE-488インタフェースを介してIBM PCインタフェースを使用するためのコントローラプログラミングについては、「付録」に記載してあります。

HP BASIC 4.0ステートメントの部分リストを表5.2に示します。これらのステートメントは、指定しなければならない1桁または3桁の引数を持っています。1桁目はインタフェース選択コードで、工場で7に設定されます。3桁の引数を必要とするステートメントの最後の2桁は一次アドレスを指定します。

3桁の引数を持つステートメントは、7の一次アドレス（工場設定の一次アドレス）を示します。当然のことですが、別のアドレスにするには、最後の2桁を必要な値に変更します。たとえば、17の一次アドレスを使って装置へGTLコマンドを送信するためには、LOCAL 717というステートメントを使います。

ステートメントの中には2つの形式を持つものがあり、厳密なコンフィギュレーションはバスを通じて送信されるコマンドに依存します。たとえば、CLEAR 7はDCLコマンドを送信し、CLEAR 716は16の一次アドレスを持つ装置へSDCコマンドを送信します。

表5-2

HP BASIC 4.0によるIEEE-488ステートメント

アクション	BASIC ステートメント
装置07へリングを送信する。	OUTPUT 707;AS
装置07からストリングを受け取る。	ENTER 707;AS
装置07へGTLを送信する。	LOCAL 707
装置07へSDCを送信する。	CLEAR 707
全装置へDCLを送信する。	CLEAR 7
リモートイネーブルを送信する。	REMOTE 7
リモートイネーブルをキャンセルする。	LOCAL 7
装置07をシリアルポールする。	SROLL (707)
ローカルロックアウトを送信する。	LOCAL LOCKOUT 7
装置07へGETを送信する。	TRIGGER 707
IFCを送信する。	ABORT 7

5.5 IEEE-488オペレーションの フロントパネル関連事項

以下に、メッセージ、ステータスインジケータ、LOCALキーなど、IEEE-488オペレーションのフロントパネル関連事項について説明します。

5.5.1 フロントパネルエラーメッセージ

IEEE-488プログラミングに関連したフロントパネルエラーメッセージを表5-3に示します。なお、Model 7001は、SRQを発生するようにプログラムすることができます。また、特定のエラー状態をチェックするためにコマンド問合せを行なうことができます。

表5-3

エラーメッセージとステータスメッセージ

コード番号	内容
+522	"Slot 2 identification error" (EE)
+521	"Slot 1 identification error" (EE)
+510	"Saved state error" (EE)
+173	"Waiting in arm layer 2" (SE)
+172	"Waiting in arm layer 1" (SE)
+171	"Waiting in trigger Layer" (SE)
+161	"Program running" (SE)
+126	"Device calculating" (SE)
+125	"Device measuring" (SE)
+124	"Device sweeping" (SE)
+123	"Device ranging" (SE)
+122	"Device settling" (SE)
+121	"Device calibrating" (SE)
+101	"Operation Complete" (SE)
0	"No error" (SE)
-100	"Command Error" (EE)
-101	"Invalid Character" (EE)
-102	"Syntax Error" (EE)
-103	"Invalid Separator" (EE)
-104	"Data Type Error" (EE)
-105	"Get not allowed" (EE)
-108	"Parameter not allowed" (EE)
-109	"Missing Parameter" (EE)
-110	"Command Header Error" (EE)
-111	"Command Header Separator Error" (EE)
-112	"Program mnemonic too long" (EE)
-113	"Undefined header" (EE)
-114	"Header suffix out of range" (EE)
-120	"Numeric data error" (EE)
-121	"Invalid character in number" (EE)
-123	"Exponent too large" (EE)
-124	"Too many digits in number" (EE)
-128	"Numeric data not allowed" (EE)
-140	"Character data error" (EE)
-141	"Invalid character data" (EE)
-144	"Character data too long" (EE)
-148	"Character data not allowed" (EE)
-150	"String data error" (EE)

表5-3

エラーメッセージとステータスメッセージ

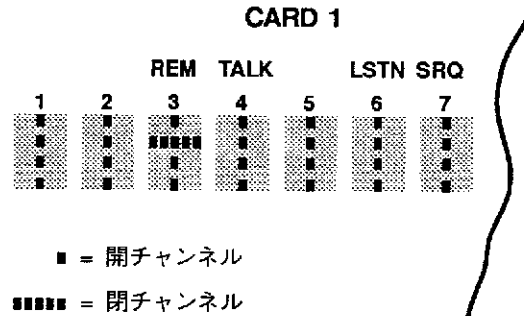
コード番号	内容
-151	"Invalid string data" (EE)
-154	"String too long"
-158	"String data not allowed" (EE)
-160	"Block data error" (EE)
-161	"Invalid block data" (EE)
-168	"Block data not allowed" (EE)
-170	"Expression error" (EE)
-171	"Invalid expression" (EE)
-178	"Expression data not allowed" (EE)
-200	"Execution error" (EE)
-201	"Invalid while in local" (EE)
-202	"Settings lost due to rti" (EE)
-210	"Trigger error" (EE)
-211	"Trigger ignored" (EE)
-212	"Arm ignored" (EE)
-213	"Init ignored" (EE)
-214	"Trigger deadlock" (EE)
-215	"Arm deadlock" (EE)
-220	"Parameter Error" (EE)
-221	"Settings conflict" (EE)
-222	"Parameter data out of range" (EE)
-223	"Too much data" (EE)
-224	"Illegal parameter value" (EE)
-241	"Hardware missing" (EE)
-260	"Expression Error" (EE)
-281	"Cannot create program" (EE)
-282	"Illegal program name" (EE)
-284	"Program currently running" (EE)
-285	"Program syntax error" (EE)
-330	"Self Test failed" (EE)
-350	"Queue overflow" (EE)
-410	"Query interrupted" (EE)
-420	"Query untermiated" (EE)
-430	"Query deadlocked" (EE)
-440	"Query untermiated after indefinite response" (EE)

EE = エラーイベント

SE = ステータスイベント

5.5.2 IEEE-488ステータスインジケータ

REM (リモート)、TALK (トーク)、LSTN (リスン) およびSRQ (サービス要求) アナラシエータは、Model 7001の現在のIEEE-488ステータスを示します。以下に、これらのインジケータについて簡単に説明します。



REM — 名前から分かるように、このインジケータは、Model 7001がリモート状態になると表示されます。なお、REMは必ずしもREMラインの状態を示すとは限りません。REMが真の状態でもModel 7001をリスンにアドレッシングしてからでなければ、REMインジケータは表示されないからです。Model 7001がリモート状態になると、LOCALキー以外の全てのフロントパネルキーはロックアウトされます。REMが消えると、Model 7001はローカル状態に戻り、フロントパネルオペレーションが回復します。

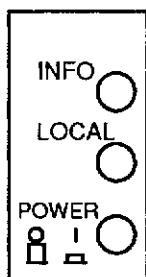
TALK — このインジケータは、Model 7001がトーカーアクティブ状態になると表示されます。Model 7001をこの状態にするには、正しいMTA (My Talk Address) コマンドを使ってModel 7001をトークにアドレッシングします。Model 7001がトーカーアイドル状態になると、TALKは消えます。Model 7001をトーカーアイドル状態にするには、Model 7001にUNT (Untalk) コマンドを送信するか、Model 7001をリスンにアドレッシングするか、またはIFC (Interface Clear) コマンドを使います。

LSTN — このインジケータは、Model 7001がリスナアクティブ状態になると表示されます。Model 7001をこの状態にするには、正しいMLA (My Listen Address) を使ってModel 7001をリスンにアドレッシングします。Model 7001がリスナアイドル状態になると、LSTNは消えます。Model 7001をリスナアイドル状態にするには、UNL (Unlisten) を送信するか、トークにアドレッシングするか、またはバスを通じてIFC (Interface Clear) を送信します。

SRQ — Model 7001は、1つまたはそれ以上のエラーあるいは条件が生じたときにサービス要求 (SRQ) を発生するようにプログラムすることができます。このインジケータが表示されたら、サービス要求が発生したということです。このインジケータは、シリアルポルバイトが読み取られるまで消えません。詳細は5.6.8項を参照してください。

5.5.3 LOCALキー

LOCALキーは、Model 7001のリモート状態をキャンセルし、ローカルオペレーションを回復するものです。



Model 7001がリモート状態になると、LOCALキー以外の全てのフロントパネルキーはロックアウトされますから、このLOCALキーはフロントパネルオペレーションを回復するのに便利です。LOCALキーを押すと、REMインジケータが消え、また、ユーザ定義メッセージが表示されていた場合にはディスプレイが通常の状態に戻ります。

なお、LOCALキーは、LLO (ローカルロックアウト) コマンドが有効になっているときは使用不能です。

5.6 ステータス構造

Model 7001のステータスレジスタ構造を図5-5に示します。以下に、このステータス構造を構成している各種のレジスタと待ち行列について説明します。

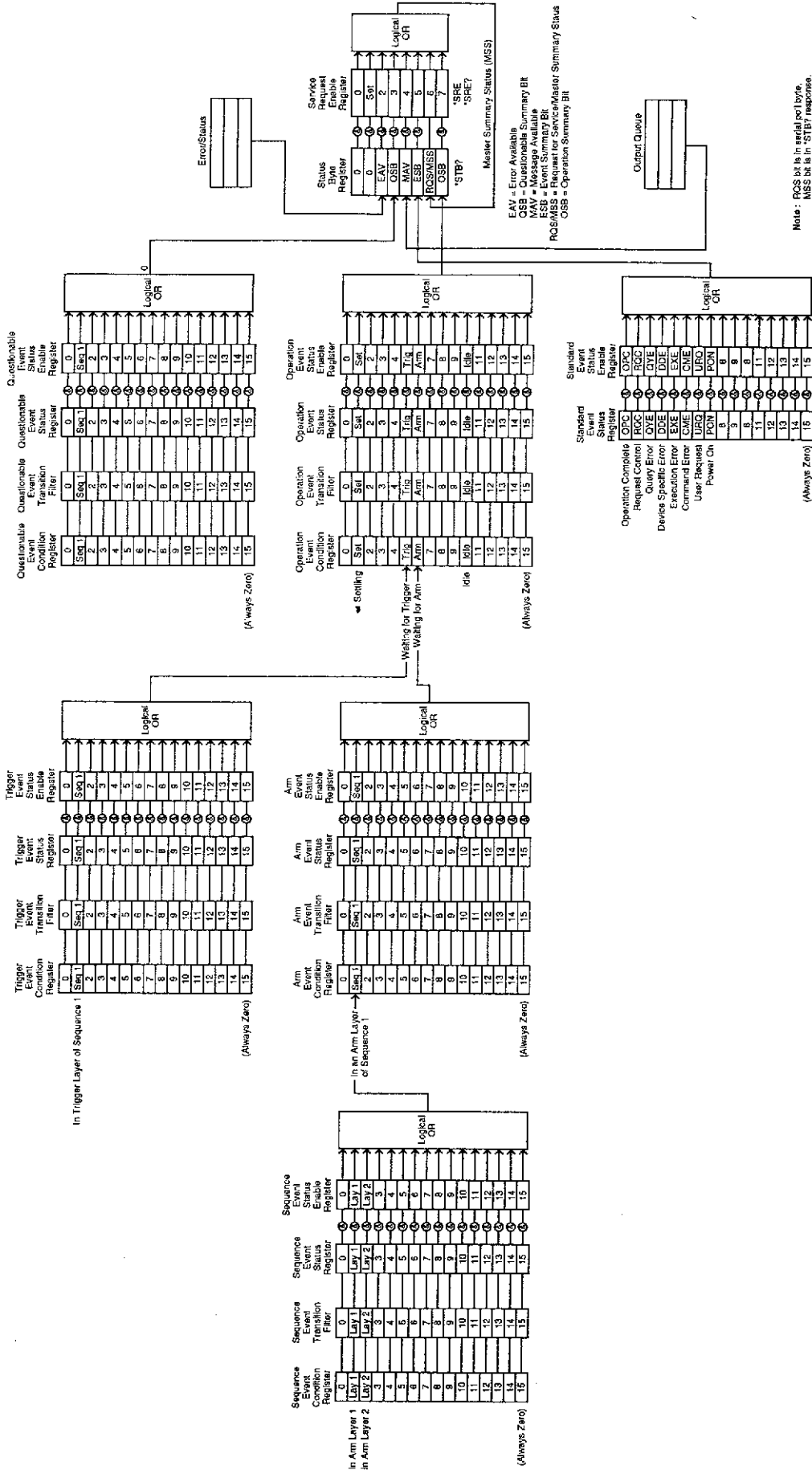


図 5-5 Model 7001 ステータス構造

5.6.1 標準イベントステータス

標準イベントの報告は、標準イベントステータスレジスタと標準イベントステータスイネーブルレジスタの2つの16ビットレジスタによってコントロールされます。これらのレジスタの構造を図5-6に示します。なお、ビット8～14は使用されませんから、図中には示してありません。

原則として、標準イベントが発生すると、標準イベントステータスレジスタの該当するビットがセットされます。このレジスタは、標準イベントが発生したかどうかを調べるために、随時に読み取ることができます。また、標準イベントステータスイネーブルレジスタを正しく使えば、標準イベントの発生で、ステータスバイトレジスタのイベントサマリビット (ESB) をセットすることができます。したがって、プログラマはサービス要求 (SRQ) 機能を利用することができます。詳細は5.6.8項を参照してください。

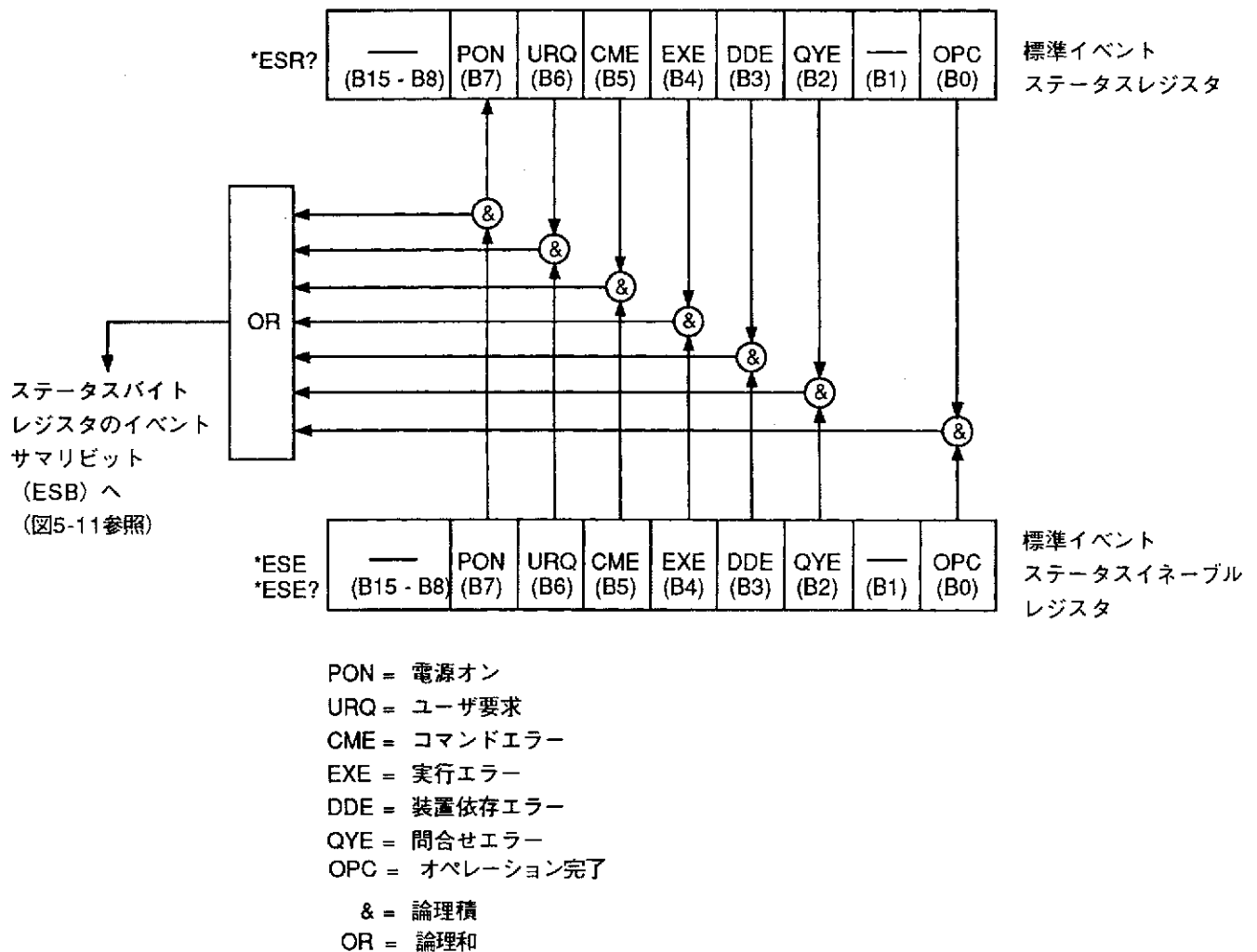


図5-6
標準イベントステータス

標準イベントステータスレジスタ — これはラッチされたレジスタで、標準イベントの発生を記録するのに使います。このレジスタの各ビット (Model 7001では使われないビットB1を除く) は、標準イベントを表わします。これらの標準イベントについては5.9.4項に説明してあります。

標準イベントが発生すると、標準イベントステータスレジスタの該当するビットがセットされます。たとえば、空の出力待ち行列からデータを読み取ろうとすると、問合せエラー (QYE) が発生し、ステータスレジスタのビットB2がセットされます。このレジスタのセットされたビットは、該当するオペレーションが実行されてレジスタがクリアされるまで、セットされたままです。

標準イベントステータスレジスタは、下記のコモン問合せコマンドを使って、随時に読み取ることができます (詳細は5.9.4.項を参照してください)。

*ESR?

*ESR?コマンドを使ってこのレジスタを読み取ると、レジスタはクリアされます。標準イベントステータスレジスタをクリアするオペレーションを示すと、次の通りです。

1. パワーサイクリング
2. *CLSコモンコマンドの送信
3. *ESR?コモンコマンドの送信

標準イベントステータスイネーブルレジスタ — このレジスタはユーザによってプログラムされ、標準イベントをマスクする働きをします。標準イベントがマスクされると、そのイベントが発生しても、ステータスバイトレジスタのイベントサマリビット (ESB) はセットされません。逆に、標準イベントがアンマスクされたときは、そのイベントが発生すると、ESBビットがセットされます。

標準イベントステータスレジスタのビットは、標準イベントステータスイネーブルレジスタの対応するビットがクリア (0) されると、マスクされます。標準イベントステータスレジスタのマスクされたビットがセットされると、標準イベントステータスイネーブルレジスタの対応するクリアされたビットとのANDがとられます。ANDゲートの論理“0”出力がORゲートの入力に加えられますから、ステータスバイトレジスタのESBビットはセットされません。

標準イベントステータスレジスタのビットは、標準イベントステータスイネーブルレジスタの対応するビットがセット (1) されると、アンマスクされます。標準イベントステータスレジスタのアンマスクされたビットがセットされると、標準イベントステータスイネーブルレジスタの対応するセットされたビットとのANDがとられます。ANDゲートの論理“1”出力がORゲートの入力に加えられますから、ステータスバイトレジスタのESBビットがセットされます。

標準イベントステータスイネーブルレジスタの個々のビットは、下記のコモンコマンドを使って、セットまたはクリアすることができます (詳細は5.9.2項を参照してください)。

*ESE<n>

標準イベントステータスイネーブルレジスタは、下記のコモン問合せコマンドを使って、随時に読み取ることができます (詳細は5.9.3項を参照してください)。

*ESE?

*ESE?コマンドを使ってこのレジスタを読み取っても、レジスタはクリアされません。標準イベントステータスイネーブルレジスタをクリアするオペレーションを示すと、次の通りです。

1. パワーサイクリング
2. *ESE 0

5.6.2 オペレーションイベントステータス

オペレーションイベントの報告は一連の16ビットレジスタによってコントロールされます。すなわち、オペレーションイベント条件レジスタ、トランジションフィルタ、オペレーションイベントステータスレジスタおよびオペレーションイベントステータスイネーブルレジスタによってコントロールされます。これらのレジスタの構造を図5-7に示します。

図5-5を見てください。オペレーションイベント条件レジスタのビットB5 (トリガ待ち) およびB6 (アーム待ち) は、アームイベントレジスタセットおよびトリガイベントレジスタセットによってコントロールされます (詳細は5.6.3項および5.6.5項を参照してください)。

これらのレジスタに使われている各ビットは、オペレーションイベントを表わします。オペレーションイベントビットについては5.10.6項で説明します。

オペレーションステータスレジスタは、:STATusサブシステムの:STATus:OPERationコマンド (5.10.6項参照) によってコントロールされます。

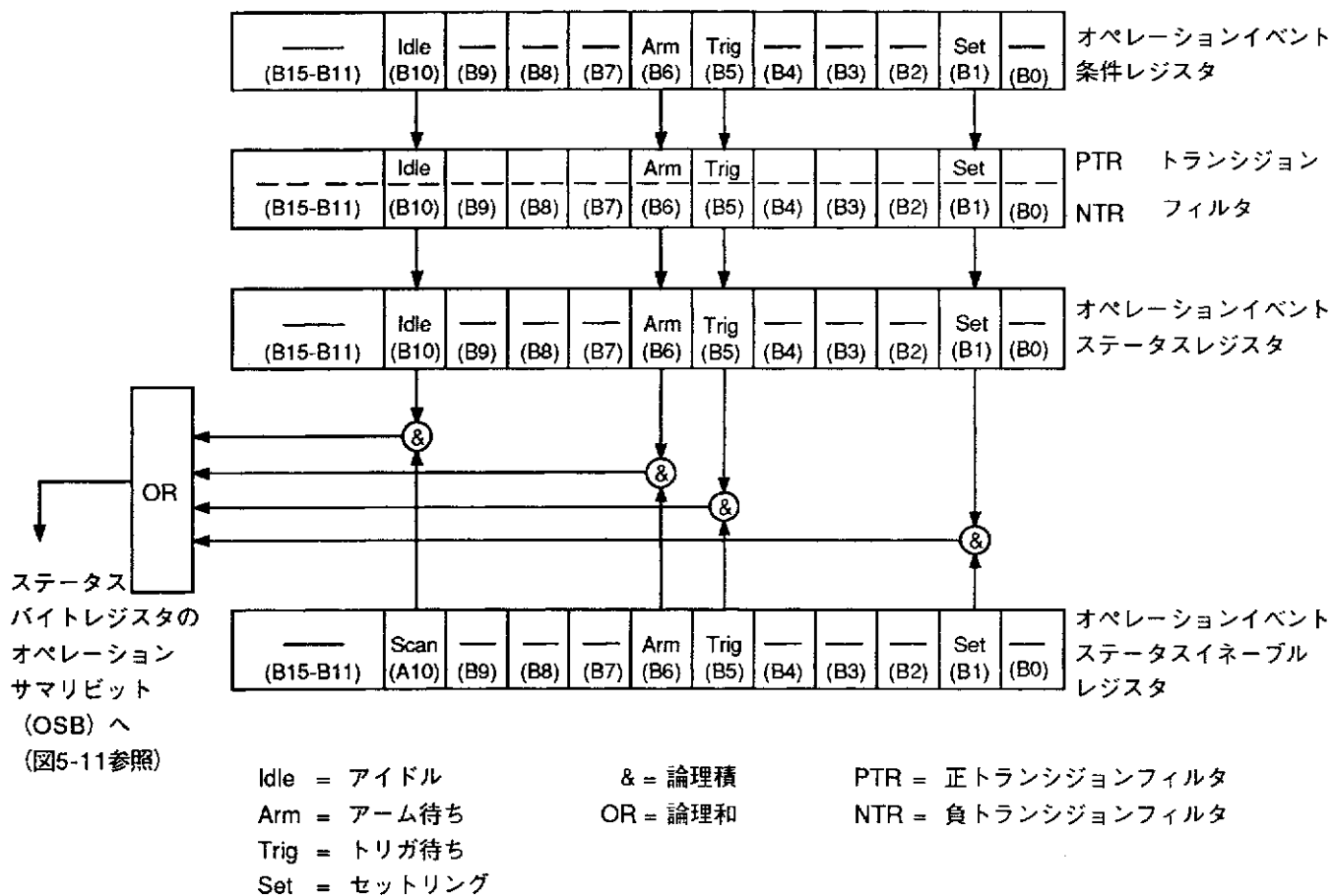


図5-7

オペレーションイベントステータス

オペレーションイベント条件レジスタ — これは16ビットの読取り専用レジスタで、Model 7001の現在の動作状態を反映するように絶えず更新されます。たとえば、リレーが閉じた後、このレジスタのビットB1 (Set) がセットされます (この場合、ステータスレジスタは正トランジションにプログラムされているものとします)。セトリング時間が経過すると、ビットB1はクリアされます。

オペレーションイベント条件レジスタは、下記のSCPI問合せコマンドを使って読み取ることができます。

```
:STATus:OPERation:CONDition?
```

オペレーションイベント条件レジスタとトランジションフィルタは、オペレーションイベントステータスレジスタのビットをセットするのに使われます。トランジションフィルタについては次に説明します。

オペレーションイベントトランジションフィルタ — このトランジションフィルタは、ユーザによってプログラムされる2つの16ビットレジスタで構成されています。このフィルタを使って、オペレーションイベント条件レジスタのどちらのトランジション (0から1または1から0) によってオペレーションイベントステータスレジスタの対応するビットをセットするかを指定します。

このフィルタは、正トランジション (PTR) または負トランジション (NTR) あるいはその両方にプログラムすることができます。イベントビットを正トランジションにプログラムした場合、オペレーションイベントステータスレジスタのイベントビットは、オペレーションイベント条件レジスタの対応するビットが0から1に変わるとセットされます。逆に、イベントビットを負トランジションにプログラムした場合、オペレーションイベントステータスレジスタのイベントビットは、オペレーションイベント条件レジスタの対応するビットが1から0に変わるとセットされます。

トランジションフィルタレジスタの個々のビットは、下記のSCPIコマンドを使ってセットまたはクリアすることができます。

```
:STATus:OPERation:PTR<n>
:STATus:OPERation:NTR<n>
```

トランジションフィルタレジスタは、下記のSCPI問合せコマンドを使って随時に読み取ることができます。

```
:STATus:OPERation:PTR?
:STATus:OPERation:NTR?
```

これらの問合せコマンドを使ってトランジションフィルタレジスタを読み取っても、レジスタの内容は影響を受けません。

下記のオペレーションでは、PTRレジスタの全ビットがセット (1) され、NTRレジスタの全ビットがリセット (0) されます。

1. パワーサイクリング
2. :STATus:PRESetコマンドの送信
3. :STATus:OPERation:PTR 65535コマンドおよび
:STATus:OPERation:NTR 0コマンドの送信

オペレーションイベントステータスレジスタ — これはラッチされた読取り専用レジスタで、そのビットはオペレーションイベント条件レジスタおよびトランジションフィルタによってセットされます。このレジスタのビットは、いったんセットされると、特定のクリアオペレーションによってレジスタがクリアされるまでセット (ラッチ) されたままです。このレジスタのビットは、オペレーションイベントステータスイネーブルレジスタのビットとのANDがとられ、ORゲートに加えられます。ORゲートの出力はオペレーションサマリビット (OSB) で、ステータスバイトレジスタに加えられます。オペレーションイベントステータスレジスタは、下記のSCPI問合せコマンドを使って読み取ることができます。

```
:STATus:OPERation?
```

このSCPIコマンドを使ってオペレーションイベントステータスレジスタを読み取ると、レジスタはクリアされます。このレジスタをクリアするオペレーションを示すと、次の通りです。

1. パワーサイクリング
2. *CLSコモンコマンドの送信
3. :STATus:OPERation?問合せコマンドの送信

オペレーションイベントステータスイネーブルレジスタ — このレジスタはユーザによってプログラムされ、オペレーションイベントステータスレジスタをマスクする働きをします。マスクされると、オペレーションイベントステータスレジスタのビットがセットされても、ステータスバイトレジスタのオペレーションサマリビット (OSB) はセットされません。逆に、アンマスクされたときは、オペレーションイベントステータスレジスタのビットがセットされると、OSBビットがセットされます。

オペレーションイベントステータスレジスタのビットは、オペレーションイベントステータスイネーブルレジスタの対応するビットがクリア (0) されると、マスクされます。オペレーションイベントステータスレジスタのマスクされたビットがセットされると、オペレーションイベントステータスイネーブルレジスタの対応するクリアされたビットとのANDがとられます。ANDゲートの論理“0”出力がORゲートの入力に加えられますから、ステータスバイトレジスタのOSBビットはセットされません。

オペレーションイベントステータスレジスタのビットは、オペレーションイベントステータスイネーブルレジスタの対応するビットがセット (1) されると、アンマスクされます。オペレーションイベントステータスレジスタのアンマスクされたビットがセットされると、オペレーションイベントステータスイネーブルレジスタの対応するセットされたビットとのANDがとられます。ANDゲートの論理“1”出力がORゲートの入力に加えられますから、ステータスバイトレジスタのOSBビットがセットされます。

オペレーションイベントステータスイネーブルレジスタの個々のビットは、下記のSCPIコマンドを使ってセットまたはクリアすることができます。

```
:STATus:OPERation:ENABle<n>
```

オペレーションイベントステータスレジスタは、下記のSCPI問合せコマンドを使って読み取ることができます。

```
:STATus:OPERation:ENABle?
```

このSCPIコマンドを使ってオペレーションイベントステータスレジスタを読み取っても、レジスタはクリアされません。このレジスタをクリアするオペレーションを示すと、次の通りです。

1. パワーサイクリング
2. :STATus:PRESetコマンドの送信
3. :STATus:OPERation:ENABle 0コマンドの送信

5.6.3 アームイベントステータス

アームイベントの報告は一連の16ビットレジスタによってコントロールされます。すなわち、アームイベント条件レジスタ、トランジションフィルタ、アームイベントステータスレジスタおよびアームイベントステータスイネーブルレジスタによってコントロールされます。これらのレジスタの構造を図5-8に示します。

図5-5を見てください。アームイベント条件レジスタのビットB1 (In An Arm Layer) は、シーケンスイベントレジスタセットによってコントロールされます (詳細は5.6.4項を参照してください)。原則として、ビットB1は、Model 7001がアームレイヤ (Arm Layer 1) またはスキャンレイヤ (Arm Layer 2) に入るとセットされます。パスを通じてのスキャンプロセスについては5.7項で説明します。

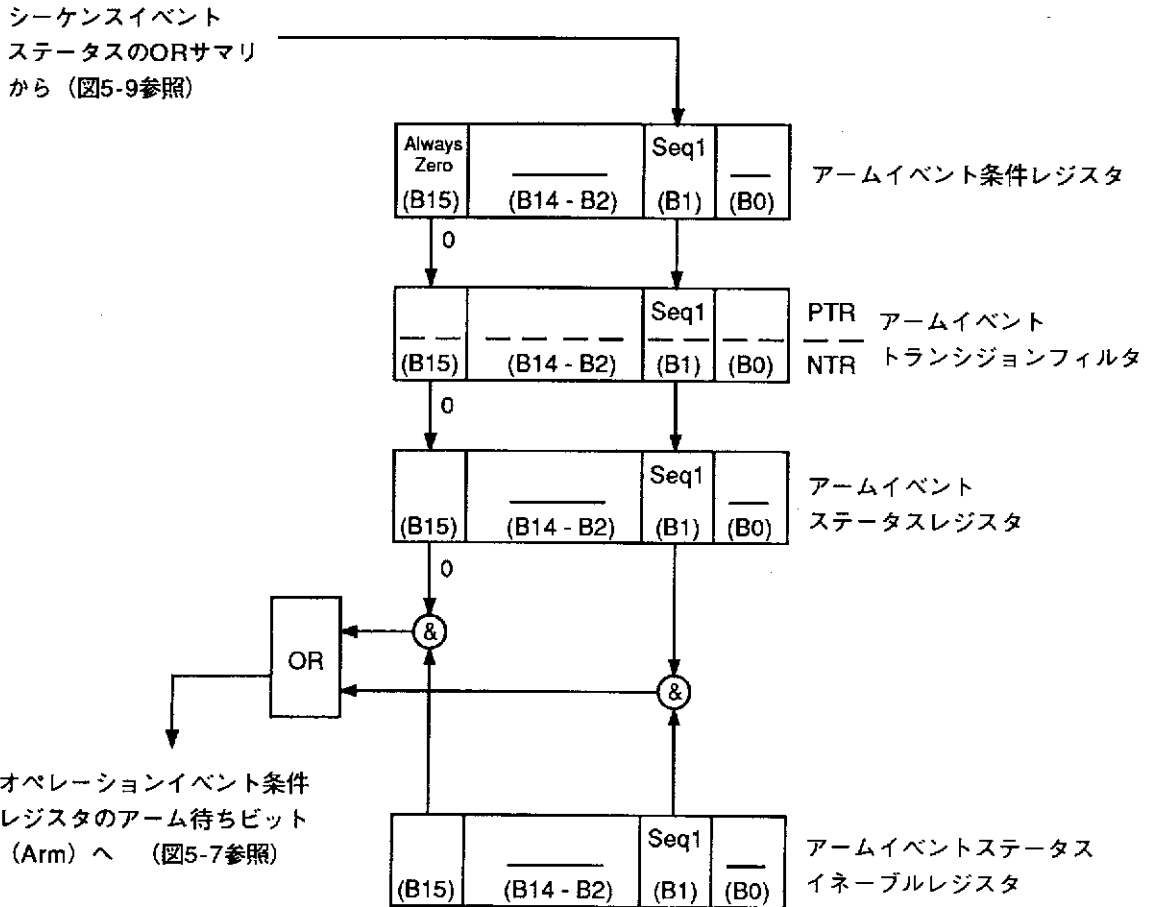
以下に、アームイベントステータス用の各種のレジスタについて説明します。なお、これらのレジスタは、:STATusサブシステムの:STATus:OPERation:ARMコマンド (5.10.6項参照) によってコントロールされます。

アームイベント条件レジスタ — これはリアルタイムの16ビット読取り専用レジスタで、シーケンスイベントレジスタセットのORサマリ [ORed summary] を反映するように絶えず更新されます。原則として、ビットB1がセットされているときは、Model 7001はアームレイヤに入っているか、またはアームレイヤから出たばかりです。

アームイベント条件レジスタは、下記のSCPI問合せコマンドを使って読み取ることができます。

```
:STATus:OPERation:ARM:CONDition?
```

アームイベント条件レジスタとトランジションフィルタは、アームイベントステータスレジスタのビットをセットするのに使われます。トランジションフィルタについては次に説明します。



Seq1 = シーケンス1 (セットされたビットは、7001がシーケンス1の
アームレイヤに入っていることを表わす)

- & = 論理積
- OR = 論理和
- PTR = 正トランジションレジスタ
- NTR = 負トランジションレジスタ

図5-8
アームイベントステータス

アームイベントトランジションフィルタ — このトランジションフィルタは、ユーザによってプログラムされる2つの16ビットレジスタで構成されています。このフィルタを使って、アームイベント条件レジスタのビットB1のどちらのトランジション (0から1または1から0) によってアームイベントステータスレジスタのビットB1をセットするかを指定します。

このフィルタは、正トランジション (PTR) または負トランジション (NTR) あるいはその両方にプログラムすることができます。イベントビットを正トランジションにプログラムした場合、アームイベントステータスレジスタのイベントビットは、アームイベント条件レジスタの対応するビットが0から1に変わるとセットされます。逆に、イベントビットを負トランジションにプログラムした場合、ステータスレジスタのイベントビットは、条件レジスタの対応するビットが1から0に変わるとセットされます。

トランジションフィルタレジスタは、下記のSCPIコマンドを使ってセットまたはクリアすることができます。

```
:STATus:OPERation:ARM:PTR<n>
:STATus:OPERation:ARM:NTR<n>
```

トランジションフィルタレジスタは、下記のSCPI問合せコマンドを使って随時に読み取ることができます。

```
:STATus:OPERation:ARM:PTR?
:STATus:OPERation:ARM:NTR?
```

これらの問合せコマンドを使ってトランジションフィルタレジスタを読み取っても、レジスタの内容は影響を受けません。

下記の実行では、PTRレジスタの全ビットがセット (1) され、NTRレジスタの全ビットがリセット (0) されます。

1. パワーサイクリング
2. :STATus:PRESetコマンドの送信
3. :STATus:OPERation:ARM:PTR 65535コマンドおよび
:STATus:OPERation:ARM:NTR 0コマンドの送信

アームイベントステータスレジスタ — これはラッチされた読み取り専用レジスタで、そのビットはアームイベント条件レジスタおよびトランジションフィルタによってセットされます。このレジスタのビットは、いったんセットされると、特定のクリアオペレーションによってレジスタがクリアされるまでセット (ラッチ) されたままです。このレジスタのビットは、アームイベントステータスイネーブルレジスタのビットとのANDがとられ、ORゲートに加えられます。

ORゲートの出力はアームサマリビット (OSB) で、ステータスバイトレジスタに加えられます。アームイベントステータスレジスタは、下記のSCPI問合せコマンドを使って読み取ることができます。

```
:STATus:OPERation:ARM?
```

このSCPIコマンドを使ってアームイベントステータスレジスタを読み取ると、レジスタはクリアされます。オペレーションイベントステータスレジスタをクリアするオペレーションを示すと、次の通りです。

1. パワーサイクリング
2. *CLSコモンコマンドの送信
3. :STATus:OPERation:ARM?問合せコマンドの送信

アームイベントステータスイネーブルレジスタ — このレジスタはユーザによってプログラムされ、アームイベントステータスレジスタをマスクする働きをします。マスクされると、アームイベントステータスレジスタのビット (B1) がセットされても、ステータスバイトレジスタのアームサマリビット (OSB) はセットされません。逆に、アンマスクされたときは、アームイベントステータスレジスタのビット (B1) がセットされると、OSBビットがセットされます。

アームイベントステータスレジスタのビットB1は、アームイベントステータスイネーブルレジスタの対応するビット (B1) がクリア (0) されると、マスクされます。アームイベントステータスレジスタのマスクされたビットがセットされると、アームイベントステータスイネーブルレジスタの対応するクリアされたビットとのANDがとられます。ANDゲートの論理 "0" 出力がORゲートの入力に加えられるから、ステータスバイトレジスタのOSBビットはセットされません。

アームイベントステータスレジスタのビットB1は、アームイベントステータスイネーブルレジスタの対応するビット (B1) がセット (1) されると、アンマスクされます。アームイベントステータスレジスタのアンマスクされたビットがセットされると、アームイベントステータスイネーブルレジスタの対応するセットされたビットとのANDがとられます。ANDゲートの論理 "1" 出力がORゲートの入力に加えられるから、ステータスバイトレジスタのOSBビットがセットされます。

アームイベントステータスイネーブルレジスタのビットB1は、下記のSCPIコマンドを使ってセットまたはクリアすることができます。

```
:STATus:OPERation:ARM:ENABLE<n>
```

アームイベントステータスレジスタは、下記のSCPI問合せコマンドを使って読み取ることができます。

```
:STATus:OPERation:ARM:ENABle?
```

このSCPIコマンドを使ってアームイベントステータスレジスタを読み取っても、レジスタはクリアされません。このレジスタをクリアするオペレーションを示すと、次の通りです。

1. パワーサイクリング
2. :STATus:PRESetコマンドの送信
3. :STATus:OPERation:ARM:ENABle 0コマンドの送信

5.6.4 シーケンスイベントステータス

シーケンスイベントの報告は一連の16ビットレジスタによってコントロールされます。すなわち、シーケンスイベント条件レジスタ、トランジションフィルタ、シーケンスイベントステータスレジスタおよびシーケンスイベントステータスイネーブルレジスタによってコントロールされます。これらのレジスタの構造を図5-9に示します。

このレジスタセットの2ビットをModel 7001がシーケンスイベントを報告するのに使います。Model 7001がアームレイヤに入ると（またはそこから出ると）、ビットB1 (In Arm Layer 1) がセットされます。Model 7001がスキャンレイヤに入ると（またはそこから出ると）、ビットB2 (In Arm Layer 2) がセットされます。バスを通じてのスキャンプロセスについては5.7項で説明します。

以下に、シーケンスイベントステータス用の各種のレジスタについて説明します。なお、これらのレジスタは、:STATusサブシステムの:STATus:OPERation:ARM:SEQuenceコマンド (5.10.6項参照) によってコントロールされます。

シーケンスイベント条件レジスタ — これはリアルタイムの16ビット読取り専用レジスタで、Model 7001の現在のアームレイヤステータスを反映するように絶えず更新されます。たとえば、現在、Model 7001がスキャンレイヤに入っていれば、このレジスタのビットB2 (In Arm Layer 2) がセットされています。

シーケンスイベント条件レジスタは、下記のSCPI問合せコマンドを使って読み取ることができます。

```
:STATus:OPERation:ARM:SEQuence:CONDition?
```

シーケンスイベント条件レジスタとトランジションフィルタは、シーケンスイベントステータスレジスタのビットをセットするのに使われます。トランジションフィルタについては次に説明します。

シーケンスイベントトランジションフィルタ — このトランジションフィルタは、ユーザによってプログラムされる2つの16ビットレジスタで構成されています。このフィルタを使って、シーケンスイベント条件レジスタのどちらのトランジション (0から1または1から0) によってシーケンスイベントステータスレジスタの対応するビットをセットするかを指します。

このフィルタは、正トランジション (PTR) または負トランジション (NTR) あるいはその両方にプログラムすることができます。イベントビットを正トランジションにプログラムした場合、シーケンスイベントステータスレジスタのイベントビットは、シーケンスイベント条件レジスタの対応するビットが0から1に変わるとセットされます。逆に、イベントビットを負トランジションにプログラムした場合、シーケンスイベントステータスレジスタのイベントビットは、シーケンスイベント条件レジスタの対応するビットが1から0に変わるとセットされます。

トランジションフィルタレジスタは、下記のSCPIコマンドを使ってセットまたはクリアすることができます。

```
:STATus:OPERation:ARM:SEQuence:PTRansition<n>
:STATus:OPERation:ARM:SEQuence:NTRansition<n>
```

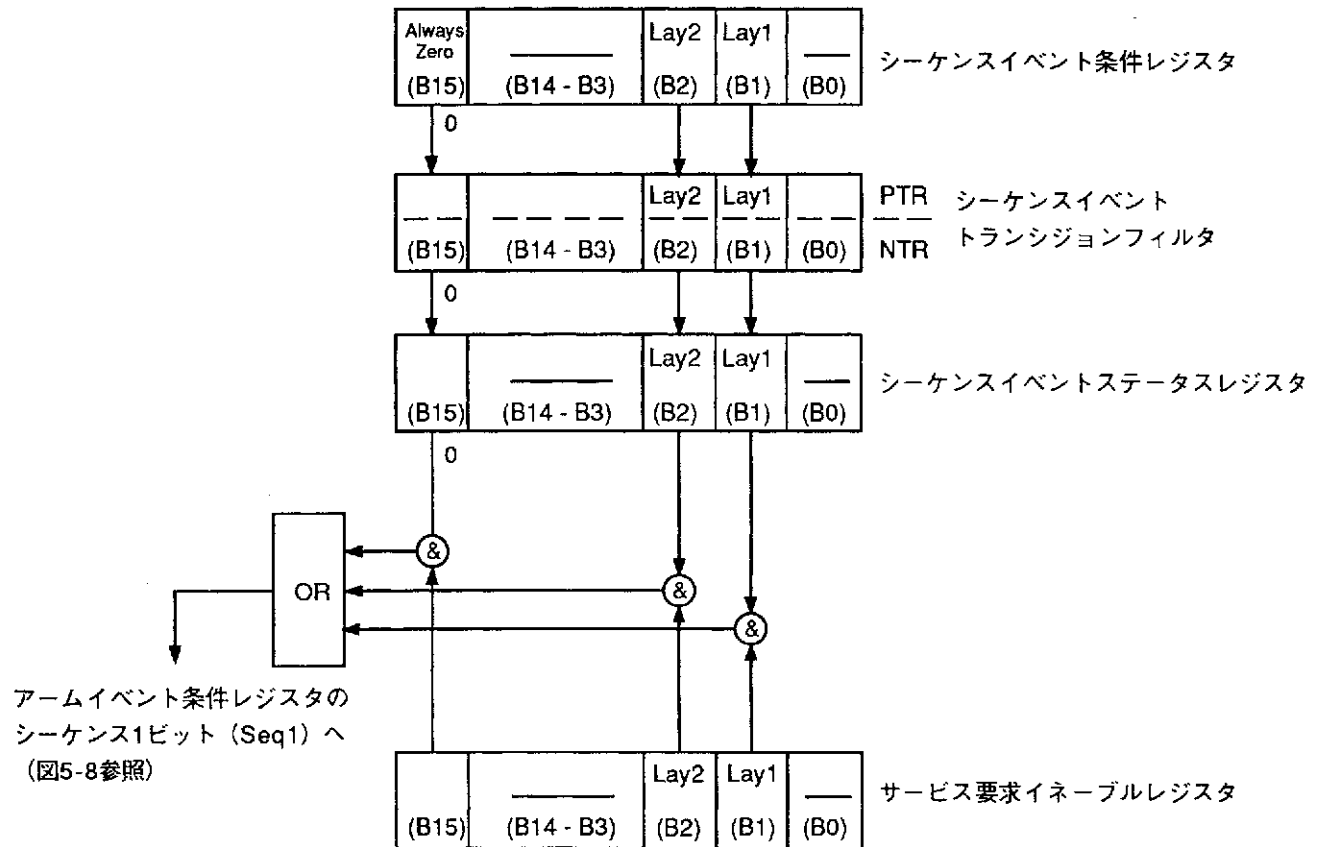
トランジションフィルタレジスタは、下記のSCPI問合せコマンドを使って随時に読み取ることができます。

```
:STATus:OPERation:ARM:SEQuence:PTRansition?
:STATus:OPERation:ARM:SEQuence:NTRansition?
```

これらの問合せコマンドを使ってトランジションフィルタレジスタを読み取っても、レジスタの内容は影響を受けません。

下記のオペレーションでは、PTRレジスタの全ビットがセット (1) され、NTRレジスタの全ビットがリセット (0) されます。

1. パワーサイクリング
2. :STATus:PRESetコマンドの送信
3. :STATus:OPERation:ARM:SEQuence:PTR 65535コマンドおよび:STATus:OPERation:ARM:SEQuence:NTR 0コマンドの送信



Lay1 = レイヤ1 (セットされたビットは、7001が最初の
アームレイヤに入っていることを表わす)

Lay2 = レイヤ2 (セットされたビットは、7001が
スキャンレイヤに入っていることを表わす)

& = 論理積

OR = 論理和

PTR = 正トランジションレジスタ

NTR = 負トランジションレジスタ

図5-9

シーケンスイベントステータス

シーケンスイベントステータスレジスタ — これはラッチされた読取り専用レジスタで、そのビットはシーケンスイベント条件レジスタおよびトランジションフィルタによってセットされます。このレジスタのビットは、いったんセットされると、特定のクリアシーケンスによってレジスタがクリアされるまでセット（ラッチ）されたままです。このレジスタのビットは、シーケンスイベントステータスイネーブルレジスタのビットとのANDがとられ、ORゲートに加えられます。ORゲートの出力は、アームイベント条件レジスタのビットB1に加えられます（5.6.3項参照）。シーケンスイベントステータスレジスタは、下記のSCPI問合せコマンドを使って読み取ることができます。

```
:STATus:OPERation:ARM:SEquence?
```

このSCPIコマンドを使ってシーケンスイベントステータスレジスタを読み取ると、レジスタはクリアされます。このレジスタをクリアするオペレーションを示すと、次の通りです。

1. パワーサイクリング
2. *CLSコモンコマンドの送信
3. :STATus:OPERation:ARM:SEquence? 問合せコマンドの送信

シーケンスイベントステータスイネーブルレジスタ — このレジスタはユーザによってプログラムされ、シーケンスイベントステータスレジスタをマスクする働きをします。マスクされると、シーケンスイベントステータスレジスタのビットがセットされても、アームイベント条件レジスタのビットB1はセットされません。逆に、アンマスクされたときは、シーケンスイベントステータスレジスタのビットがセットされると、アームイベント条件レジスタのビットB1はセットされます。

シーケンスイベントステータスレジスタのビットは、シーケンスイベントステータスイネーブルレジスタの対応するビットがクリア（0）されると、マスクされます。シーケンスイベントステータスレジスタのマスクされたビットがセットされると、シーケンスイベントステータスイネーブルレジスタの対応するクリアされたビットとのANDがとられます。ANDゲートの論理“0”出力がORゲートの入力に加えられますから、アームイベント条件レジスタのビットB1はセットされません。

シーケンスイベントステータスレジスタのビットは、シーケンスイベントステータスイネーブルレジスタの対応するビットがセット（1）されると、アンマスクされます。

シーケンスイベントステータスレジスタのアンマスクされたビットがセットされると、シーケンスイベントステータスイネーブルレジスタの対応するセットされたビットとのANDがとられます。ANDゲートの論理“1”出力がORゲートの入力に加えられますから、アームイベント条件レジスタのビットB1がセットされます。

シーケンスイベントステータスイネーブルレジスタの個々のビットは、下記のSCPIコマンドを使ってセットまたはクリアすることができます。

```
:STATus:OPERation:ARM:SEquence:ENABle<n>
```

シーケンスイベントステータスレジスタは、下記のSCPI問合せコマンドを使って読み取ることができます。

```
:STATus:OPERation:ARM:SEquence:ENABle?
```

このSCPIコマンドを使ってシーケンスイベントステータスレジスタを読み取っても、レジスタはクリアされません。このレジスタをクリアするオペレーションを示すと、次の通りです。

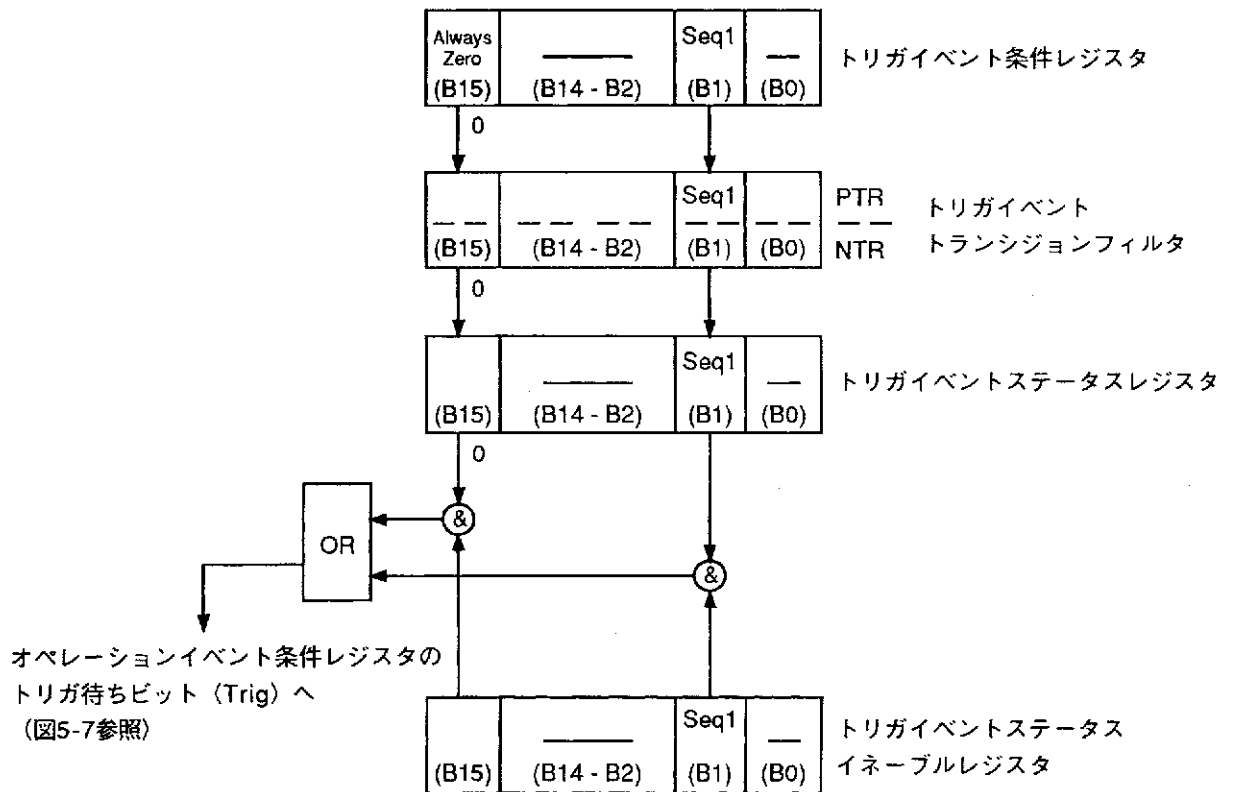
1. パワーサイクリング
2. :STATus:PRESetコマンドの送信
3. :STATus:OPERation:ARM:SEquence:ENABle 0コマンドの送信

5.6.5 トリガイイベントステータス

トリガイイベントの報告は一連の16ビットレジスタによってコントロールされます。すなわち、トリガイイベント条件レジスタ、トランジションフィルタ、トリガイイベントステータスレジスタおよびトリガイイベントステータスイネーブルレジスタによってコントロールされます。これらのレジスタの構造を図5-10に示します。

このレジスタセットのビットB1 (Seq1) は、トリガイイベント (In Trigger Layer of Sequence 1) に使われます。原則として、ビットB1は、Model 7001がチャンネルレイヤに入ると（またはそこから出ると）セットされます。スキャンプロセスについては5.7項で説明します。

以下に、トリガイイベントステータス用の各種のレジスタについて説明します。なお、これらのレジスタは、:STATusサブシステムの:STATus:OPERation:TRIGgerコマンド（5.10.6項参照）によってコントロールされます。



Seq1 = シーケンス1 (セットされたビットは、
7001がシーケンス1のトリガレイヤに
入っていることを表わす)

& = 論理積
OR = 論理和
PTR = 正トランジションレジスタ
NTR = 負トランジションレジスタ

図5-10
トリガイベントステータス

トリガイベント条件レジスタ — これはリアルタイムの16ビット読取り専用レジスタで、Model 7001のトリガレイヤステータスを反映するように絶えず更新されます。ビットB1がセットされているときは、Model 7001はトリガレイヤ（チャンネルレイヤ）に入っています。

トリガイベント条件レジスタは、下記のSCPI問合せコマンドを使って読み取ることができます。

```
:STATus:OPERation:TRIGger:CONDition?
```

トリガイベント条件レジスタとトランジションフィルタは、トリガイベントステータスレジスタのビットB1をセットするのに使われます。トランジションフィルタについては次に説明します。

トリガイベントトランジションフィルタ — このトランジションフィルタは、ユーザによってプログラムされる2つの16ビットレジスタで構成されています。このフィルタを使って、トリガイベント条件レジスタのビットB1のどちらのトランジション（0から1または1から0）によってトリガイベントステータスレジスタのビットB1をセットするかを指定します。

このフィルタは、正トランジション（PTR）または負トランジション（NTR）あるいはその両方にプログラムすることができます。イベントビットを正トランジションにプログラムした場合、トリガイベントステータスレジスタのイベントビットは、トリガイベント条件レジスタの対応するビットが0から1に変わるとセットされます。逆に、イベントビットを負トランジションにプログラムした場合、ステータスレジスタのイベントビットは、条件レジスタの対応するビットが1から0に変わるとセットされます。

トランジションフィルタレジスタは、下記のSCPIコマンドを使ってセットまたはクリアすることができます。

```
:STATus:OPERation:TRIGger:PTRansition<n>
:STATus:OPERation:TRIGger:NTRansition<n>
```

トランジションフィルタレジスタは、下記のSCPI問合せコマンドを使って随時に読み取ることができます。

```
:STATus:OPERation:TRIGger:PTRansition?
:STATus:OPERation:TRIGger:NTRansition?
```

これらの問合せコマンドを使ってトランジションフィルタレジスタを読み取っても、レジスタの内容は影響を受けません。

下記のオペレーションでは、PTRレジスタの全ビットがセット（1）され、NTRレジスタの全ビットがリセット（0）されます。

1. パワーサイクリング
2. :STATus:PRESetコマンドの送信
3. :STATus:OPERation:TRIGger:PTR 65535コマンドおよび:STATus:OPERation:TRIGger:NTR 0コマンドの送信

トリガイベントステータスレジスタ — これはラッチされた読取り専用レジスタで、そのビットはトリガイベント条件レジスタおよびトランジションフィルタによってセットされます。このレジスタのビットは、いったんセットされると、特定のクリアオペレーションによってレジスタがクリアされるまでセット（ラッチ）されたままです。このレジスタのビットは、トリガイベントステータスイネーブルレジスタのビットとのANDがとられ、ORゲートに加えられます。ORゲートの出力はオペレーションイベント条件レジスタのビットB5（トリガ待ち）に加えられます。

トリガイベントステータスレジスタは、下記のSCPI問合せコマンドを使って読み取ることができます。

```
:STATus:OPERation:TRIGger?
```

このSCPIコマンドを使ってトリガイベントステータスレジスタを読み取ると、レジスタはクリアされます。このレジスタをクリアするオペレーションを示すと、次の通りです。

1. パワーサイクリング
2. *CLSコモンコマンドの送信
3. :STATus:OPERation:TRIGger?問合せコマンドの送信

トリガイベントステータスイネーブルレジスタ — このレジスタはユーザによってプログラムされ、トリガイベントステータスレジスタをマスクする働きをします。マスクされると、トリガイベントステータスレジスタのビット（B1）がセットされても、オペレーションイベント条件レジスタのビットB5（トリガ待ち）はセットされません。逆に、アンマスクされたときは、トリガイベントステータスレジスタのビット（B1）がセットされると、オペレーションイベント条件レジスタのビットB5がセットされます。

トリガイベントステータスレジスタのビットB1は、トリガイベントステータスイネーブルレジスタの対応するビット (B1) がクリア (0) されると、マスクされます。トリガイベントステータスレジスタのマスクされたビットがセットされると、トリガイベントステータスイネーブルレジスタの対応するクリアされたビットとのANDがとられます。ANDゲートの論理“0”出力がORゲートの入力に加えられるから、オペレーションイベント条件レジスタのビットB5はセットされません。

トリガイベントステータスレジスタのビットB1は、トリガイベントステータスイネーブルレジスタの対応するビット (B1) がセット (1) されると、アンマスクされます。トリガイベントステータスレジスタのアンマスクされたビットがセットされると、トリガイベントステータスイネーブルレジスタの対応するセットされたビットとのANDがとられます。ANDゲートの論理“1”出力がORゲートの入力に加えられるから、オペレーションイベント条件レジスタのビットB5がセットされます。

トリガイベントステータスイネーブルレジスタのビットB1は、下記のSCPIコマンドを使ってセットまたはクリアすることができます。

```
:STATus:OPERation:TRIGger:ENABle<n>
```

トリガイベントステータスレジスタは、下記のSCPI問合せコマンドを使って読み取ることができます。

```
:STATus:OPERation:TRIGger:ENABle?
```

このSCPIコマンドを使ってトリガイベントステータスレジスタを読み取っても、レジスタはクリアされません。このレジスタをクリアするオペレーションを示すと、次の通りです。

1. パワーサイクリング
2. :STATus:PRESetコマンドの送信
3. :STATus:OPERation:TRIGger:ENABle 0コマンドの送信

5.6.6 被疑イベントステータス

Model 7001では被疑イベントは使いません。したがって、図5-5に示すようにQSBサマリビットは常にゼロです。被疑イベントレジスタは、使われなくても、SCPIが必要とするステータス構造の一部です。

5.6.7 待ち行列

Model 7001では、出力待ち行列とエラー/ステータス待ち行列の2つの待ち行列を使います。出力待ち行列とエラー/ステータス待ち行列は先入れ先出し方式 (FIFO) のレジスタで、それぞれ、データメッセージとエラー/ステータスメッセージを保持するのに使います。Model 7001ステータスモデル (図5-5) に、これら2つの待ち行列が他のレジスタとの関係においてどのように構成されているかを示してあります。

出力待ち行列 - 出力待ち行列は、Model 7001のノーマルオペレーションに関する全てのデータを保持するのに使います。たとえば、問合せコマンドが送信されると、その問合せに関するデータメッセージは出力待ち行列に入れられます。

データメッセージが出力待ち行列に入れられると、ステータスバイトレジスタのMessage Available (MAV) ビットがセットされます。データメッセージは、読み取られると出力待ち行列からクリアされます。出力待ち行列は、空のときはクリアされているとみなされます。出力待ち行列が空になると、ステータスバイトレジスタのMAVビットはクリアされます。

出力待ち行列からメッセージを読み取るには、Model 7001をトークにアドレッシングします。以下に示すHP BASIC 4.0によるプログラミング例は、問合せコマンドを送信し、データメッセージをコンピュータに送信し、それをCRT上に表示させるものです。

```
10 OUTPUT;707 "*"IDN?"      ! Request identification code
20 ENTER 707;AS             ! Address 7001 to talk
30 PRINT AS                  ! Display ID code
40 END
```

エラー/ステータス待ち行列 - エラー/ステータス待ち行列は、エラーメッセージおよびステータスメッセージを保持するのに使います。エラーまたはステータスイベントが発生すると、そのエラー/ステータスを定義するメッセージがエラー/ステータス待ち行列に入れられます。この待ち行列は最大10メッセージを保持します。

メッセージがエラー/ステータス待ち行列に入れられると、ステータスバイトレジスタのError Available (EAV) ビットがセットされます。エラーメッセージは、読み取られるとエラー/ステータス待ち行列からクリアされます。エラー/ステータス待ち行列は、空のときはクリアされているとみなされます。エラー/ステータス待ち行列が空になると、ステータスバイトレジスタのEAVビットはクリアされます。

エラー/ステータス待ち行列からエラーメッセージを読み取るには、下記のSCPIコマンドのどちらかを送信し、Model 7001をトークにアドレッシングします。

```
:SYSTem:ERRor?
:STATus:QUEue?
```

エラーメッセージの読取りについて詳しくは5.10.6項 (:STATus:QUEue?) および5.10.7項 (:SYSTem:ERRor?) を参照してください。

5.6.8 ステータスバイトおよびサービス要求 (SRQ)

サービス要求は、ステータスバイトレジスタとサービス要求イネーブルレジスタの2つの16ビットレジスタによってコントロールされます。これらのレジスタの構造を図5-11に示します。なお、ビットB8～B15は使われませんから図中には示してありません。

ステータスバイトレジスタ — ステータスレジスタおよび待ち行列からのサマリメッセージは、ステータスバイトレジスタの該当するビット (B2、B3、B4、B5、B7) をセットまたはクリアするのに使われます。これらのビットはラッチされており、それぞれの状態 (0または1) はもっぱらサマリメッセージ (0または1) に依存します。たとえば、標準イベントステータスレジスタが読み取られると、そのレジスタはクリアされます。そうすると、そのサマリメッセージが0にリセットされ、その結果、ステータスバイトレジスタのESBビットがリセットされます。

ステータスバイトレジスタのビットB6は下記のどちらかです。

- **Master Summary Status (MSS) ビット。**
これは、*STB?コマンドに応答して送信されるもので、セットされたビット (対応するイネーブルビットがセットされたもの) のステータスを表わします。
- **Request for Service (RQS) ビット。**
これはシリアルポールに응答して送信されるもので、どの装置がSRQラインをプルしてサービスを要求したかを表わします。

ステータスバイトレジスタの他のビットについては、5.9.14項を参照してください。

IEEE-488.2規格では、単にステータスバイトレジスタの内容を読み取るだけのコマンドとして、下記のコモン問合せコマンドを使います。

```
*STB?
```

*STB?コマンドを使ってステータスバイトレジスタを読み取るとき、ビットB6はMSSビットと呼ばれます。*STB?コマンドを使ってステータスバイトレジスタを読み取っても、そのレジスタのビットはどれもクリアされません。

IEEE-488.1規格のシリアルポールシーケンスは、ステータスバイトレジスタの読取りも行ない、この方がサービス要求 (SRQ) の検出に適しています。シリアルポールを使うとき、ビットB6はRQSビットと呼ばれます。シリアルポーリングが行なわれると、ビットB6 (RQS) はリセットされます。シリアルポーリングについては後で詳しく説明します (「シリアルポールおよびSRQ」参照)。

下記の実操作では、ステータスバイトレジスタの全ビットがクリアされます。

1. パワーサイクリング
2. *CLSコモンコマンドの送信

注

MAVビットはクリアされることもされないこともあります。

サービス要求イネーブルレジスタ — このレジスタはユーザーによってプログラムされ、ステータスバイトレジスタのStatus Summary Messageビット (B2、B3、B4、B5、B7) をマスクする働きをします。マスクされると、ステータスバイトレジスタのサマリビットがセットされても、ステータスバイトレジスタのビットB6 (MSS/RQS) はセットされません。逆に、アンマスクされたときは、ステータスバイトレジスタのサマリビットがセットされると、ビットB6はセットされます。

ステータスバイトレジスタのStatus Summary Messageビットは、サービス要求イネーブルレジスタの対応するビットがクリア (0) されると、マスクされます。ステータスバイトレジスタのマスクされたサマリビットがセットされると、サービス要求イネーブルレジスタの対応するクリアされたビットとのANDがとられます。ANDゲートの論理“0”出力がORゲートの入力に加えられるから、ステータスバイトレジスタのMSS/RQSビットはセットされません。

ステータスバイトレジスタのStatus Summary Messageビットは、サービス要求イネーブルレジスタの対応するビットがセット (1) されると、アンマスクされます。ステータスバイトレジスタのマスクされたサマリビットがセットされると、サービス要求イネーブルレジスタの対応するセットされたビットとのANDがとられます。ANDゲートの論理

“1” 出力がORゲートの入力に加えられますから、ステータスバイトレジスタのMSS/RQSビットがセットされます。

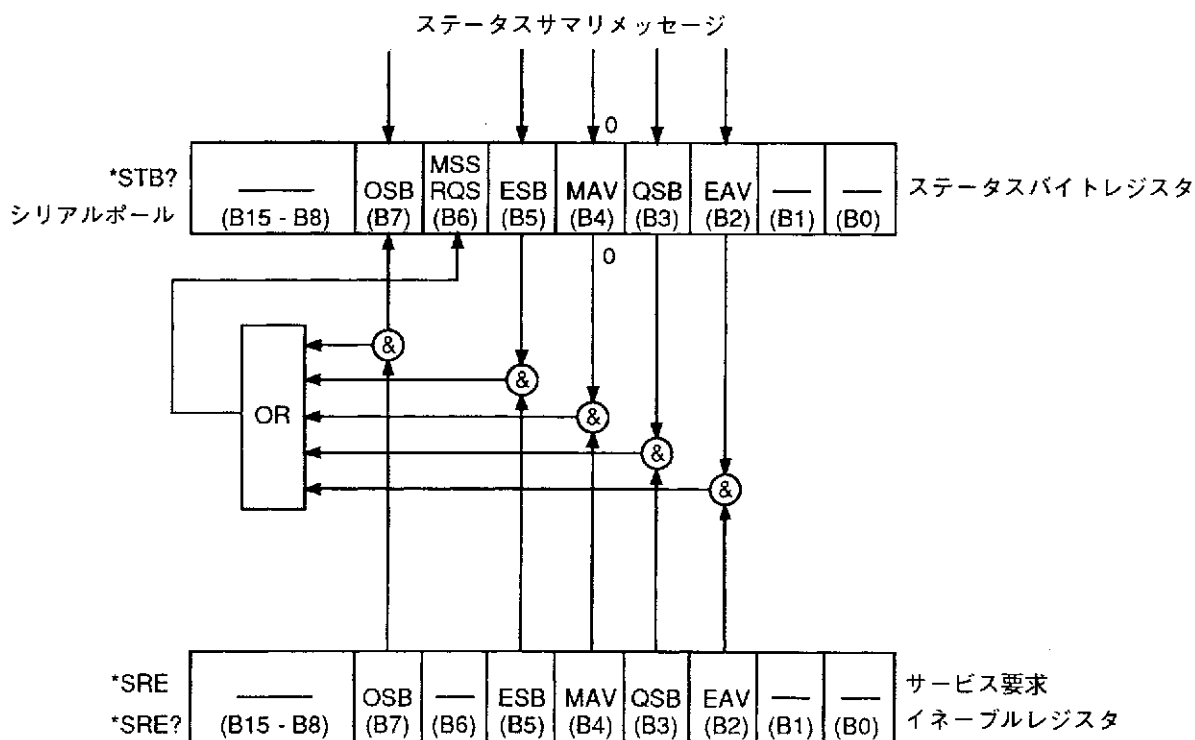
サービス要求イネーブルレジスタの個々のビットは、下記のコモンコマンドを使ってセットまたはクリアすることができます (詳細は5.9.12項を参照してください)。

*SRE<n>

サービス要求イネーブルレジスタは、下記のコモン問合せコマンドを使って読み取ることができます (詳細は5.9.13項を参照してください)。

*SRE?

*SRE?問合せコマンドを使ってサービス要求イネーブルレジスタを読み取っても、レジスタはクリアされません。このレジスタは、パワーサイクリングが行なわれるか、または*SREコマンド (*SRE 0) でゼロのパラメータ (n) 値が送信されると、クリアされます。



OSB = オペレーションサマリビット
 MSS = マスタサマリステータス (IEEE-488.2)
 RQS = サービス要求 (IEEE-488.1)
 ESB = イベントサマリビット
 MAV = メッセージアベイラブル
 QSB = 被疑サマリビット
 EAV = エラーアベイラブル

& = 論理積

OR = 論理和

図5-11

ステータスバイトおよびサービス要求 (SRQ)

シリアルポールおよびSRQ

ユーザのテストプログラムは、ステータスバイトレジスタを定期的に取り、サービス要求 (SRQ) が発生したかどうか、また、何がそれを発生させたかを調べることができます。SRQが発生した場合、テストプログラムは、たとえば、その要求に応える適切なサブルーチンへジャンプすることができます。

通常、サービス要求 (SRQ) は、Model 7001のシリアルポールシーケンスによって管理されます。SRQが発生しなければ、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なわれた後、プログラムは通常通り進行します。SRQが発生した場合は、ステータスバイトレジスタのビットB6がセットされ、プログラムはシリアルポールによってSRQが検出された時点でサービスサブルーチンへジャンプ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ビットB6を自動的にリセットします。したがって、以後のシリアルポールでも、そのビットをチェックし、次のSRQが発生したかどうかを調べることができます。

以下のHP BASIC 4.0によるプログラミング例は、シリアルポールによるSRQの検出の仕方を示すものです。

```

10  REMOTE 707
20  CLEAR 707
30  OUTPUT 707;"*ESE 32"    ! Unmask command error
                             (B5) in SESER.
40  OUTPUT 707;"*SRE 32"    ! Unmask event summary
                             (B5) in SRER.
50  OUTPUT 707;"*ESE"      ! Program command error
                             (missing parameter) to
                             generate SRQ.

60  WAIT 1
70  S=SPOLL (707)          ! Serial poll 7001.
80  IF BIT (S,6) THEN      ! Go to Service (line 200) to
    service                 service CALL Service SRQ.
90  END
200 SUB Service
210 PRINT "B7 B6 B5 B4 B3  ! Identify bits.
    B2 B1 B0"
220 FOR I=7 TO 0 STEP -1   ! Loop eight times.
230 PRINT BIT (S,I)       ! Display bit values
240 NEXT I
250 SUBEND

```

プログラム解析:

- 行10 - Model 7001をリモート状態にする。
- 行20 - 装置クリアを行なう。
- 行30 - 標準イベントステータスイネーブルレジスタのビットB5をセットする。これでコマンドエラーイベントがアンマスクされる。
- 行40 - サービス要求イネーブルレジスタのビットB5をセットする。これでイベントサマリメッセージがアンマスクされる。
- 行50 - SRQを発生させるために違法コマンドを送信する。
- 行60 - SRQを完了させるために1秒待つ。
- 行70 - SRQが発生したら、ステータスバイトレジスタビットB6がセットされ、プログラムがサービスサブルーチンへ進む。
- 行200~240 - サービスサブルーチンがステータスバイトレジスタのビット値 (0または1) を表示させる。この例では、ビットB5 (ESB) とビットB6 (RQS) がセットされる。

実際のシリアルポールシーケンスについては5.8.8項で説明します。

5.7 トリガモデル (IEEE-488スキャンプロセス)

以下に、IEEE-488バスを通じてのスキャンプロセスについて説明します。図5-12の流れ図は、バスを通じてのスキャンオペレーションの流れを示すもので、トリガモデルと呼ばれます。トリガモデルと呼ばれる理由は、オペレーションがトリガサブシステムからのSCPIコマンド (5.10.8項参照) によってコントロールされるからです。トリガモデルには主要なSCPIコマンドが含まれていますから、注意してください。

Idle

Model 7001は、トリガモデルのレイヤのどれか1つの中で動作していない限り、アイドル状態にあるとみなされます。Model 7001がアイドル状態にあるときは、フロントパネル上のARMインジケータは消えています。Model 7001はアイドル状態のときはスキャンを実行できません。

バスを通じて送信し、Model 7001をアイドル状態から抜け出させるのに使うことができるSCPIコマンドが2つあります。すなわち、:INITiateまたは:INITiate:CONTinuous ONです。Model 7001がアイドル状態でないとき（ARMインジケータが点灯しているとき）は、スキャンファンクションはイネーブルされているとみなされます。

なお、連続開始がイネーブルされているときは (:init:cont on)、Model 7001は、プログラムされた全スキャンオペレーションが完了した後もアイドル状態には戻りません。ただし、*RSTを送信することによって随時にアイドル状態に戻すことができます。また、連続開始がディスエーブルされているときは (:init:cont off)、:ABORtを送信することによってアイドル状態に戻すこともできます。

Model 7001がアイドル状態から抜け出ると、オペレーションはトリガモデルのアームレイヤ（Arm Layer 1）に入ります。

Arm Layer 1

注

フロントパネルオペレーション関係では、このレイヤをアームレイヤと呼びます。

原則として、Model 7001は、オペレーションを次のレイヤ（Arm Layer 2）に進ませるのにイベントを必要とします。Immediateコントロールソースを選択した場合には (:ARM:SOURce IMMEDIATE)、Model 7001がアイドル状態から抜け出ると、オペレーションは直ちに次のレイヤに進みます。*RSTコマンドと:SYSTem:PRESetコマンドも、アームコントロールソースをImmediateに設定します。他のコントロールソースのどれか1つを選択した場合は、Model 7001は該当するイベントが発生するまで待ちます。

Manualコントロールソースを選択した場合 (:ARM:SOURce MANual)、Model 7001はフロントパネルのSTEPキーが押されるまで待ちます。なお、Model 7001は、リモート状態から抜け出させてから（LOCALキーを押すか、またはバスを通じてLOCAL 707を送信してから）でなければ、STEPキーに回答しません。Busコントロールソースを選択した場合 (:ARM:SOURce BUS)、Model 7001はバストリガ（GETまたは*TRG）が受信されるまで待ちます。Externalコントロールソースを選択した場合 (:ARM:SOURce EXTernal)、Model 7001は入力トリガ（リヤパネル上のEXTERNAL TRIGGERコネクタ経由）が受信されるまで待ちます。Trigger Linkコントロールソースを選択した場合 (:ARM:SOURce TRIINK)、Model 7001は入力トリガ（TRIGGER LINK経由）が受信されるまで待ちます。Holdコントロールソースを選択した場合 (:ARM:SOURce HOLD)、Model 7001はいかなるコントロールソースイベントにも回答しません。

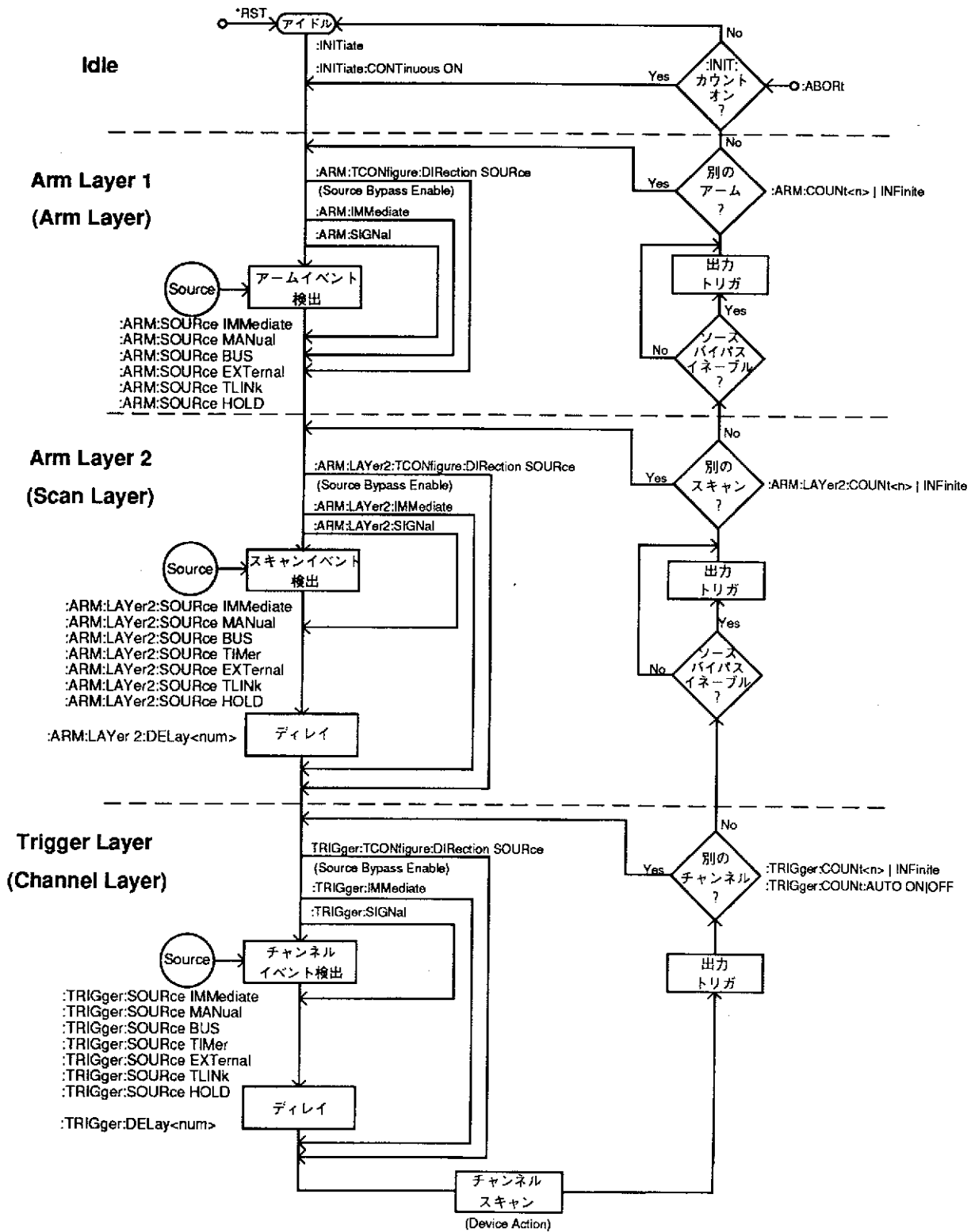


図5-12

トリガモデル (IEEE-488スキャンプロセス)

流れ図に示されているように、オペレーションをコントロールソースに対してループアラウンドさせるパスが3つあります。以下に、これら3つのパスについて説明します。

1. ソースバイパスをイネーブルし

(:ARM:TCONfigure:DIRection SOURce)、ExternalまたはTrigger Linkコントロールソースを選択した場合、オペレーションは最初にArm Layer 1を通過するときコントロールソースをループアラウンドします。さらにアームを行なうようにプログラムされている場合(アームカウント>1)、バイパスループはイネーブルされていても無効になります。ソースバイパスループは、オペレーションがIdleに入るとリセットされます(有効になりません)。

ソースバイパスをイネーブルすると、Output Triggerもイネーブルされます。オペレーションがArm Layer 2からArm Layer 1に戻ると、出力トリガパルスが発生します。Trigger Linkコントロールソースを選択した場合、出力トリガパルスにはプログラムされたTRIGGER LINK出力ラインが使われます。これ以外のコントロールソースを選択した場合、出力トリガパルスにはCHANNEL READYコネクタが使われます。このレイヤのOutput Triggerは、ソースバイパスがディスエーブルされると(:ARM:TCONfigure:DIRection ACCeptor)、ディスエーブルされます。

2. :ARM:IMMediateコマンドを送信すると、その都度、オペレーションはアームコントロールソースをループアラウンドします。このコマンドは、プログラムされたアームイベントが発生するのを待ちたくないとき(あるいはHoldコントロールソースを選択したとき)に使います。
3. シグナルパスをイネーブルすると(:ARM:SIGnal)、オペレーションは最初にArm Layer 1を通過するときアームコントロールソースをループアラウンドします。このパスは、Output Triggerをイネーブルしない点を除き、ソースバイパスと同じ働きをします。

他の全てのスキャンオペレーションが完了した後、Model 7001は、さらにアームを行なうようにプログラムすることによってArm Layer 1に戻すことができます。:ARM:COUNt <n> | INFiniteコマンドは、アームカウントを有限値(n = 1~9999)に、あるいは無限(INF)回数のアームを行なうように設定することができます。

*RSTおよび:SYSTem:PRESetコマンドはアームカウントを1に設定します。

Model 7001がArm Layer 1から出た後、オペレーションはスキャンレイヤ(Arm Layer 2)に入ります。

Arm Layer 2

注

フロントパネルオペレーション関係では、このレイヤをスキャンレイヤと呼びます。

原則として、Model 7001は、オペレーションを次のレイヤ(Trigger Layer)に進ませるのにスキャンイベントを必要とします。Immediateコントロールソースを選択した場合には(:ARM:LAYer2:SOURce IMMEDIATE)、オペレーションは直ちに次のレイヤに進みます。*RSTコマンドと:SYSTem:PRESet コマンドも、スキャンコントロールソースをImmediateに設定します。他のコントロールソースのどれか1つを選択した場合は、Model 7001は該当するイベントが発生するまで待ちます。

Timerコントロールソースを選択した場合

(:ARM:LAYer2:SOURce TIMer)、オペレーションは最初にArm Layer 2を通過すると直ちにTrigger Layerに入ります。それ以後の各スキャンは、プログラムされた時間間隔が経過するまで行なわれません。タイマは、1msecから99999.999秒までの時間間隔に設定できます。

Manualコントロールソースを選択した場合

(:ARM:LAYer2:SOURce MANual)、Model 7001はフロントパネルのSTEPキーが押されるまで待ちます。なお、Model 7001は、リモート状態から抜け出させてから(LOCALキーを押すか、またはバスを通じてLOCAL 707を送信してから)でなければ、STEPキーにตอบสนองしません。Busコントロールソースを選択した場合(:ARM:LAYer2:SOURce BUS)、Model 7001はバストリガ(GETまたは*TRG)が受信されるまで待ちます。Externalコントロールソースを選択した場合(:ARM:LAYer2:SOURce EXTernal)、Model 7001は入力トリガ(フロントパネル上のEXTERNAL TRIGGERコネクタ経由)が受信されるまで待ちます。Trigger Linkコントロールソースを選択した場合(:ARM:LAYer2:SOURce TLINK)、Model 7001は入力トリガ(TRIGGER LINK経由)が受信されるまで待ちます。Holdコントロールソースを選択した場合(:ARM:LAYer2:SOURce HOLD)、Model 7001はいかなるコントロールソースイベントにもตอบสนองしません。

プログラムされたスキャンイベントが検出された後、Model 7001はプログラムされたディレイがタイムアウトになるのを待ちます。:ARM:LAYer2:DELay <num>コマンドは、ディレイ(num = 0~99999.999秒)を設定するのに使います。*RSTおよび:SYSTem:PRESetコマンドはディレイを0秒に設定します。

流れ図に示されているように、オペレーションをコントロールソースに対してループアラウンドさせるバスが3つあります。これらのバスのうち2つはDelayもループアラウンドします。以下に、これら3つのバスについて説明します。

1. ソースバイパスをイネーブルし

(:ARM:TCONfigure:DIRection SOURce)、ExternalまたはTrigger Linkコントロールソースを選択した場合、オペレーションは最初にArm Layer 2を通過するときコントロールソースをループアラウンドします。さらにスキャンを行なうようにプログラムされている場合(スキャンカウント>1)、バイパスループはイネーブルされていても無効になります。ソースバイパスループは、オペレーションがArm Layer 1に戻るとセットされます(有効になります)。

ソースバイパスをイネーブルすると、Output Triggerもイネーブルされます。オペレーションがTrigger LayerからArm Layer 2に戻ると、出力トリガパルスが発生します。Trigger Linkコントロールソースを選択した場合、出力トリガパルスにはプログラムされたTRIGGER LINK出力ラインが使われます。これ以外のコントロールソースを選択した場合、出力トリガパルスにはCHANNEL READYコネクタが使われます。このレイヤのOutput Triggerは、ソースバイパスがディスエーブルされると(:ARM:TCONfigure:DIRection ACCeptor)、ディスエーブルされます。

2. :ARM:LAYer2:IMMediateコマンドを送信すると、その都度、オペレーションはスキャンコントロールソースおよびDelayをループアラウンドします。このコマンドは、プログラムされたスキャンイベントが発生するのを待ちたくないとき(あるいはHoldコントロールソースを選択したとき)に使います。

3. シグナルバスをイネーブルすると

(:ARM:LAYer2:SIGnal)、オペレーションは最初にArm Layer 2を通過するときスキャンコントロールソースをループアラウンドします。このバスは、Output Triggerをイネーブルしない点とDelayをループアラウンドしない点を除き、ソースバイパスと同じ働きをします。

次のレイヤでの他の全てのオペレーションが完了した後、Model 7001は、さらにスキャンを行なうようにプログラムすることによってArm Layer 2に戻ることができます。

:ARM:LAYer2:COUNT <n> | INFiniteコマンドは、スキャンカウントを有限値($n = 1 \sim 9999$)に、あるいは無限(INF)回数のスキャンを行なうように設定することができます。:RSTおよび:SYSTEM:PRESetコマンドはスキャンカウントを無限に設定します。

Model 7001がArm Layer 2から出た後、オペレーションはチャンネルレイヤ(Trigger Layer)に進みます。

Trigger Layer

注

フロントパネルオペレーション関係では、このレイヤをチャンネルレイヤと呼びます。

原則として、チャンネルスキャンレートはチャンネルイベントによってコントロールされます。Immediateコントロールを選択した場合には(:TRIGger:SOURce IMMEDIATE)、オペレーションは直ちにDelayに進みます。*RSTコマンドもチャンネルコントロールソースをImmediateに設定します。他のコントロールソースのどれか1つを選択した場合は、Model 7001は該当するイベントが発生するまで待ちます。

Timerコントロールソースを選択した場合

(:TRIGger:SOURce TIMer)、直ちに最初のチャンネルがスキャンされます。それ以降の各チャンネルは、プログラムされたタイマ時間間隔(1msec~99999.999秒)によって決められたレートでスキャンされます。

Manualコントロールソースを選択した場合

(:TRIGger:SOURce MANual)、Model 7001はフロントパネルのSTEPキーが押されるまで待ちます。なお、Model 7001は、リモート状態から抜け出させてから(LOCALキーを押すか、またはバスを通じてLOCAL 707を送信してから)でなければ、STEPキーに応答しません。:SYSTEM:PRESetコマンドもManualコントロールソースを選択します。

Busコントロールソースを選択した場合(:TRIGger:SOURce BUS)、Model 7001はバストリガ(GETまたは*TRG)が受信されるまで待ちます。Externalコントロールソースを選択した場合(:TRIGger:SOURce EXTernal)、Model 7001は入力トリガ(リヤパネル上のEXTERNAL TRIGGERコネクタ経由)が受信されるまで待ちます。Trigger Linkコントロールソースを選択した場合(:TRIGger:SOURce TRINK)、Model 7001は入力トリガ(TRIGGER LINK経由)が受信されるまで待ちます。Holdコントロールソースを選択した場合(:TRIGger:SOURce HOLD)、Model 7001はいかなるコントロールソースイベントにも応答しません。

プログラムされたチャンネルイベントが検出された後、Model 7001はプログラムされたダイレイがタイムアウトになるのを待ちます。:TRIGger:DELay <num> コマンドは、ダイレイ(num = 0~99999.999秒)を設定するのに使います。*RSTおよび:SYSTEM:PRESetコマンドはダイレイを0秒に設定します。

流れ図に示されているように、オペレーションをコントロールソースに対してループアラウンドさせるパスが3つあります。これらのパスのうち2つはDelayもループアラウンドします。以下に、これら3つのパスについて説明します。

1. ソースバイパスをイネーブルし
(:ARM:TCONfigure:DIRrection SOURCE)、ExternalまたはTrigger Linkコントロールソースを選択した場合、オペレーションは最初にArm Layer 2を通過するときコントロールソースをループアラウンドします。さらにチャンネルをスキャンするようにプログラムされている場合(チャンネルカウント>1)、バイパスループはイネーブルされていても無効になります。ソースバイパスループは、オペレーションがArm Layer 2に戻るとセットされます(有効になります)。
2. :TRIGger:IMMediateコマンドを送信すると、その都度、オペレーションはスキャンコントロールソースおよびDelayをループアラウンドします。このコマンドは、プログラムされたスキャンイベントが発生するのを待ちたくないとき(あるいはHoldコントロールソースを選択したとき)に使います。
3. シグナルパスをイネーブルする(:TRIGger:SIGnal)、オペレーションは最初にArm Layer 2を通過するときスキャンコントロールソースをループアラウンドします。このパスは、Delayをループアラウンドしない点を除き、ソースバイパスと同じ働きをします。

オペレーションがここまでくると、スキャンリスト中の最初の(または次の)チャンネルがスキャンされます(Device Action)。チャンネルをスキャンすると、前のチャンネルが開き、次のチャンネルが閉じます(ブレーク・ピフォア・メーク)。Device Actionには、内部セットリングダイレイと、:ROUteサブシステムの:STIMeコマンド(5.10.3項参照)を使って設定された付加的セットリングダイレイも含まれます。

チャンネルがスキャンされた後、出力トリガが発生します。チャンネルコントロールソースがImmediate、Manual、BusまたはTimerにプログラムされている場合、出力トリガパルスにはCHANNEL READYコネクタが使われます。Trigger Linkコントロールソースを選択した場合の出力トリガアクションは次の通りです。

1. 非同期Trigger Linkモードを選択した場合、出力トリガパルスにはプログラムされたTRIGGER LINK出力ラインが使われます。
2. 半同期Trigger Linkモードを選択し、ソースバイパスをディスエーブルした場合(:trig:tcon:dir acc)、Trigger Linkラインはリリースされます(Highになります)。
3. 半同期Trigger Linkモードを選択し、ソースバイパスをイネーブルした場合(:trig:tcon:dir sour)、Trigger LinkラインはLowにされ、それからリリースされます。

注: Trigger Linkの使い方について詳細は4.11項を参照してください。

チャンネルがスキャンされ、出力トリガが発生した後、Model 7001は、さらにチャンネルをスキャンするためにTrigger Layerの始めに戻ります(そうするようにプログラムされている場合)。:TRIGger:COUNT <n> | INFiniteコマンドは、チャンネルカウントを有限値($n = 1 \sim 9999$)に、あるいはチャンネルリストの連続スキャン(INFinite)を行なうように設定するのに使います。:TRIGger:COUNt:AUTo ONコマンドは、チャンネルカウントを自動的にスキャンリスト長に設定します。*RSTコマンドはチャンネルカウントを1に設定し、:SYSTem:PRESetコマンドは自動チャンネルカウントをイネーブルします。

5.8 汎用バスコマンド

汎用バスコマンドとは、たとえばDCLのように、Model 7001に関係なく全体を通して同じ意味をもつコマンドのことです。Model 7001によってサポートされているコマンドを表5-4に示します。この表には、各コマンドを送信するのに必要なBASICステートメントも示されています。なお、一次アドレスを指定する必要があるコマンドは、一次アドレスが16に設定されていることを前提としています。これは工場で設定されたアドレスです。

5.8.1 REN (リモートイネーブル)

リモートイネーブルコマンドは、Model 7001をリモートオペレーションに設定するためにコントローラがModel 7001へ送信するものです。原則として、バスを通じてModel 7001をプログラムするときは、その前に、Model 7001をリモートモードにしなければなりません。RENを真に設定しても、それだけではModel 7001はリモート状態にはなりません。RENを真に設定した後にModel 7001をリスンにアドレッシングしなければ、リモート状態にはなりません。

なお、Model 7001はリモート状態でもトーカになります。

プログラミング例 — このシーケンスは、下記をタイプインするとコントローラによって自動的に実行されます。

REMOTE 707

RETURNキーを押すと、Model 7001はリモート状態になります。リモート状態になったことはREMアナンシエータによって示されます。リモート状態にならない場合は、Model 7001が正しい一次アドレスに設定されているかどうか、また、バス接続が正しく行なわれているかどうかチェックしてください。

Model 7001がリモート状態になっているときは、LOCALキー（およびPOWERキー）以外のフロントパネルキーは使用不能です。LOCALキーを押すことによって通常のフロントパネルオペレーションを回復することができます。

表5-4

汎用バスコマンドおよび付随のBASICステートメント

コマンド	HP BASIC 4.0 ステートメント	Model 7001への影響
REN	REMOTE 7	次にリスンにアドレッシングされたときに有効になる。
IFC	ABORT 7	トーカおよびリスナアイドル状態になる。
LLO	LOCAL LOCKOUT 7	LOCALキーがロックアウトされる。
GTL	LOCAL 707	リモートをキャンセルし、フロントパネルオペレーションを回復する。
DCL	CLEAR 7	7001を既知の状態に戻す。
SDC	CLEAR 707	全装置を既知の状態に戻す。
GET	TRIGGER 707	トリガを発生させる。
SPE, SPD	SPOLL (707)	Model 7001をシリアルポールする。

5.8.2 IFC (インタフェースクリア)

IFCコマンドは、Model 7001をローカル、トーカーおよびリズナアイドル状態にするためにコントローラが送信するものです。Model 7001が前にこれらの状態のどれか1つに入っていた場合は、フロントパネル上のTALKまたはLISTENインジケータランプを消せば、Model 7001はIFCコマンドに応答します。

このコマンドはModel 7001のステータスには影響を及ぼしません。すなわち、各種の設定、データおよびイベントレジスタは変更されません。

IFCコマンドを送信するには、コントローラは、IFCラインを少なくとも100 μ secの間、真に設定するだけで構いません。

プログラミング例 — IFCコマンドを実際に試す前に、下記のステートメントでTALKインジケータを点灯させます。

```
ENTER 707;A$
```

現在、TALKインジケータが点灯しているはずですが、IFCコマンドは、下記のステートメントをコンピュータに入力することによって送信することができます。

```
ABORTIO 7
```

RETURNキーを押すと、TALKインジケータが消えます。つまり、Model 7001がトーカーアイドル状態になったということです。

5.8.3 LLO (ローカルロックアウト)

LLOコマンドは、Model 7001のローカルオペレーションを防止するのに使います。Model 7001がLLOコマンドを受信した後、POWERキー以外の全てのフロントパネルキーは使用不能になります。この場合、LOCALキーを押しても、フロントパネルにコントロールは戻りません。

プログラミング例 — 下記のステートメントを入力することによってLLOコマンドを送信します。

```
REMOTE 707 LOCAL LOCKOUT 7
```

2番目のステートメントが実行された後、LOCALキーを含めたModel 7001の全てのフロントパネルキーがロックアウトされます。

LLOをアサートした後、フロントパネルオペレーションを回復するには、

```
LOCAL 7
```

を入力します。

5.8.4 GTL (go to local) とローカル

GTLコマンドは、Model 7001をリモート状態から抜け出させるのに使います。LLOが有効になっていない限り、フロントパネルオペレーションもGTLによって回復されます。LLOをキャンセルするには、RENを偽に設定しなければなりません。

プログラミング例 — 下記のステートメントでModel 7001をリモート状態にします。

```
REMOTE 707
```

そして、下記のステートメントでGTLを送信します。

```
LOCAL 707
```

これで、Model 7001がローカル状態になり、フロントパネルオペレーションが回復しました（この場合、LLOが無効になっているものとします）。

5.8.5 DCL (装置クリア)

DCLコマンドは、Model 7001をクリアし、既知の状態に戻すのに使うことができます。なお、DCLコマンドはアドレスによるコマンドではありませんから、DCLを実行するよりに装備された全ての機器は同時にDCLを実行します。

Model 7001は、DCLコマンドを受信すると、入力バッファと出力待ち行列をクリアし、実行猶予中のコマンドをキャンセルし、他の装置コマンドの処理を妨げるコマンドをすべてクリアします。DCLは、機器設定、記憶データ、チャンネルリスト、スキャンリストには影響を及ぼしません。

プログラミング例 — 下記のステートメントでDCLを実行します。

```
CLEAR 7
```

RETURNキーを押すと、Model 7001はDCLを実行します。

5.8.6 SDC (選択的装置クリア)

SDCコマンドはアドレスによるコマンドで、実質上はDCLコマンドと同じ機能を果たします。ただし、各装置を個別にアドレッシングしなければなりませんから、SDCコマンドは、DCLの場合のように全機器を同時にクリアするのではなく、選択された機器だけをクリアするのに使えます。

プログラミング例 — 下記のステートメントでSDCを実行します。

```
CLEAR 707
```

これでIEEE-488コマンドUNT UNL LISTEN 07 SDCが送信されます。ステートメントが実行されると、Model 7001はSDCを実行します。

5.8.7 GET (グループ実行トリガ)

GETはGPIBトリガで、スキャンプロセスをコントロールするためのアームイベント、スキャンイベントおよび/またはチャンネルイベントとして使います。Model 7001がこのトリガに応答するのは、それがコントロールソースとしてプログラムされている場合です。コントロールソースは、SCPI :TRIGgerサブシステム (5.10.8項参照) からプログラムします。

Model 7001がプログラムされ、GPIBトリガを待っているとき、下記のプログラミングステートメントでGETを実行します。

```
TRIGGER 707
```

これでIEEE-488コマンドUNT UNL LISTEN 07 GETが送信されます。ENTERキーを押すと、トリガイイベントが発生します。(コマンドTRIGGER 7はGETを送信するだけです。ENTERキーを押すと、他のリスナがトリガされます)。

5.8.8 SPE、SPD (シリアルポーリング)

シリアルポーリングシーケンスは、Model 7001のシリアルポールバイトの情報を得るのに使います。シリアルポールバイトには、5.6.8項に記したように、内部ファンクションに関する重要な情報が入っています。原則として、シリアルポーリングシーケンスは、複数の機器のうちのどれがSRQラインでサービスを要求したかを確認するためにコントローラを使います。ただし、シリアルポーリングシーケンスは、Model 7001からステータスバイトの情報を得るために随時に実行できます。

プログラミング例 — SPOLLステートメントは、シリアルポールシーケンスを自動的に実行します。シリアルポーリングを実際に試すために、下記のステートメントをコンピュータに入力します。

```
S=SPOLL (707)
PRINT"SPOLL BYTE=" ;S
```

最初のステートメントが実行されると、コントローラがシリアルポーリングシーケンスを実行します。2番目のステートメントが実行されると、シリアルポールバイトの10進値がコントローラのCRT上に表示されます。

5.9 コモンコマンド

コモンコマンドとは、バス上の全ての装置に共通する装置コマンドのことです。これらのコマンドは、IEEE-488.2規格によって命名、定義されているものです。

表5-5は、Model 7001で使われるコモンコマンドをアルファベット順に示したものです。以下に、これらのコマンドについて、HP BASIC 4.0によるプログラミング例を織り混ぜて詳しく説明します。

表5-5

IEEE-488.2コモンコマンドおよび問合せ

モニタック	名前	説明
*CLS	Clear status	全イベントレジスタおよびエラー待ち行列をクリアする。
*ESE<n>	Event status enable command	標準イベントステータスイネーブルレジスタの内容をセットする。
*ESE?	Event status enable query	標準イベントステータスイネーブルレジスタのプログラム値を要求する。
*ESR?	Event status register query	標準イベントステータスレジスタのプログラム値を要求し、それをクリアする。
*IDN?	Identification query	Model 7001のメーカー名、型番、シリアルナンバーおよびファームウェア改訂水準を要求する。
*OPC	Operation complete command	保留中の全コマンドが実行された後、標準イベントステータスレジスタのOperation Completeビットをセットする。
*OPC?	Operation complete query	選択された保留中の全装置オペレーションが実行された後、ASCII "1" を出力待ち行列に入れる。
*OPT?	Option identification query	各スロットごとに指定された型番を要求する。
*RCL<n>	Recall command	Model 7001を、指定されたメモリロケーションにストアされているセットアップコンフィギュレーションに戻す。
*RST	Reset command	Model 7001を*RSTデフォルトに戻す。
*SAV<n>	Save command	セットアップコンフィギュレーションを指定されたメモリロケーションにセーブする。
*SRE<n>	Service request enable command	サービス要求イネーブルレジスタの内容をセットする。
*SRE?	Service request enable query	Model 7001がサービス要求イネーブルレジスタの値に戻る。
*STB?	Read status byte query	ステータスバイトレジスタの値を返す。
*TRG	Trigger command	このコマンドは、グループ実行トリガ (GET) コマンドと同じ効果をもつバストリガを発生させる。
*TST?	Self-test query	この問合せが送信されると、Model 7001がROMのチェックサムテストを実行し、結果を返す。
*WAI	Wait-to-continue command	前の全コマンドが実行されるまで待つ。

5.9.1 *CLS — clear status

目的 ステータスレジスタおよびエラー待ち行列をクリアする。

フォーマット *CLS

説明 *CLSコマンドは、Model 7001の下記のレジスタのビットをクリア (0にリセット) するのに使います。

標準イベントステータスレジスタ
オペレーションイベントステータスレジスタ
エラー待ち行列
トリガイベントステータスレジスタ
シーケンスイベントステータスレジスタ
アームイベントステータスレジスタ

このコマンドはまた、Model 7001を*OPCアイドル状態および*OPC?アイドル状態にします。

プログラミング例 OUTPUT 707;"*CLS" !Clear registers and Error Queue

5.9.2 *ESE — event status enable

目的	標準イベントステータスイネーブルレジスタの内容をセットする。																
フォーマット	*ESE <n>																
パラメータ	<table border="0"> <tr> <td>n=0</td> <td>レジスタをクリア (リセット) する。</td> </tr> <tr> <td>n=1</td> <td>イネーブルレジスタのOPC (B0) をセットする。</td> </tr> <tr> <td>n=4</td> <td>イネーブルレジスタのQYE (B2) をセットする。</td> </tr> <tr> <td>n=8</td> <td>イネーブルレジスタのDDE (B3) をセットする。</td> </tr> <tr> <td>n=16</td> <td>イネーブルレジスタのEXE (B4) をセットする。</td> </tr> <tr> <td>n=32</td> <td>イネーブルレジスタのCME (B5) をセットする。</td> </tr> <tr> <td>n=64</td> <td>イネーブルレジスタのURQ (B6) をセットする。</td> </tr> <tr> <td>n=128</td> <td>イネーブルレジスタのPON (B7) をセットする。</td> </tr> </table>	n=0	レジスタをクリア (リセット) する。	n=1	イネーブルレジスタのOPC (B0) をセットする。	n=4	イネーブルレジスタのQYE (B2) をセットする。	n=8	イネーブルレジスタのDDE (B3) をセットする。	n=16	イネーブルレジスタのEXE (B4) をセットする。	n=32	イネーブルレジスタのCME (B5) をセットする。	n=64	イネーブルレジスタのURQ (B6) をセットする。	n=128	イネーブルレジスタのPON (B7) をセットする。
n=0	レジスタをクリア (リセット) する。																
n=1	イネーブルレジスタのOPC (B0) をセットする。																
n=4	イネーブルレジスタのQYE (B2) をセットする。																
n=8	イネーブルレジスタのDDE (B3) をセットする。																
n=16	イネーブルレジスタのEXE (B4) をセットする。																
n=32	イネーブルレジスタのCME (B5) をセットする。																
n=64	イネーブルレジスタのURQ (B6) をセットする。																
n=128	イネーブルレジスタのPON (B7) をセットする。																
デフォルト	<table border="0"> <tr> <td>電源投入</td> <td>レジスタをクリア</td> </tr> <tr> <td>*CLS</td> <td>無効</td> </tr> <tr> <td>*RST</td> <td>無効</td> </tr> <tr> <td>:STATus:PRESet</td> <td>無効</td> </tr> <tr> <td>DCL、SDC</td> <td>無効</td> </tr> </table>	電源投入	レジスタをクリア	*CLS	無効	*RST	無効	:STATus:PRESet	無効	DCL、SDC	無効						
電源投入	レジスタをクリア																
*CLS	無効																
*RST	無効																
:STATus:PRESet	無効																
DCL、SDC	無効																
説明	<p>*ESEコマンドは、標準イベントステータスイネーブルレジスタの内容をセットするのに使います。このコマンドは、レジスタのビットの目的の状態 (0または1) を決める2進値と等値の10進値を付して送信されます。</p> <p>このレジスタは、標準イベントステータスレジスタに対するマスクとして使われます。標準イベントがマスクされると、そのイベントが発生しても、イベントサマリビット (ESB) はセットされません。逆に、標準イベントがアンマスク (イネーブル) されたときは、そのイベントが発生すると、ESBビットはセットされます。標準イベントステータスレジスタおよび標準イベントビットについては5.9.5項を参照してください。ステータスバイトレジスタについては5.6.8項に説明してあります。</p> <p>イネーブルレジスタのビットがクリアされると (ビットが“0”にセットされると)、ステータスバイトレジスタのESBビットは、該当する標準イベントが発生してもセットされません (マスクされます)。イネーブルレジスタのビットがセットされると (ビットが“1”にセットされると)、ESBビットは、該当する標準イベントが発生するとセットされます。ステータスレジスタ構造の詳細は5.6項を参照してください。</p> <p>標準イベントステータスイネーブルレジスタを図5-13に示します。図中には各ビットの10進重みが表示されています。セットしたいビットの10進重みの和が*ESEコマンドで送信される値です。たとえば、標準イベントステータスイネーブルレジスタのCMEビットとQYEビットをセットするには、下記のコマンドを送信します。</p>																

*ESE 36

ここで、 CME (ビットB5) =10進値 32
 QYE (ビット2) = 10進値 4
 パラメータ = 36

コマンドエラー (CME) が発生すると、標準イベントステータスレジスタのビットB5がセットされます。問合せエラー (QYE) が発生すると、標準イベントステータスレジスタのビットB2がセットされます。これらのイベントは両方ともアンマスク (イネーブル) されていますから、それらのどちらか1つが発生すると、ステータスパイトレジスタのESBビットがセットされます。

標準イベントステータスイネーブルレジスタの内容は、*ESE?問合せコマンドを使って読み取ることができます (次項参照)。

プログラミング例

OUTPUT 707; "*ESE 24"

! Set bits B3 (EXE) and B4 (DDE) of the Standard Event Status Enable Register.

ビット位置	B7	B6	B5	B4	B3	B2	B1	B0
イベント	PON	URQ	CME	DDE	EXE	QYE	—	OPC
10進重み	128 (2 ⁷)	64 (2 ⁶)	32 (2 ⁵)	16 (2 ⁴)	8 (2 ³)	4 (2 ²)	— (2 ¹)	1 (2 ⁰)
値	0/1	0/1	0/1	0/1	0/1	0/1	—	0/1

値 : 1 = 標準イベントをネーブルする。
 0 = 標準イベントをディスエーブル (マスク) する。

イベント : PON = 電源投入
 URQ = ユーザ要求
 CME = コマンドエラー
 DDE = 装置依存エラー
 EXE = 実行エラー
 QYE = 問合せエラー
 OPC = オペレーション完了

図5-13

標準イベントステータスイネーブルレジスタ

5.9.3 *ESE? — event status enable query

目的 標準イベントステータスイネーブルレジスタの内容を読み取る。

フォーマット *ESE?

説明 このコマンドは、標準イベントステータスイネーブルレジスタの値（10進値）を取得するのに使います。10進値と等値の2進値によって、レジスタのどのビットがセットされるかが決まります。*ESE?問合せコマンドが送信されると、10進値が出力待ち行列に入れられます。Model 7001がトークにアドレッシングされると、その値が出力待ち行列からコンピュータへ送られます。

たとえば、取得された10進値が48の場合、それと等値の2進値は00110000です。この2進値に対して、標準イベントステータスイネーブルレジスタのビットB4およびB5がセットされます。

*ESE?問合せコマンドは、標準イベントステータスイネーブルレジスタをクリアしません。

プログラミング例

```

10  OUTPUT 707;"*ESE?"      ! Send query command
20  ENTER 707;AS           ! Address 7001 to talk
30  PRINT AS               ! Display value of register contents
40  END

```

5.9.4 *ESR? — event status register query

目的 標準イベントステータスレジスタの内容を読み取り、レジスタをクリアする。

フォーマット *ESR?

デフォルト

電源投入	レジスタをクリア
*CLS	レジスタをクリア
*RST	無効
:STATus:PRESet	無効
DCL、SDC	無効

説明 このコマンドは、標準イベントステータスレジスタ（図5-14参照）の値（10進値）を取得するのに使います。返された10進値と等値の2進値によって、レジスタのどのビットがセットされるかが決まります。*ESR?コマンドが送信されると、10進値が出力待ち行列に入れられ、標準イベントステータスレジスタがクリアされます。Model 7001がトークにアドレッシングされると、出力待ち行列内の値がコンピュータに送られます。

このレジスタのビットがセットされたということは、特定のイベントが発生したということです。たとえば、取得された10進値が48の場合、それと等値の2進値は00110000です。この2進値に対して、標準イベントステータスレジスタのビットB4およびB5がセットされます。これらのビットがセットされたということは、装置依存エラーおよびコマンドエラーが発生したということです。

以下に、標準イベントステータスレジスタの各ビットについて説明します。

ビットB0、Operation Complete (OPC) — このビットがセットされたということは、選択された保留中の装置オペレーションがすべて完了し、Model 7001が新しいコマンドを受け付けることが可能になったということです。このビットは、*OPC?問合せコマンドだけに応答してセットされます。

ビットB1 — 使用せず

ビットB2、Query Error (QYE) — このビットがセットされたということは、空の出力待ち行列からデータを読み取ろうとしたということです。

ビットB3、Device-dependent Error (DDE) — このビットがセットされたということは、何等かの内部条件により機器オペレーションが正常に行なわれなかったということです。

ビットB4、Execution Error (EXE) — このビットがセットされたということは、コマンドを実行しようとしているときにModel 7001がエラーを検出したということです。

ビットB5、Command Error (CME) — このビットがセットされたということは、コマンドエラーが発生したということです。コマンドエラーとしては次のものがあります。

1. IEEE-488.2シンタックスエラー： Model 7001がIEEE-488.2規格に定義されたシンタックスに従わないメッセージを受信した。

2. 意味論的エラー： Model 7001が綴り間違いのコマンドを受信したか、またはModel 7001では使わないオプションのIEEE-488.2コマンドを受信した。
3. Model 7001がプログラムメッセージの中のグループ実行トリガ (GET) を受信した。

ビットB6、User Request (URQ) — このビットがセットされたということは、Model 7001フロントパネル上のLOCALキーが押されたということです。

ビットB7、Power On (PON) — このビットがセットされたということは、このレジスタが最後に読み取られてから以降、Model 7001の電源が切られ、再び投入されたということです。

プログラミング例

```

10  OUTPUT 707;“*ESE?”      ! Request register value
20  ENTER 707;A$           ! Address 7001 to talk
30  PRINT A$               ! Display value

```

ビット位置	B7	B6	B5	B4	B3	B2	B1	B0
イベント	PON	URQ	CME	EXE	DDE	QYE	—	OPC
10進重み	128 (2 ⁷)	64 (2 ⁶)	32 (2 ⁵)	16 (2 ⁴)	8 (2 ³)	4 (2 ²)	— (2 ¹)	1 (2 ⁰)
値	0/1	0/1	0/1	0/1	0/1	0/1	—	0/1

値 : 1 = イベントビットがセットされた。
0 = イベントビットがクリアされた。

イベント : PON = 電源投入
URQ = ユーザ要求
CME = コマンドエラー
EXE = 実行エラー
DDE = 装置依存エラー
QYE = 問合せエラー
OPC = オペレーション完了

図5-14

標準イベントステータスレジスタ

5.9.5 *IDN? — identification query

目的 Model 7001の識別コードを読み取る。

フォーマット *IDN?

説明 *IDN?問合せコマンドは、Model 7001の識別コードを出力待ち行列に入れます。Model 7001がトークにアドレッシングされると、識別コードがコンピュータへ送られます。

識別コードは、メーカー名、型番、シリアルナンバーおよびファームウェア改訂水準を含み、下記のフォーマットで送られます。

KEITHLEY INSTRUMENTS INC., MODEL 7001, ??????, ZZZZZZZZ

ここで、 ??????はシリアルナンバー、
ZZZZZZZZはデジタルボード用ROMと表示ボード用ROMのファームウェア改訂水準。

プログラミング例	10	OUTPUT 707; "*IDN?"	! Request 7001 ID data
	20	ENTER 707; A\$! Address 7001 to talk
	30	PRINT AS	! Display identification code

5.9.6 *OPC — operation complete

目的 保留中の全コマンドが実行完了した後、標準イベントステータスレジスタのOperation Completeビットをセットする。

フォーマット *OPC

説明 *OPCコマンドが送信された後、保留中の最後のコマンドが実行完了すると直ちに標準イベントステータスレジスタのOperation Completeビット（ビットB0）がセットされます。

プログラミング例

10	OUTPUT 707;“*ESE 1”	! Unmask bit 0 of enable register
20	OUTPUT 707;“*OPC”	! Send *OPC command
30	WAIT 0.5	! Allow pending operations to finish
40	OUTPUT 707;“*ESR?”	! Request value of Standard Event Status Register
50	ENTER 707;A\$! Address 7001 to talk
60	PRINT A\$! Display value of Standard Event Status Register
70	END	

5.9.7 *OPC? — operation complete query

目的 保留中のオペレーションが完了した後、“1”を出力待ち行列に入れる。

フォーマット *OPC?

説明 このコマンドが送信されると、保留中の最後のオペレーションが完了した後、ASCII “1”が出力待ち行列に入れます。Model 7001がトークにアドレッシングされると、“1”がコンピュータへ送られます。

プログラミング例

10	OUTPUT 707;“*OPC?”	! Send *OPC? command
20	ENTER 707;AS	! Address 7001 to talk
30	PRINT AS	! Display “1” on CRT
40	END	

5.9.8 *OPT? — option identification query

目的 Model 7001に装着されたオプションスイッチカードの識別番号を読み取る。

フォーマット *OPT?

説明 オプション識別コードは、メインフレームの各スロットごとに指定された型番を示します。たとえば、スロット1に7011、スロット2に7012が装着されていれば、下記のメッセージがコンピュータへ送られます。

7011, 7012

プログラミング例	10	OUTPUT 707;"OPT?"	! Request option ID code
	20	ENTER 707;A\$! Address 7001 to talk
	30	PRINT A\$! Display option ID code

5.9.9 *RCL — recall

目的	Model 7001を、前にメモリにストアされたセットアップコンフィギュレーションに戻す。
フォーマット	*RCL <n>
パラメータ	n = 0~9のメモリロケーション
説明	<p>Model 7001は、最大10セットアップコンフィギュレーションをメモリロケーション0~9にストアすることができます。セットアップコンフィギュレーションをセーブするには、バスを通じて*SAVコマンドを送信します (5.9.11項参照)。</p> <p>*RCLコマンドは、Model 7001を、メモリロケーションのどれか1つにストアされているセットアップコンフィギュレーションに戻すのに使います。たとえば、前にメモリロケーション9にセットアップコンフィギュレーションがストアされた場合、下記のコマンドを送信することによってModel 7001をそのセットアップ状態に戻すことができます。</p> <p style="text-align: center;">*RCL 9</p>
プログラミング例	<p>下例は、メモリロケーション6にセットアップコンフィギュレーションがストアされていることを前提とするものです (セットアップコンフィギュレーションのストアについては5.9.11項を参照してください)。</p> <p>OUTPUT 707;“*RCL 6” ! Recall setup stored at memory location 6.</p>

5.9.10 *RST — reset

目的 Model 7001をリセットする。

フォーマット *RST

説明 *RSTコマンドが送信されると、Model 7001は以下のオペレーションを実行します。

1. Model 7001を*RSTデフォルト状態に戻す（表5-6参照）。
2. 保留中の全オペレーションをキャンセルする。
3. 前に受信された*OPCおよび*OPC?コマンドへの応答をキャンセルする。

下記の状態はこのコマンドの影響を受けません。

IEEE-488アドレス
標準イベントステータスレジスタ
ステータスバイトレジスタ

プログラミング例 OUTPUT 707;“*RST” ! Return 7001 to power-up defaults

図5-6

デフォルト

コマンド名	*RST値	:SYSTem:PRESet値
:INITiate		
:CONTinuous	OFF	OFF
:ARM		
:SEQuence[1]		
:LAYer[1]		
:COUNt	1	1
:SOURce	IMMediate	IMMediate
:TCONfigure		
:DIRection	ACcEptor	ACcEptor
:ASYNchronous		
:ILINe	1	1
:OLINe	2	2
:LAYer2		
:COUNt	1	INFinite
:DELay	0	0
:SOURce	IMMediate	IMMediate
:TIMer	0.001	0.001
:TCONfigure		
:DIRection	ACcEptor	ACcEptor
:ASYNchronous		
:ILINe	1	1
:OLINe	2	2
:TRIGger		
:SEQuence1		
:COUNt	1	Scan list length
:AUTO	OFF	ON
:DELay	0	0
:SOURce	IMMediate	MANual
:TIMer	0.001	0.001
:TCONfigure		
:PROTocol	ASYNchronous	ASYNchronous
:DIRection	ACcEptor	SOURce
:ASYNchronous		
:ILINe	1	1
:OLINe	2	2
:SSYNchronous		
:LINE	1	1
[:ROUte]		
:CONFigure		
:SLOT[1]		
:STIMe	0	0
:SLOT2		
:STIMe	0	0
:CPAir	OFF	OFF
:SCHannel	OFF	OFF
:DISPlay		
:SMESsage	OFF	OFF

5.9.11 *SAV — save

目的	現在のセットアップコンフィギュレーションをメモリにセーブする。
フォーマット	*SAV <n>
パラメータ	n = 0~9のメモリロケーション
説明	Model 7001は、*SAVコマンドを使うことで、最大10セットアップコンフィギュレーションをメモリロケーション0~9にストアすることができます。たとえば、現在のセットアップコンフィギュレーションをメモリロケーション6にストアするには、下記のコマンドを送信します。 *SAV 6 *SAVコマンドによってセーブできるセットアップ項目を（それらをプログラムするためのバスコマンドと共に）表5-7に示します。
プログラミング例	OUTPUT 707;“*SAV6” ! Store setup at memory location 6.

表5-7
*SAVおよび*RCLオプション

コマンド名
:INITiate
:CONTInuous
:ARM
:SEQuence[1]
:LAYer[1]
:COUNT
:SOURce
:TCONfigure
:DIRection
:ASYNchronous
:ILINe
:OLINe
:LAYer2
:COUNT
:DELay
:SOURce
:TIMer
:TCONfigure
:DIRection
:ASYNchronous
:ILINe
:OLINe
:TRIGger
:SEQuence1
:COUNT
:AUTO
:DELay
:SOURce
:TIMer
:TCONfigure
:PROTOcol
:DIRection
:ASYNchronous
:ILINe
:OLINe
:SSYNchronous
:LINE
[:ROUTE]
:CONFigure
:SLOT[1]
:STIMe
:SLOT2
:STIMe
:CPAir
:SCHannel
:DISPlay
:SMESsage

5.9.12 *SRE — service request enable

目的	サービス要求イネーブルレジスタの内容をセットする。	
フォーマット	*SRE <n>	
パラメータ	n=0	イネーブルレジスタをクリアする。
	n=4	EAVビット (ビット2) をセットする。
	n=8	QSBビット (ビット3) をセットする。
	n=16	MAVビット (ビット4) をセットする。
	n=32	ESBビット (ビット5) をセットする。
	n=128	OSBビット (ビット7) をセットする。
デフォルト	電源投入	レジスタをクリア
	*CLS	無効
	*RST	無効
	:STATus:PRESet	無効
	DCL、SDC	無効
説明	*SREコマンドは、サービス要求イネーブルレジスタの内容をセットするのに使います。このコマンドは、レジスタの各ビットの目的の状態 (0または1) を決める2進値と等値の10進値を付して送信されます。	

サービス要求イネーブルレジスタは、ステータスバイトレジスタと共に使い、サービス要求 (SRQ) を発生させます。サービス要求イネーブルレジスタのビットがセットされると、ステータスバイトの対応するビットが適当なイベントによってセットされたとき、SRQが発生します。ステータス構造の詳細は5.6項を参照してください。

サービス要求イネーブルレジスタを図5-15に示します。図中には各ビットの10進重みが表示されています。セットしたいビットの10進重みの和が*SREコマンドで送信される値です。たとえば、サービス要求イネーブルレジスタのESBビットとSSBビットをセットするには、下記のコマンドを送信します。

*SRE 34

ここで、	ESB (ビットB5) = 10進値	32
	MAV (ビットB4) = 10進値	16
	パラメータ =	<u>48</u>

サービス要求イネーブルレジスタの内容は、*SRE? 問合せコマンド (次項参照) を使って読み取ることができます。

プログラミング例

OUTPUT707;“*SRE 32”

! Set ESB bit

ビット位置	B7	B6	B5	B4	B3	B2	B1	B0
イベント	OSB	—	ESB	MAV	QSB	EAV	—	—
10進重み	128 (2 ⁷)	— (2 ⁶)	32 (2 ⁵)	16 (2 ⁴)	8 (2 ³)	4 (2 ²)	— (2 ¹)	— (2 ⁰)
値	0/1	—	0/1	0/1	0/1	0/1	—	—

値 : 1 = サービス要求イベントをイネーブルする。
0 = サービス要求イベントをディスエーブル (マスク) する。

イベント : OSB = オペレーションサマリビット
ESB = イベントサマリビット
MAV = メッセージアベイラブル
QSB = 被疑サマリビット
EAV = エラーアベイラブル

図5-15

サービス要求イネーブルレジスタ

5.9.13 *SRE? — service request enable query

目的 サービス要求イネーブルレジスタの内容を読み取る。

フォーマット *SRE?

説明 *SRE? コマンドは、サービス要求イネーブルレジスタの値（10進値）を取得するのに使います。10進値と等値の2進値によって、レジスタのどのビットがセットされるかが決まります。*SRE? コマンドが送信されると、10進値が出力待ち行列に入れられます。Model 7001がトークにアドレッシングされると、その値が出力待ち行列からコンピュータへ送られます。

たとえば、取得された10進値が17の場合、それと等値の2進値は00010001です。この2進値は、サービス要求イネーブルレジスタのビット4とビット0がセットされることを表わします（前項参照）。

プログラミング例	10	OUTPUT 707;“*SRE?”	! Request contents of SRER
	20	ENTER 707;A\$! Address 7001 to talk
	30	PRINT A\$! Display value register

5.9.14 *STB? — status byte query

目的 ステータスバイトレジスタの内容を読み取る。

フォーマット *STB?

説明 *STB?問合せコマンドは、ステータスバイトレジスタの値（10進値）を取得するのに使います。ステータスバイトレジスタを図5-16に示します。10進値と等値の2進値によって、レジスタのどのビットがセットされるかが決まります。このレジスタのセットされたビットで、Model 7001の動作状態が定義されます。

*STB?問合せコマンドが送信されると、10進値が出力待ち行列に入れられます。Model 7001がトークにアドレッシングされると、その値が出力待ち行列からコンピュータへ送られます。

このレジスタのビット6以外の全ビットは、他のイベントレジスタおよび待ち行列によってセットされます。ビット6は、1つまたはそれ以上のイネーブルされた状態が発生するとセットされます。

*STB?問合せコマンドは、ステータスバイトレジスタをクリアしません。このレジスタは、関連したレジスタおよび待ち行列をクリアすることによってしかクリアできません。レジスタおよび待ち行列の構造については5.6項に説明してあります。

たとえば、取得された10進値が48の場合、それと等値の2進値は00110000です。この2進値は、ステータスバイトレジスタのビット4とビット5がセットされることを表わします。

以下に、ステータスバイトレジスタのビットについて説明します。

ビット0 — 使用せず

ビット1 — 使用せず

ビット2、Error Available (EAV) — このビットがセットされたということは、エラー待ち行列にエラーメッセージが入っているということです。エラーメッセージは、下記のSCPIコマンドの1つを使って読み取ることができます。

```
:SYSTem:ERRor?
```

```
:STATus:QUEue?
```

詳細は5.10.7項および5.10.6項を参照してください。

ビット3、Questionable Summary Bit (QSB) — 常に0です (Model 7001では被疑イベントは使いません)。

ビット4、Message Available (MAV) — このビットがセットされたということは、出力待ち行列にメッセージが入っているということです。Model 7001がトークにアドレッシングされると、メッセージはコンピュータへ送られます。

ビット5、Event Summary Bit (ESB) — このビットがセットされたということは、イネーブルされた標準イベントが発生したということです。イベントは、*ESE?問合せコマンドで標準イベントステータスレジスタを読み取ることによって識別できます (5.9.2項参照)。

ビット6、Master Summary Status (MSS) /Request Service (RQS) — このビットがセットされたということは、1つまたはそれ以上のネーブルされた状態が生じ、その結果、サービス要求 (SRQ) が発生したということです。なお、IEEE-488.2規格ではビット6をRQSビットと指定しており、その読取りにはシリアルポールを行ないません。

ビット7、Operation Summary Bit (OSB) — このビットがセットされたということは、イネーブルされたオペレーションイベントが発生したということです。イベントは、下記のSCPIコマンドでオペレーションイベントステータスレジスタを読み取ることによって識別できます。

:STATus:OPERation?

詳細は5.10.6項を参照してください。

プログラミング例

```

10  OUTPUT 707;“*SRE?”      ! Request contents of register
20  ENTER 707;A$           ! Address 7001 to talk
30  PRINT A$               ! Display value of register
40  END
    
```

ビット位置	B7	B6	B5	B4	B3	B2	B1	B0
イベント	OSB	MSS, RQS	ESB	MAV	QSB	EAV	—	—
10進重み	128 (2 ⁷)	64 (2 ⁶)	32 (2 ⁵)	16 (2 ⁴)	8 (2 ³)	4 (2 ²)	— (2 ¹)	— (2 ⁰)
値	0/1	0/1	0/1	0/1	0/1	0/1	—	—

値 : 1 = イベントビットがセットされた。
0 = イベントビットがクリアされた。

イベント : OSB = オペレーションサマリビット
MSS = マスタサマリビット
RQS = サービス要求
ESB = イベントサマリビット
MAV = メッセージアベイラブル
QSB = 被疑サマリビット
EAV = エラーアベイラブル

図5-16
ステータスバイトレジスタ

5.9.15 *TRG — trigger

目的	Model 7001へのGPIBトリガを発生させる。
フォーマット	*TRG
説明	<p>*TRGコマンドは、Model 7001へのGPIBトリガを発生させるのに使い、グループ実行トリガ (GET) と同じ効果をもっています。</p> <p>*TRGコマンドは、スキャンサイクルをコントロールするためのアームイベント、スキャンイベントおよび/またはチャンネルイベントとして使われます。Model 7001がこのトリガに応答するのは、プログラムされたコントロールソースがGPIBである場合です。コントロールソースは、SCPI: TRIGgerサブシステム (5.10.8項参照) からプログラムします。</p>
プログラミング例	<pre>OUTPUT 707; "*TRG" ! Trigger 7001</pre>

5.9.16 *TST? — self-test query

目的 セルフテストを実行し、結果を得る。

フォーマット *TST?

説明 *TST?問合せコマンドは、ROMのチェックサムテストを実行し、コード化された結果（0または1）を出力待ち行列に入れるのに使います。Model 7001がトークにアドレッシングされると、コード化された結果が出力待ち行列からコンピュータへ送られます。

返された値が0であれば、テストに通ったということであり、返された値が1であれば、テストに通らなかったということです。

プログラミング例

10	OUTPUT 707;“*TST?”	! Run and request test result
20	ENTER 707;AS	! Address 7001 to talk
30	PRINT AS	! Display coded result

5.9.17 *WAI — Wait-to-continue

目的 前の全コマンドが実行完了するまで、それ以降のコマンド実行を阻止する。

フォーマット *WAI

説明 *WAIコマンドは、前の全コマンドが実行完了するまで、それ以降のコマンドが実行されないようにするのに使います。プログラム中の前のコマンドがすべて実行された後、*WAIコマンドに続くコマンドが実行可能になります。

5.10 SCPIコマンドサブシステム

SCPIコマンドはいくつかのサブシステムに類別されます。以下各項に、サブシステムをアルファベット順に取り上げ、説明します。

- 5.10.1 **:DISPlayサブシステム** — ユーザ定義テキストメッセージを制御するためのコマンドについて説明します。
- 5.10.2 **:OUTPutサブシステム** — デジタル出力ポートビットの極性（アクティブハイまたはアクティブロー）をプログラムするためのコマンドについて説明します。
- 5.10.3 **:ROUteサブシステム** — チャンネルを開閉し、スキャンリストを定義し、制限（禁止）チャンネルを指定し、シングルチャンネルおよびカードペアをイネーブル/ディスエーブルするためのコマンドについて説明します。また、各スロットのリレーセットリング時間、ポールモードおよびカードタイプをプログラムするためのスロットコンフィギュレーションコマンドについても説明します。さらに、チャンネルパターンをストア（セーブ）するためのメモリコマンドについても説明します。
- 5.10.4 **:SENSeサブシステム** — Model 7001のデジタル入力ポートを読み取るためのコマンドについて説明します。
- 5.10.5 **:SOURceサブシステム** — デジタル出力ポートビットの状態（0または1）をセットするためのコマンドについて説明します。
- 5.10.6 **:STATusサブシステム** — Model 7001のステータスレジスタをコントロールするためのコマンドについて説明します。
- 5.10.7 **:SYSTemサブシステム** — Model 7001のソフトウェア改訂水準を読み取るための雑コマンドについて説明します。
- 5.10.8 **トリガサブシステム** — スキャンに使うコマンドについて説明します。

5.10.1 : DISPlayサブシステム

ディスプレイサブシステムは、Model 7001のディスプレイをコントロールするもので、表5-8に示す通りです。

表5-8

DISPlayコマンドサマリ

コマンド	説明
:DISPlay	
[:WINDow[1]]	メッセージをディスプレイの上部に位置指定するためのパス：
:TEXT	ユーザテキストメッセージを制御するためのパス：
:DATA<a>	最大20文字を使ってASCIIメッセージ“a”を定義する。
:STATe	メッセージをイネーブル（1またはON）あるいはディスエーブル（0またはOFF）する。
:STATe?	メッセージステータス（イネーブル状態またはディスエーブル状態）を要求する。
:WINDow2	メッセージをディスプレイの下部に位置指定するためのパス：
:TEXT	ユーザテキストメッセージを制御するためのパス：
:DATA<a>	最大32文字を使ってASCIIメッセージ“a”を定義する。
:STATe	メッセージをイネーブル（1またはON）あるいはディスエーブル（0またはOFF）する。
:STATe?	メッセージステータス（イネーブル状態またはディスエーブル状態）を要求する。
:SMESsage	表示されたステータスメッセージをイネーブル（1またはON）あるいはディスエーブル（0またはOFF）する。
:SMESsage?	ステータスメッセージディスプレイの状態を要求す

注：大文字はコマンド短縮形を表わす。たとえば、“display:text:state on”を送信する代わりに、“disp:text:stat on”を送信することができる。

[: WINDow [1]]

:DISPlay[:WINDow[1]]

このコマンドパスは、ディスプレイの上部をコントロールするためのものです。このコマンドを囲んでいる角括弧“[]”は、黙示的なもので、無視することができます。

:TEXT

:DISPlay[:WINDow[1]]:TEXT

このコマンドパスは、Model 7001ディスプレイの上部に表示されるユーザテキストメッセージを制御するためのものです。Model 7001では、このパスで以下のコマンドを使います。

:DATA <a>

:DISPlay[:WINDow[1]]TEXT:DATA<a>

パラメータ

a = メッセージ用のASCII文字

フォーマット

ストリング	:disp:text:data	'aa...a'
不確定ブロック	:disp:text:data	#0aa...a
確定ブロック	:disp:text:data	#XYaa...a

ここで、 Y = メッセージ中の文字数 (1~20)、
X = Yを構成している桁数 (1または2)。

デフォルト

電源投入	メッセージをキャンセル
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

説明

このコマンドは、表示させたいテキストメッセージを定義します。メッセージの長さは最大20文字まで可能です。スペースは1文字と数えられます。過剰のメッセージ文字 (20文字を超えるもの) は無視されます。

不確定ブロックメッセージは、プログラム行における唯一のコマンドであるか、またはプログラム行における最後のコマンドでなければなりません。不確定ブロックメッセージの後に (同じ行に) コマンドを含めた場合、そのコマンドはメッセージの一部として扱われ、実行されずに表示されます。

**:STATe **

:DISPlay[:WINDow[1]]TEXT:STATe

パラメータ

b = 0またはOFF	テキストメッセージをディスエーブルする。
b = 1またはON	テキストメッセージをイネーブルする。

フォーマット

:disp:text:stat b

デフォルト

電源投入	メッセージをディスエーブル
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

説明

このコマンドは、“data”を使って前回定義されたテキストメッセージをイネーブルまたはディスエーブルします。イネーブルされると、メッセージはModel 7001ディスプレイの上部に表示されます。ディスエーブルされると、メッセージはキャンセルされます。

テキストメッセージは、定義されたメッセージがなくてもイネーブルすることができます（前のコマンドを参照してください）。イネーブルされると、メッセージは定義され次第表示されます。

ユーザ定義テキストメッセージは、Model 7001がリモート状態にある間に限って表示されています。Model 7001をリモート状態から抜け出させると（LOCALキーを押すか、またはLOCAL 707を送信することによって）、テキストメッセージはディスエーブルされます。

:STATe?

:DISPlay[:WINDow[1]]:TEXT:STATe?

フォーマット

:disp:text:stat?

説明

この問合せコマンドは、ディスプレイの上部のメッセージ表示状態を確認するのに使います。このコマンドを送信した後、Model 7001をトークにアドレッシングすると、ステータスを表わす値が送り返されます。

“0”はメッセージ表示状態がディスエーブルの状態であることを表わし、“1”はメッセージ表示状態がイネーブルの状態であることを表わします。

:WINDow2

:DISPlay:WINDow2

このコマンドパスは、ディスプレイの下部をコントロールするためのものです。このパスで使うパスおよびコマンドは以下の通りです。

:TEXT

:DISPlay:WINDow2:TEXT

このコマンドパスは、Model 7001ディスプレイの下部に表示されるユーザテキストメッセージを制御するためのものです。Model 7001では、このパスで以下のコマンドを使います。

:DATA <a>

:DISPlay:WINDow2:TEXT:DATA<a>

パラメータ

a = メッセージ用のASCII文字

フォーマット

ストリング	disp:wind2:text:data	'aa...a'
不確定ブロック	disp:wind2:text:data	#0aa...a
確定ブロック	disp:wind2:text:data	#XYaa...a

ここで、Y = メッセージ中の文字数（1～32）、
X = Yを構成している桁数（1または2）。

デフォルト	電源投入	メッセージをキャンセル
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	無効

説明 このコマンドは、表示させたいテキストメッセージを定義します。メッセージの長さは最大32文字まで可能です。スペースは1文字と数えられます。過剰のメッセージ文字（32文字を超えるもの）は無視されます。

不確定ブロックメッセージは、プログラム行における唯一のコマンドであるか、またはプログラム行における最後のコマンドでなければなりません。不確定ブロックメッセージの後に（同じ行に）コマンドを含めた場合、そのコマンドはメッセージの一部として扱われ、実行されずに表示されます。

:STATe

:DISPlay:WINDow2:TEXT:STATe

パラメータ	b = 0またはOFF	テキストメッセージをディスエーブルする。
	b = 1またはON	テキストメッセージをイネーブルする。

フォーマット :disp:wind2:text:stat b

デフォルト	電源投入	メッセージをキャンセル
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	無効

説明 このコマンドは、“data”を使って前回定義されたテキストメッセージをイネーブルまたはディスエーブルします。イネーブルされると、メッセージはModel 7001ディスプレイの下部に表示されます。ディスエーブルされると、メッセージはキャンセルされます。

テキストメッセージは、定義されたメッセージがなくてもイネーブルすることができます（前のコマンドを参照してください）。イネーブルされると、メッセージは定義され次第表示されます。

ユーザ定義テキストメッセージは、Model 7001がリモート状態にある間に限って表示されています。Model 7001をリモート状態から抜け出させると（LOCALキーを押すか、またはLOCAL 707を送信することによって）、テキストメッセージはディスエーブルされます。

:STATe?

:DISPlay:WINDow2:TEXT:STATe?

フォーマット :disp:wind2:text:stat?

説明 この問合せコマンドは、ディスプレイの下部のメッセージ表示状態を確認するのに使います。このコマンドを送信した後、Model 7001 をトークにアドレッシングすると、ステータスを表わす値が送り返されます。

“0” はメッセージ表示状態がディスエーブルの状態であることを表わし、“1” はメッセージ表示状態がイネーブルの状態であることを表わします。

:SMESsage

:DISPlay:SMESsage

パラメータ

b = 0またはOFF	ステータスメッセージをディスエーブルする。
b = 1またはON	ステータスメッセージをイネーブルする。

フォーマット :disp:smes b

デフォルト

電源投入	セーブされているデフォルトを選択
*RST	ステータスメッセージをディスエーブル
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	ステータスメッセージをディスエーブル
:STATus:PRESet	無効

説明 このコマンドは、Model 7001のステータスメッセージモードをイネーブル/ディスエーブルするのに使います。ステータスメッセージモードは、Model 7001の現在の動作状態に関するリアルタイムのメッセージを与えてくれる診断ツールです。

:SMESsage?

:DISPlay:SEMSsage?

フォーマット :disp:smes?

説明 この問合せコマンドは、ステータスメッセージディスプレイの状態を要求するのに使います。このコマンドを送信した、Model 7001 をトークにアドレッシングした後、ステータスメッセージ (0または1) がコンピュータへ送られます。

プログラミング例 以下のプログラミング例は、DISPlayサブシステムのコマンドを使ってユーザ定義メッセージを表示させるものです。

```

10 OUTPUT 707;":disp:text:data 'Keithley Model 7001';stat on;stat?"
20 ENTER 707;AS
30 PRINT "Top message display status=";A$
40 OUTPUT 707;":disp:wind2:text:data 'Switch System';stat on;stat?"
50 ENTER 707;BS
60 PRINT "Bottom message display status=";B$
70 END

```

- 行10 この行の3つのコマンドは上部ディスプレイをコントロールするものである。DATAはメッセージを定義し、STATEはメッセージを表示させ、STATE?はメッセージディスプレイの状態を要求する。
- 行20 7001をトークにアドレッシングする。
- 行30 メッセージディスプレイの状態 (1;イネーブル状態) を表示させる (CRT上に)。
- 行40 この行の3つのコマンドは下部ディスプレイをコントロールするものである。DATAはメッセージを定義し、STATEはメッセージを表示させ、STATE?はメッセージディスプレイの状態を要求する。
- 行50 7001を トークにアドレッシングする。
- 行60 メッセージディスプレイの状態 (1;イネーブル状態) を表示させる (CRT上に)。

5.10.2 : OUTPutサブシステム

OUTPutサブシステムは、デジタル出力ポートの極性を設定するのに使います。このサブシステムに含まれるコマンドを表5-9に示します。

表5-9

OUTPutコマンドサマリ

コマンド	説明
:OUTPut :TTLn :LSENse AHIGH ALOW :LSENse?	出力ライン “n” (1~4) を指定するためのパスおよびコマンド： 極性、すなわち、アクティブハイ (AHIGH) またはアクティブロー (ALOW) を設定する。 指定された出力ライン (n) を要求する。

注: 大文字はコマンド短縮形を表わす。たとえば、“:output:t1:lense ahigh”を送信する代わりに、“:out:t1:lсен ahig”を送信することができる。

:TTLn

:OUTPut:TTLn

パラメータ

n=1	出力ライン1 (ビット1)
n=2	出力ライン2 (ビット2)
n=3	出力ライン3 (ビット3)
n=4	出力ライン4 (ビット4)

説明

このコマンドパスは、指定されたデジタル出力ライン (n) の極性を設定する (かつ問合せ) ためのものです。このパスで使うコマンドは以下通りです。

LSENse AHIGH | ALOW

:OUTPut:TTLn:LSENse AHIGH|ALOW

パラメータ

AHIGH 指定された出力をアクティブハイに設定する。
ALOW 指定された出力をアクティブローに設定する。

フォーマット

:out:t1n:lсен ahig
:out:t1n:lсен alow

デフォルト

電源投入	アクティブハイ極性に対応する全ビットをセット
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTEM:PRESet	無効
:STATus:PRESet	無効

説明

このコマンドは、指定された出力ライン (n) の極性を設定します。出力ライン (ビット) をアクティブハイ (ahig) にプログラムした場合、“1またはON”出力レベルは+5V、“0またはOFF”出力レベルは0Vです。逆に、アクティブロー (alow) にプログラムした場合は、“1またはON”出力レベルは0V、“0またはOFF”出力レベルは+5Vです。

出力ポートのビットは、SOURCEサブシステム（5.10.5項参照）を使ってプログラムします（“0またはOFF”あるいは“1またはON”）。

:LSENse ?

:OUTPut:TTLn:LSENse?

フォーマット

outp:ttl:n:lsen?

説明

この問合せコマンドは、指定されたデジタル出力ライン（n）の極性を要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングした後、極性を定義するメッセージがコンピュータへ送られます。メッセージ“AHIG”はアクティブハイ極性を表わし、メッセージ“ALOW”はアクティブロー極性を表わします。

プログラミング例

下例は、OUTPutサブシステムのコマンドの使い方を示すものです。

```
10  OUTPUT 707;":outp:ttl1:lsen alow;lsen?
20  ENTER 707;A$
30  PRINT AS
40  END
```

行10 この行の3つのコマンドのうち、TTLは出力ライン1を指定し、LSENseは極性をアクティブローに設定し、LSENse?は極性を要求する。

行20 7001をトークにアドレッシングする。

行30 極性メッセージをCRT上に表示させる。

5.10.3 [:ROUTe]サブシステム

ROUTeサブシステムは、スイッチシステム内のシグナルルーティングをコントロールするためのもので、表5-10に示す通りです。角括弧は、このコマンドがオプションのコマンドで、コマンドメッセージには含める必要がないことを表わします。

表5-10

ROUTeコマンドサマリ

コマンド	説明
[:ROUTe]	
:CLOSe<list> :STATe?	指定されたチャンネルを閉じるためのパスおよびコマンド： 閉じているチャンネルを要求する。
:OPEN<list> ALL	指定された（または全部の）チャンネルを開く。 チャンネルリストを指定するためのパスおよびコマンド：
:SCAN<list> :POINts?	スキャンリスト中のチャンネルの数を要求する。
:SCAN?	プログラムされたスキャンリストを要求する。
:FCHannels<list>	閉じてはならないチャンネル（禁止チャンネル）を指定する。
:FCHannels?	制限チャンネルのリストを要求する。
:CONFigure:	コンフィギュレーションコマンドパス：
:SCHannel	Single Channelをイネーブル(1またはON)あるいはディスエーブル(0またはOFF)する。
:SCHannel?	Single Channelモードのステータスを要求する。
:CPAir	Channel Pairingをイネーブル(1またはON)あるいはディスエーブル(0またはOFF)する。
:CPAir?	Channel Paringのステータスを要求する。
:SLOT1	CARD 1を設定するためのパス：
:CTYPE<type>	スロットに装着されたカードタイプを指定する。
:CTYPE?	スロットに装着されたカードタイプを要求する。
:POLE<n>	ポールモード（1、2または4）を選択する。
:POLE?	プログラムされたポール数を要求する。
:STIME<num>	秒単位のリレーセットリング時間を（0～99999.999）を設定する。
:STIME?	プログラムされたリレーセットリング時間を要求する。
:SLOT2	CARD 2を設定するためのパス：
:CTYPE<type>	スロットに装着されたカードタイプを指定する。
:CTYPE?	スロットに装着されたカードタイプを要求する。
:POLE<n>	ポールモード（1、2または4）を選択する。
:POLE?	プログラムされたポール数を要求する。
:STIME<num>	秒単位のリレーセットリング時間を（0～99999.999）を設定する。
:STIME?	プログラムされたリレーセットリング時間を要求する。
:MEMory	メモリをプログラムするためのパス：
:SAVe M<num>	チャンネルパターンをメモリロケーション（1～100）にセーブする。
:RECall M<num>	メモリロケーション（1～100）からチャンネルパターンをリコールする。

注： 大文字はコマンド短縮形を表わす。たとえば、“:configure:schannel?”を送信する代わりに、“:conf:sch?”を送信することができる。

:CLOSe <list>

[:ROUte]:CLOSe<list>

パラメータ

list = (@ chanlist)

ここで、chanlistは、指定された閉じられるべきチャンネルリストのリスト（チャンネルリスト）。

フォーマット

:clos (@ chanlist)

デフォルト

電源投入	全チャンネル開
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

説明

closeはコマンドパスの役割をし、<list>パラメータが含まれている場合に、指定されたチャンネルを閉じるのに使います。chanlist（チャンネルリスト）は、閉じるべきチャンネル（1つまたは複数）を指定します。下例は、chanlistへのチャンネル入力を表現するための各種の形式を示すものです。

Chanlist

= 1!2!4	シングルチャンネル（スロット1、行2、列4）。
= 2!36	シングルチャンネル（スロット2、チャンネル36）。
= 1!6, 1!18, 2!3!6	カンマで区切られた複数のチャンネル。
= 1!2:1!10	チャンネルのレンジ（2～10）。
= 1!2:1!10, 2!3!6	カンマで区切られたレンジ入力とシングルチャンネル入力。
= 1!30:1!40, 2!1:2!10	マルチプレックスチャンネルのレンジ（Card 1の最後の10チャンネル、Card 2の最初の10チャンネル）。
= 1!4!1:1!4!10, 2!1!1:2!4!10	マトリックスチャンネルのレンジ（Card 1の行4の10列、Card 2の行1の10列）。
= M36	メモリロケーションM36にストアされたチャンネルパターン。

注：このコマンドをパスとしてのみ使う場合は、<list>パラメータは使いません。

:STATe ?

[:ROUte]:CLOSe:STATe?

フォーマット

:clos:stat?

説明

この問合せコマンドは、現在閉じているチャンネルを要求するのに使います。たとえば、チャンネル1!2!3および2!36が閉じている場合、このコマンドを送信し、Model 7001をトークにアドレッシングした後、下記のメッセージがコンピュータへ送られます。

(@1!2!3, 2!36)

プログラミング例

以下のプログラミング例は、ROUTeサブシステムのCLOSEおよびSTATe? コマンドの使い方を示すものです。

```
10 OUTPUT 707;":clos (@ 1!1, 1!5:1!10);clos:stat?"
20 ENTER 707;A$
30 PRINT A$
40 END
```

行10 この行の2つのコマンドのうち、CLOSEはリスト中のチャンネルを閉じ、STATe? は閉じているチャンネルのリストを要求する。
 行20 Model 7001をトークにアドレッシングする。
 行30 閉じているチャンネルのリストを表示させる (CRT上に)。

:OPEN <list> | ALL

```
[:ROUTe]:OPEN<list>|ALL
```

パラメータ

list = (@ chanlist)

ALLは全チャンネルを開くのに使う。

ここで、chanlistは、指定された、開くべきチャンネルのリスト(チャンネルリスト)。

フォーマット

```
:open (@ chanlist)
:open all
```

デフォルト

電源投入	全チャンネル開
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

説明

このコマンドは、パラメータによって指定されたチャンネルを開くのに使います。<list>パラメータはchanlist (チャンネルリスト) を指定するのに使い、ALLパラメータは全チャンネルを開くのに使います。

下例は、chanlistへのチャンネル入力を表現するための各種の形式を示すものです。

Chanlist

= 1!2!4	シングルチャンネル (スロット1、行2、列4)。
= 2!36	シングルチャンネル (スロット2、チャンネル36)。
= 1!6, 1!18, 2!3!6	カンマで区切られた複数のチャンネル。
= 1!2:1!10	チャンネルのレンジ (2~10)。
= 1!2:1!10, 2!3!6	カンマで区切られたレンジ入力とシングルチャンネル入力。
= 1!30:1!40, 2!1:2!10	マルチプレックスチャンネルのレンジ(Card 1の最後の10チャンネル、Card 2の最初の10チャンネル)。
= 1!4!1:1!4!10, 2!1!1:2!4!10	マトリックスチャンネルのレンジ(Card 1の行4の10列、Card 2の行1の10列)。
= M36	メモリロケーションM36にストアされたチャンネルパターン。

プログラミング例

以下のプログラミング例は、ROUTEサブシステムのOPENコマンドの使い方を示すものです。

```
10 OUTPUT 707;":clos (@ 1!1:1!10)"
20 OUTPUT 707;":open (@ 1!1:1!6)"
30 WAIT 3
40 OUTPUT 707;":open all"
50 END
```

行10 スロット1のチャンネル1~10を閉じる (CLOSEコマンド参照)。
 行20 チャンネル1~6を閉じる。
 行30 3秒のディレイ。
 行40 3秒待った後、残りのチャンネル (7~10) が開く。

:SCAN <list>

```
[:ROUTE]:SCAN<list>
```

パラメータ

```
list = (@ scanlist)
```

ここで、scanlistは、指定された、スキャンすべきチャンネルのリスト。

フォーマット

```
:scan (@ scanlist)
```

デフォルト

電源投入	前回定義されたスキャンリスト
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTEM:PRESet	無効
:STATus:PRESet	無効

説明

scanはコマンドパスとして使います。また、パラメータを付けて使う場合は、スキャンリストを定義するためのコマンドとして使います。scanlistは、スキャンすべきチャンネル (1つまたは複数) を定義します。下例は、scanlistを表現するための各種の形式を示すものです。

Scanlist	
= 1!6, 1!18, 2!3!6	カンマで区切られたチャンネル。
= 1!2:1!10	チャンネルのレンジ (2~10)。
= 1!10:1!2	チャンネルのレンジ (10~2)。
= 1!2:1!10, 1!20	カンマで区切られたレンジ入力とシングルチャンネル入力。
= 1!30:1!40, 2!1:2!10	マルチプレックスチャンネルのレンジ (Card 1の最後の10チャンネル、Card 2の最初の10チャンネル)。
= 1!4!1:1!4!10, 2!1!1:2!4!10	マトリックスチャンネルのレンジ (Card 1の行4の10列、Card 2の行1の10列)。
= M36	メモリロケーションM36にストアされたチャンネルパターン。

注： このコマンドをパスとしてのみ使う場合は、パラメータは使いません。

:POINTs ?

[:ROUTe]:SCAN:POINTs?

フォーマット

:scan:poin?

説明

この問合せコマンドは、スキャンリスト中のチャンネルの数を要求するのに使います。スキャンリスト中のチャンネルパターン（すなわち、M1）は1チャンネルと数えられます。たとえば、次のようなチャンネルリストを考えてみます。

(@ 1!1:1!5, 1!10, M2)

poin?コマンドが送信され、Model 7001がトークにアドレッシングされた後、下記のメッセージがコンピュータへ送られます。

7

“7” は、チャンネルリスト中に7つのチャンネルがあることを表わします。

プログラミング例

以下のプログラミング例は、ROUTeサブシステムのSCANおよびPOINTs ? コマンドの使い方を示すものです。

```
10  OUTPUT707;“:scan (@ 1!2, 1!4);scan:poin?”
20  ENTER707;AS
30  DISP AS
40  END
```

行10 この行の2つのコマンドのうち、SCANはスキャンリストを作成し、POINTs ? はスキャンリスト中のチャンネルの数を要求する。

行20 Model 7001をトークにアドレッシングする。

行30 スキャンリスト中のチャンネルの数 (2) を表示させる (CRT上に)。

:SCAN ?

[:ROUTe]:SCAN?

フォーマット

:scan?

説明

この問合せコマンドは、現在定義されているスキャンリストを要求するのに使います。このコマンドを送信した後、Model 7001がトークにアドレッシングされると、スキャンリストがコンピュータへ送られます。返されたスキャンリストの一例を示すと、次の通りです。

(@ 1!1:1!5, 1!8)

このスキャンリストは、チャンネル1、2、3、4、5、8（全チャンネルともスロット1）で構成されています。スキャンリストが存在しない（クリアされている）場合、SCAN? 問合せコマンドは下記のメッセージを送信します。

(@)

プログラミング例

以下のプログラミング例は、ROUTeサブシステムのSCAN? コマンドの使い方を示すものです。

例 #1 :

```
10 OUTPUT 707;“.scan?”
20 ENTER 707;A$
30 PRINT AS
40 END
```

行10 スキャンリストを要求する。
 行20 7001をトークにアドレッシングする。
 行30 スキャンリストをCRT上に表示させる。

例 #2 :

```
10 OUTPUT 707;“.scan (@ 1!1:1!5, 1!8);scan?”
20 ENTER 707;A$
30 PRINT AS
40 END
```

行10 この行の2つのコマンドのうち、SCANはスキャンリストを定義し、SCAN? はスキャンリストを要求する。
 行20 7001をトークにアドレッシングする。
 行30 スキャンリストをCRT上に表示させる。

:FCHannels <list>

[:ROUTe]:FCHannels <list>

パラメータ

list = (@ forblist)

ここで、forblistは、指定された、閉じてはならない“禁止”チャンネルのリスト。

フォーマット

:fch (@ forblist)

デフォルト

電源投入	前回定義された禁止チャンネルリスト
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

説明

このコマンドは、そのパラメータとともに、閉じてはならないチャンネルのリストを定義するのに使います。下例は、forblistを表現するための各種の形式を示すものです。

Forblist = 1!1	シングルチャンネル (スロット1、チャンネル1)。
= 1!6, 1!18, 2!3!6	カンマで区切られたチャンネル。
= 1!2:1!10	チャンネルのレンジ (2~10)
= 1!2:1!10, 1!20	カンマで区切られたレンジ入力とシングルチャンネル入力。
= M36	メモリロケーションM36にストアされたチャンネルパターン。

“禁止”チャンネルリストは、次のように空のforblistを送信することによってキャンセルすることができます。

```
:fch (@)
```

プログラミング例

以下のプログラミングステートメントは、チャンネル1!1および1!4で構成される“禁止”チャンネルリストを定義するものです。

```
OUTPUT 707;“:fch (@ 1!1, 1!4)”
```

:FCHannels?

```
[:ROUTe]:FCHannels?
```

フォーマット

```
:fch?
```

説明

この問合せコマンドは、“禁止”チャンネルのリストを要求するのに使います。これらの制限チャンネルは閉じることはできません。このコマンドを送信した後、Model 7001をトークにアドレッシングすると、制限チャンネルリストが送り返されます。たとえば、チャンネル1!1および1!4が制限されている場合、下記のメッセージがコンピュータへ送られます。

```
(@ 1!1, 1!4)
```

現在定義されている制限チャンネルがなければ、下記のメッセージが繰り返されます。

```
(@)
```

プログラミング例

以下のプログラミング例は、ROUTeサブシステムのFCHannelsコマンドの使い方を示すものです。

例 #1 :

```
10 OUTPUT 707;“:fch?”
20 ENTER 707;AS
30 PRINT AS
40 END
```

行10 “禁止”チャンネルリストを要求する。
 行20 7001をトークにアドレッシングする。
 行30 制限チャンネルリストをCRT上に表示させる。

例 #2 :

```
10 OUTPUT 707;“:fch (@ 1!1, 1!4);fch?”
20 ENTER 707;AS
30 PRINT AS
40 END
```

行10 この行のコマンドのうち、FCHannelsは制限チャンネルリストを作成し、FCHannels?はリストを要求する。
 行20 7001をトークにアドレッシングする。
 行30 制限チャンネルリストをCRT上に表示させる。

:CONFigure

[:ROUte]:CONFigure

このコマンドパスは、Model 7001のスロットを設定するのに使います。また、Model 7001のCard Paring機能およびシングルチャンネルモードをコントロールするのにも使います。

:SCHannel

[:ROUte]:CONFigure:SCHannels

パラメータ	b = 0またはOFF	シングルチャンネルモードをディスエーブルする。
	b = 1またはON	シングルチャンネルモードをイネーブルする。
フォーマット	:conf:sch b	
デフォルト	電源投入	セーブされているデフォルトを選択 (*SAV参照)
	*RST	シングルチャンネルをディスエーブル
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	シングルチャンネルをディスエーブル
	:STATus:PRESet	無効

説明

シングルチャンネルモード（イネーブルされているとき）は、複数のチャンネルが同時に閉じるのを防止するのに使います。シングルチャンネルが閉じると、前の閉チャンネルが開きます。これによれば、チャンネルをランダムに閉じる場合でも、一度に1つのチャンネルしか閉じません。シングルチャンネルモードがイネーブルされると、全チャンネルが開きます。シングルチャンネルの詳細は第4節を参照してください。

:SCHannel?

[:ROUte]:CONFigure:SCHannels?

フォーマット :conf:sch?**説明**

この問合せコマンドは、シングルチャンネルモードのステータスを要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングした後、値（0または1）がコンピュータへ送られます。

“0”はシングルチャンネルがディスエーブルされていることを表わし、“1”はシングルチャンネルがイネーブルされていることを表わします。

:CPAir

[:ROUte]:CONFigure:CPair

パラメータ	b = 0またはOFF	Card Pairをディスエーブルする。
	b = 1またはON	Card Pairをイネーブルする。

フォーマット	:conf:cpa b	
デフォルト	電源投入	セーブされているデフォルトを選択 (*SAV参照)
	*RST	Card Pairをディスエーブル
	*CLS	無効
	DCL、SDC	無効
	:SYSTEM:PRESet	Card Pairをディスエーブル
	:STATus:PRESet	無効

説明

Card Pairがイネーブルされると、メインフレーム内に装着されている2枚のカードの対応するチャンネルが同時にコントロールされます。カードは両方ともスロット1に指定されることになります。たとえば、CARD 1のチャンネル10を閉じると、CARD 2のチャンネル10も閉じます。

Card Pairは、同タイプのカードに対してしかイネーブルできません。たとえば、一方のスロットにマルチプレクサカード、他方のスロットにマトリックスカードが装着されている場合、Card Pairをイネーブルすることはできません。

:CPAir?

```
[:ROUTE]:CONFigure:CPAir?
```

フォーマット :conf:cpa?

説明

この問合せコマンドは、Card Pair機能のステータスを要求するのに使います。このコマンドを送信した後、Model 7001がトークにアドレッシングされると、ステータス (0または1) がコンピュータへ送られます。“0”はCard Pairがディスエーブルされていることを表わし、“1”はCard Pairがイネーブルされていることを表わします。

:SLOT1

```
[:ROUTE]:CONFigure:SLOT1
```

このパスは、スロット1を正しく動作するように設定するのに使います。このパスで使うコマンドは以下の通りです。

:CTYPE <type>

```
[:ROUTE]:CONFigure:SLOT1:CTYPE<type>
```

パラメータ

type = C7052	type = C7062	type = C7154	type = C9991
type = C7053	type = C7063	type = C7156	type = C9990
type = C7054	type = C7064	type = C7158	
type = C7056	type = C7065	type = C7164	
type = C7057	type = C7066	type = C7166	
type = C7058	type = C7067	type = C7168	
type = C7059	type = C7152	type = C7169	
type = C7061	type = C7153	type = C7402	

フォーマット :conf:slot1:ctype:type

デフォルト

電源投入	701Xを検出または前回プログラムされたタイプを使用
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

説明 スロット1にModel 701Xシリーズのカード（すなわち、Model 7011、7012または7013）が装着されていない場合、このコマンドを使って適切な型番をスロット1に指定します。たとえば、スロット1にModel 7156カードが装着されていれば、typeパラメータC7156を使います。

C9990およびC9991はカードシミュレータです。C9990をスロット1に指定することにより、Model 7001は40チャンネル・スイッチカードが装着されているかのように動作します。C9991を指定することにより、Model 7001は4×10マトリックスカードが装着されているかのように動作します。

:CTYPE?

[[:ROUte]:CONFigure:SLOT1:CTYPE?

フォーマット :conf:slot1:ctype?

説明 この問合せコマンドは、スロット1に対する型番指定を要求するに使用します。このコマンドを送信した後、Model 7001がトークにアドレッシングされると、型番がコンピュータへ送られます。たとえば、スロット1にModel 7156スキャナカードが指定されている場合、下記のメッセージが送られます。

7156

:POLE <n>

[[:ROUte]:CONFigure:SLOT1:POLE<n>

パラメータ

n=1	1ポールモードを選択
n=2	2ポールモードを選択
n=4	4ポールモードを選択

フォーマット :conf:slot1:pole n

デフォルト

電源投入	前回選択されたポールモード
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

説明

カードの中には、2つ以上のスイッチタイプとして動作できるものがあります。たとえば、Model 7011マルチプレクサカードは、40チャンネル、2ポールスイッチングにも使えますし、20チャンネル、4ポールスイッチングにも使えます。

付加ポールモード機能をもつスイッチカードを示すと、次の通りです。

ポールモード			
カード	1ポール	2ポール	4ポール
7011	使用不可	40チャンネル	20チャンネル
7056	20チャンネル	10チャンネル	使用不可
7156	20チャンネル	10チャンネル	使用不可

選択したポールモードに応じてカードを適切に配線しなければなりませんから、注意してください。ポールモードの詳細は第4節を参照してください。

出荷時デフォルトは全カードとも2ポールモードです。このコマンドは、スロット1に装着されたカードの代替ポールモードを選択するのに使います。無効のポールモードは選択できません。無効のポールモードを選択しようとすると、エラーメッセージが表示されます。

:POLE?

```
[:ROUTE]:CONFigure:SLOT1:POLE?
```

フォーマット

```
:conf:slot1:pole?
```

説明

この問合せコマンドは、スロット1に装着されたカードに対して現在選択されているポールモードを要求するのに使います。このコマンドを送信した後、Model 7001がトークにアドレッシングされると、ポールモード（1、2または4）がコンピュータへ送られます。

“1”は1ポールモードが選択されていること、“2”は2ポールモードが選択されていること、“4”は4ポールモードが選択されていることをそれぞれ表わします。

:STIME <num>

```
[:ROUTE]:CONFigure:SLOT1:STIME<num>
```

パラメータ

num = CARD 1の秒単位のセトリング時間 (0~99999.999秒)

フォーマット

```
:conf:slot1:stim num
```

デフォルト

電源投入	デイレイ = 0 sec
*RST	デイレイ = 0 sec
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	デイレイ = 0 sec
:STATus:PRESet	無効

説明

内部的に設定されたリレーセットリング時間は、リレーが閉じるときのスイッチバウンスを考慮して、そのぶん余裕が見込まれています。電源を投入すると、取得された識別情報に基づいてセットリング時間が自動的に設定されます。このコマンドでは、スロット1に装着されたカード (CARD 1) のダイレイを大きくすることができます。

ダイレイ値は指数形で入力することができます。たとえば、2000 (秒) のパラメータ値を送信する代わりに、2E3を送信することができます。このダイレイをキャンセルするには、パラメータ値を0としてこのコマンドを送信します。このダイレイの詳細は2.7項の「DELAY」を参照してください。

:STIME?

```
[:ROUTE]:CONFigure:SLOT1:STIME?
```

フォーマット

```
:conf:slot1:stim?
```

説明

この問合せコマンドは、CARD 1のセットリング時間を要求するのに使います。このコマンドを送信した後、Model 7001がトークにアドレッシングされると、セットリング時間がコンピュータへ送られます。たとえば、セットリング時間が2.25秒の場合、下記のメッセージが送られます。

```
2.250
```

:SLOT2

```
[:ROUTE]:CONFigure:SLOT2
```

このパスは、スロット2を正しく動作するように設定するためのものです。このパスで使うコマンドは以下の通りです。

:CTYPE <type>

```
[:ROUTE]:CONFigure:SLOT2:CTYPE<type>
```

パラメータ

```
type = C7052  type = C7062  type = C7154  type = C9991
type = C7053  type = C7063  type = C7156  type = C9990
type = C7054  type = C7064  type = C7158
type = C7056  type = C7065  type = C7164
type = C7057  type = C7066  type = C7166
type = C7058  type = C7067  type = C7168
type = C7059  type = C7152  type = C7169
type = C7061  type = C7153  type = C7402
```

フォーマット

```
:conf:slot2:ctype type
```

デフォルト	電源投入	701Xカードを検出または前回プログラムされたタイプを使用
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	無効

説明 スロット2にModel 701Xシリーズのカード（すなわち、Model 7011、7012または7013）が装着されていない場合、このコマンドを使って適切な型番をスロット2に指定します。たとえば、スロット2にModel 7156カードが装着されていれば、typeパラメータC7156を使います。

C9990およびC9991はカードシミュレータです。C9990をスロット2に指定することにより、Model 7001は40チャンネル・スイッチカードが装着されているかのように動作します。C9991を指定することにより、Model 7001は4×10マトリックスカードが装着されているかのように動作します。

:CTYPE?

[[:ROUTE]:]CONFigure:SLOT2:CTYPE?

フォーマット :conf:slot2:ctype?

説明 この問合せコマンドは、スロット2に対する型番指定を要求するのに使います。このコマンドを送信した後、Model 7001がトークにアドレッシングされると、型番がコンピュータへ送られます。たとえば、スロット2にModel 7156スキャナカードが指定されている場合、下記のメッセージが送られます。

C7156

:POLE <n>

[[:ROUTE]:]CONFigure:SLOT2:POLE<n>

パラメータ	n=1	1ポールモードを選択
	n=2	2ポールモードを選択
	n=4	4ポールモードを選択

フォーマット :conf:slot2:pole n

デフォルト	電源投入	前回選択されたポールモード
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	無効

説明 カードの中には、2つ以上のスイッチタイプとして動作できるものがあります。たとえば、Model 7011マルチプレクサカードは、40チャンネル、2ポールスイッチングにも使えますし、20チャンネル、4ポールスイッチングにも使えます。

付加ポールモード機能をもつスイッチカードを示すと、次の通りです。

ポールモード			
カード	1ポール	2ポール	4ポール
7011	使用不可	40チャンネル	20チャンネル
7056	20チャンネル	10チャンネル	使用不可
7156	20チャンネル	10チャンネル	使用不可

選択したポールモードに応じてカードを適切に配線しなければなりませんから、注意してください。ポールモードの詳細は第4節を参照してください。

出荷時デフォルトは全カードとも2ポールモードです。このコマンドは、スロット2に装着されたカードの代替ポールモードを選択するのに使います。無効のポールモードは選択できません。無効のポールモードを選択しようとすると、エラーメッセージが表示されます。

:POLE?

```
[:ROUTE]:CONFigure:SLOT2:POLE?
```

フォーマット

```
:conf:slot2:pole?
```

説明

この問合せコマンドは、スロット2に装着されたカードに対して現在選択されているポールモードを要求するのに使います。このコマンドを送信した後、Model 7001がトークにアドレッシングされると、ポールモード（1、2または4）がコンピュータへ送られます。

“1”は1ポールモードが選択されていること、“2”は2ポールモードが選択されていること、“4”は4ポールモードが選択されていることをそれぞれ表わします。

:STIMe <num>

```
[:ROUTE]:CONFigure:SLOT2:STIMe<num>
```

パラメータ

num = CARD 2の秒単位のセットリング時間（0～99999.999秒）

フォーマット

```
:conf:slot2:stim num
```

デフォルト

電源投入	デレイ = 0 sec
*RST	デレイ = 0 sec
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	デレイ = 0 sec
:STATus:PRESet	無効

説明

内部的に設定されたリレーセットリング時間は、リレーが閉じるときのスイッチバウンスを考慮して、そのぶんの余裕が見込まれています。電源を投入すると、取得された識別情報に基づいてセットリング時間が自動的に設定されます。このコマンドでは、スロット2に装着されたカード（CARD 2）のデレイを大きくすることができます。

ディレイ値は指数形で入力することができます。たとえば、2000 (秒) のパラメータ値を送信する代わりに、2E3を送信することができます。このディレイをキャンセルするには、パラメータ値を0としてこのコマンドを送信します。このディレイの詳細は2.7項の「DELAY」を参照してください。

:STIMe?

[:ROUTe]:CONFigure:SLOT2:STIMe?

フォーマット

:conf:slot2:stim?

説明

この問合せコマンドは、CARD 2のセットリング時間を要求するのに使います。このコマンドを送信した後、Model 7001がトークにアドレッシングされると、セットリング時間がコンピュータへ送られます。たとえば、セットリング時間が2.25秒の場合、下記のメッセージが送られます。

2.250

プログラミング例

以下のプログラミング例は、ROUTeサブシステムのCONFigureコマンドの使い方を示すものです。

```
10 OUTPUT 707;":conf:sch on;cpa off;slot1:stim 1"
20 OUTPUT 707;":conf:sch?;cpa?;slot1:stim?;pole?"
30 ENTER 707 AS
40 PRINT AS
50 END
```

- 行10 この行の3つのコマンドのうち、SCHannelはシングルチャンネルをイネーブルし、CPAirはCard-Pairをディスエーブルし、STIMeはスロット1のディレイを1秒に設定する。
- 行20 この行の4つのコマンドのうち、SCHannel?はシングルチャンネルのステータスを要求し、CPAir?はCard-Pairのステータスを要求し、POLE?はスロット1のポールモードを要求し、STIMe?はスロット1のディレイを要求する。
- 行30 7001をトークにアドレッシングする。全ての問合せデータがコンピュータへ送られる。
- 行40 問合せメッセージをCRT上に表示させる。データは問合せコマンドの送信順に表示される。たとえば、

1, 0, 1.000, 2

“1”はシングルチャンネルがイネーブルされていること、“0”はCard-Pairがディスエーブルされていること、“1.000”はディレイが1秒に設定されていること、“2”はスロット1が2ポールモードに設定されていることをそれぞれ表わします。

:MEMory

[:ROUte]:MEMory

このコマンドパスは、チャンネルパターンをメモリにセーブ（ストア）し、それらをリコールするためのものです。このパスで使うコマンドは以下の通りです。

:SAVe M<num>

[:ROUte]:MEMory:SAVe M<num>

パラメータ

num = 1~100

Mはメモリロケーション指示子。

フォーマット

:mem:sav Mnum

説明

このコマンドは、チャンネルパターンをメモリロケーションにセーブ（ストア）するのに使います。最大100チャンネルパターンをメモリロケーション1~100にセーブすることができます。

チャンネルパターンは、RECallコマンドを使ってメモリからリコールすることができます（次のコマンドを参照してください）。

デフォルト

電源投入	無効
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

:RECall M<num>

[:ROUte]:MEMory:RECall M<num>

パラメータ

num = 1~100

Mはメモリロケーション指示子。

フォーマット

:mem:rec Mnum

説明

このコマンドは、メモリロケーションにストアされているチャンネルパターンをリコールするのに使います。チャンネルパターンはメモリロケーション1~100からリコールできます。チャンネルパターンがリコールされると、直ちに、フロントパネルのステータスチャンネルディスプレイがそのチャンネルパターンに更新されます。

チャンネルパターンがストアされていないメモリロケーションからチャンネルパターンをリコールした場合は、ブランクのチャンネルパターンをリコールしたということです。ブランクのチャンネルパターンには閉チャンネルはありません。両スロットとも全チャンネルが開いています。

チャンネルパターンは、SAVeコマンドを使ってメモリにセーブすることができます（前のコマンドを参照してください）。

プログラミング例

下例は、ROUTeサブシステムのMEMoryコマンドの使い方を示すものです。

```
10  OUTPUT 707;“:clos (@ 1!1:1!10)”
20  OUTPUT 707;“:mem:sav M36”
30  OUTPUT 707;“:open all”
40  WAIT 3
50  OUTPUT 707;“:mem:rec M36”
60  END
```

行10 スロット1のチャンネル1～10を閉じる。

行20 チャンネルパターンをメモリロケーションM36にセーブする。

行30 全チャンネルを開く。

行40 3秒のディレイ。

行50 3秒後、M36のチャンネルパターンがリコールされ、チャンネル1～10が閉じる。

5.10.4 :SENSe [1] サブシステム

SENSe [1] サブシステムは、Model 7001のシングルチャンネルデジタル入力ポートを読み取るのに使います。このセンスサブシステムを表5-11に示します。

表5-11
ASENSEコマンドサマリ

コマンド	説明
:SENSe[1]	
:TTL1	入力ポートのためのコマンドバス。 デジタル入力ポートを読み取る。
:DATA?	

:TTL1

:SENSe[1]:TTL1

このコマンドバスは、下記のコマンドと共に、デジタル入力ポートを読み取るのに使います。

:DATA?

:SENSe[1]:TTL1:DATA?

フォーマット

:sens:tll1:data?

説明

この問合せコマンドは、デジタル入力ポートを読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、入力ポートの状態 (0または1) がコンピュータへ送られます。

プログラミング例

```
10 OUTPUT 707;":sens:tll1:data?"
20 ENTER 707;AS
30 PRINT AS
40 END
```

行10 入力ポートのステータスを要求する。
 行20 7001をトークにアドレッシングする。
 行30 入力ポートステータスをCRT上に表示させる。

注

Model 7001はSENSe2およびSENSe3サブシステムのコマンドも持っており、これらは将来の使用のために確保されています。

5.10.5 : SOURceサブシステム

SOURceサブシステムは、デジタル出力ポートの4チャンネルのビットをコントロールするのに使います。このソースサブシステムを表5-12に示します。

表5-12
SOURceコマンドサマリ

コマンド	説明
:SOURce	デジタル出力ポートのビットをプログラムするためのコマンドバス:
:TTL1 [:LEVel] [:LEVel]?	出力ポートのビット1をプログラムするためのバス: セット (1またはON) あるいはリセット (0またはOFF)。 ビット1のステータスを要求する。
:TTL2 [:LEVel] [:LEVel]?	出力ポートのビット2をプログラムするためのバス: セット (1またはON) あるいはリセット (0またはOFF)。 ビット2のステータスを要求する。
:TTL3 [:LEVel] [:LEVel]?	出力ポートのビット3をプログラムするためのバス: セット (1またはON) あるいはリセット (0またはOFF)。 ビット3のステータスを要求する。
:TTL4 [:LEVel] [:LEVel]?	出力ポートのビット4をプログラムするためのバス: セット (1またはON) あるいはリセット (0またはOFF)。 ビット4のステータスを要求する。

注: 大文字はコマンド短縮形を表わす。たとえば、“source:t11 on”を送信する代わりに、“sour:t11 on”を送信することができる。

:TTL1

:SOURce:TTL1

このコマンドバスは、デジタル出力ポートのビット1をプログラムする (かつ問い合わせる) ためのものです。このバスで使うコマンドは以下の通りです。

[:LEVel]

:SOURce:TTL1[:LEVel]

このコマンドは暗黙に含まれるもの (角括弧 [] で示す通り) で、ビット1をプログラムするためのコマンドストリングに含める必要はありません。パラメータを含めるだけで構いません。

パラメータ

b = 0またはOFF	ビット1をクリア
b = 1またはON	ビット1をセット

フォーマット :sour:t1l b

デフォルト

電源投入	全ビットをクリア
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

説明

このコマンドは、デジタル出力ポートのビット1をセット (1) またはリセット (0) するのに使います。上記のフォーマットを見てください。:LEVelはオプションですからコマンドストリングには含まれていません。

ビット1のレベルはプログラムされたその極性に依存します。ビット1の極性がアクティブハイであれば、“1またはON”は+5V、“0またはOFF”は0Vです。逆に、ビット1の極性がアクティブローであれば、“1またはON”は0V、“0またはOFF”は+5Vです。極性はOUTPutサブシステムからプログラムします。

[:LEVel] ?

:SOURce:TTL1[:LEVel]?

このコマンドの角括弧 “ [] ” 内の部分は暗黙に含まれるもので、ビット1を問い合わせるためのコマンドストリングに含める必要はありません。疑問符 (?) を含めるだけで構いません。

フォーマット :sour:t1l?

説明

この問合せコマンドは、デジタル出力ポートのビット1のステータスを要求するのに使います。上記のフォーマットを見てください。:LEVelはオプションですからコマンドストリングには含まれていません。

このコマンドを送信し、Model 7001をトークにアドレッシングすると、ビット1のステータス (0または1) がコンピュータへ送られます。

:TTL2

:SOURce:TTL2

このコマンドパスは、デジタル出力ポートのビット2をプログラムする (かつ問い合わせる) ためのものです。このパスで使うコマンドは以下の通りです。

**[:LEVel] **

:SOURce:TTL2[:LEVel]

このコマンドは暗黙に含まれるもの (角括弧 [] で示す通り) で、ビット2をプログラムするためのコマンドストリングに含める必要はありません。パラメータを含めるだけで構いません。

パラメータ	b = 0またはOFF	ビット2をクリア
	b = 1またはON	ビット2をセット
フォーマット	:sour:t12 b	
デフォルト	電源投入	全ビットをクリア
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	無効

説明

このコマンドは、デジタル出力ポートのビット2をセット (1) またはリセット (0) するのに使います。上記のフォーマットを見てください。:LEVelはオプションですからコマンドストリングには含まれていません。

ビット2のレベルはプログラムされたその極性に依存します。ビット2の極性がアクティブハイであれば、“1またはON”は+5V、“0またはOFF”は0Vです。逆に、ビット2の極性がアクティブローであれば、“1またはON”は0V、“0またはOFF”は+5Vです。極性はOUTPutサブシステムからプログラムします。

[[:LEVel]]?

:SOURce:TTL2[:LEVel]?

このコマンドの角括弧“[]”内の部分は暗黙にに含まれるもので、ビット2を問い合わせるためのコマンドストリングに含める必要はありません。疑問符(?)を含めるだけで構いません。

フォーマット

:sour:t12?

説明

この問合せコマンドは、デジタル出力ポートのビット2のステータスを要求するのに使います。上記のフォーマットを見てください。:LEVelはオプションですからコマンドストリングには含まれていません。

このコマンドを送信し、Model 7001をトークにアドレッシングすると、ビット2のステータス (0または1) がコンピュータへ送られます。

:TTL3

:SOURce:TTL3

このコマンドバスは、デジタル出力ポートのビット3をプログラムする (かつ問い合わせる) ためのものです。このバスで使うコマンドは以下の通りです。

**[[:LEVel] **

:SOURce:TTL3[:LEVel]

このコマンドは暗黙に含まれるもの（角括弧 [] で示す通り）で、ビット3をプログラムするためのコマンドストリングに含める必要はありません。パラメータを含めるだけで構いません。

パラメータ	b = 0またはOFF	ビット3をクリア
	b = 1またはON	ビット3をセット
フォーマット	:sour:ttl3 b	
デフォルト	電源投入	全ビットをクリア
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	無効

説明

このコマンドは、デジタル出力ポートのビット3をセット (1) またはリセット (0) するのに使います。上記のフォーマットを見てください。:LEVelはオプションですからコマンドストリングには含まれていません。

ビット3のレベルはプログラムされたその極性に依存します。ビット3の極性がアクティブハイであれば、“1またはON”は+5V、“0またはOFF”は0Vです。逆に、ビット3の極性がアクティブローであれば、“1またはON”は0V、“0またはOFF”は+5Vです。極性はOUTPutサブシステムからプログラムします。

[[:LEVel]?

:SOURce:TTL3[:LEVel]?

このコマンドの角括弧 “ [] ” 内の部分は暗黙に含まれるもので、ビット3を問い合わせるためのコマンドストリングに含める必要はありません。疑問符 (?) を含めるだけで構いません。

フォーマット :sour:ttl3?**説明**

この問合せコマンドは、デジタル出力ポートのビット3のステータスを要求するのに使います。上記のフォーマットを見てください。:LEVelはオプションですからコマンドストリングには含まれていません。

このコマンドを送信し、Model 7001をトークにアドレッシングすると、ビット3のステータス (0または1) がコンピュータへ送られます。

:TTL4

:SOURCE:TTL4

このコマンドパスは、デジタル出力ポートのビット4をプログラムする（かつ問い合わせる）ためのものです。このパスで使うコマンドは以下の通りです。

**[:LEVel] **

:SOURCE:TTL4[:LEVel]

このコマンドは暗黙に含まれるもの（角括弧 [] で示す通り）で、ビット4をプログラムするためのコマンドストリングに含める必要はありません。パラメータを含めるだけで構いません。

パラメータ	b = 0またはOFF	ビット4をクリア
	b = 1またはON	ビット4をセット
フォーマット	:sour:ttl4 b	
デフォルト	電源投入	全ビットをクリア
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	無効

説明

このコマンドは、デジタル出力ポートのビット4をセット (1) またはリセット (0) するのに使います。上記のフォーマットを見てください。:LEVelはオプションですからコマンドストリングには含まれていません。

ビット4のレベルはプログラムされたその極性に依存します。ビット4の極性がアクティブハイであれば、“1またはON”は+5V、“0またはOFF”は0Vです。逆に、ビット4の極性がアクティブローであれば、“1またはON”は0V、“0またはOFF”は+5Vです。極性はOUTPUTサブシステムからプログラムします。

[:LEVel]?

:SOURCE:TTL4[:LEVel]?

このコマンドの角括弧 “ [] ” 内の部分は暗黙に含まれるもので、ビット4を問い合わせるためのコマンドストリングに含める必要はありません。疑問符 (?) を含めるだけで構いません。

フォーマット

sour:ttl4?

説明

この問合せコマンドは、デジタル出力ポートのビット4のステータスを要求するのに使います。上記のフォーマットを見てください。:LEVelはオプションですからコマンドストリングには含まれていません。

このコマンドを送信し、Model 7001をトークにアドレッシングすると、ビット4のステータス (0または1) がコンピュータへ送られます。

プログラミング例

以下のプログラミング例は、SOURCEサブシステムのコマンドの使い方を示すものです。

```
10 OUTPUT 707;":sour:t11 on;t12 off;t13 on;t14 off"  
20 OUTPUT 707;":sour:t11?t12?t13?t14?"  
30 ENTER 707;AS  
40 PRINT AS  
50 END
```

- 行10 この行の4つのコマンドのうち、TTL1はビット1をセットし、TTL2はビット2をリセットし、TTL3はビット3をセットし、TTL4はビット4をリセットする。
- 行20 この行の4つの問合せコマンド、TTL1?~TTL4?は4つの出力ポートビットの状態を要求する。
- 行30 7001をトークにアドレッシングする。
- 行40 出力ポートビットの状態を問合せコマンドの送信順 (0、1、0、1) に表示させる (CRT上に)。

5.10.6 :STATusサブシステム

STATusサブシステムは、Model 7001のステータスレジスタをコントロールするのに使います。これらのレジスタと全体のステータス構造については5.6項に説明してあります。

このサブシステムのコマンドを表5-13に示します。

表5-13
STATusコマンドサマリ

コマンド	説明
:STATus	
:OPERation	オペレーションイベントレジスタをコントロールするためのパス：
[:EVENTi]?	ステータスレジスタを読み取る。
:CONDition?	条件レジスタを読み取る。
:PTRansition<n>	正トランジションレジスタをプログラムする (0~1122)。
:PTRansition?	正トランジションレジスタを読み取る。
:NTRansition<n>	負トランジションレジスタをプログラムする (0~1122)。
:NTRansition?	負トランジションレジスタを読み取る。
:ENABle<n>	イネーブルレジスタをプログラムする (0~1122)。
:ENABle?	イネーブルレジスタを読み取る。
:ARM	アームイベントレジスタをコントロールするためのパス：
[:EVENTi]?	ステータスレジスタを読み取る。
:CONDition?	条件レジスタを読み取る。
:PTRansition<n>	正トランジションレジスタをプログラムする (0または2)。
:PTRansition?	正トランジションレジスタを読み取る。
:NTRansition<n>	負トランジションレジスタをプログラムする (0または2)。
:NTRansition?	負トランジションレジスタを読み取る。
:ENABle<n>	イネーブルレジスタをプログラムする (0または2)。
:ENABle?	イネーブルレジスタを読み取る。
:SEQuence	シーケンスイベントレジスタをコントロールするためのパス：
[:EVENTi]?	ステータスレジスタを読み取る。
:CONDition?	条件レジスタを読み取る。
:PTRansition<n>	正トランジションレジスタをプログラムする (0~6)。
:PTRansition?	正トランジションレジスタを読み取る。
:NTRansition<n>	負トランジションレジスタをプログラムする (0~6)。
:NTRansition?	負トランジションレジスタを読み取る。
:ENABle<n>	イネーブルレジスタをプログラムする (0~6)。
:ENABle?	イネーブルレジスタを読み取る。

コマンド	説明
:TRIGger [:EVENT]? :CONDition? :PTRansition<n> :PTRansition? :NTRansition<n> :NTRansition? :ENABle<n> :ENABle?	トリガイベントレジスタをコントロールするためのパス： ステータスレジスタを読み取る。 条件レジスタを読み取る。 正トランシジョンレジスタをプログラムする (0または2)。 正トランシジョンレジスタを読み取る。 負トランシジョンレジスタをプログラムする (0または2)。 負トランシジョンレジスタを読み取る。 イネーブルレジスタをプログラムする (0または2)。 イネーブルレジスタを読み取る。
:PRESet	ステータスレジスタをデフォルト状態に戻す。
:QUEue [:NEXT]? :ENABle<list>	エラー待ち行列にアクセスするためのパス： 一番新しいエラーメッセージを要求する。 待ち行列に入るエラーおよびステータスメッセージを指定する。
:QUEStionable [:EVENT]? :CONDition? :PTRansition<n> :PTRansition? :NTRansition<n> :NTRansition? :ENABle<n> :ENABle?	被疑イベントレジスタをコントロールするためのパス： ステータスレジスタを読み取る。 条件レジスタを読み取る。 正トランシジョンレジスタをプログラムする。 正トランシジョンレジスタを読み取る。 負トランシジョンレジスタをプログラムする。 負トランシジョンレジスタを読み取る。 イネーブルレジスタをプログラムする。 イネーブルレジスタを読み取る。

* Model 7001では、STATusサブシステムのQUEStionableコマンドは使いません。この表に含めてありますが、単にSCPI構造の一部であるからにすぎません。

注：大文字はコマンド短縮形を表わす。たとえば、“status:operation:condition?”を送信する代わりに、“stat:oper:cond?”を送信することができる。

:OPERation

:STATus:OPERation

このコマンドパスは、オペレーションイベントレジスタをコントロールするためのものです。Model 7001では、このパスで以下のコマンドを使います。

[EVENT]?

:STATus:OPERation[:EVENT]?

:EVENTを囲んでいる角括弧 “[] ” は、コマンドが暗黙に含まれるもので、使う必要がないことを表わします (SCPI定義による)。したがって、“フォーマット”に示されているように、疑問符 (?) だけを使うことで、コマンドを簡略化できます。

フォーマット	:stat:oper?	
デフォルト	電源投入	全ビットをクリア (0)
	*RST	無効
	*CLS	全ビットをクリア (0)
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	無効

説明 この問合せコマンドは、オペレーションイベントステータスレジスタを読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。オペレーションイベントステータスレジスタを図5-17に示します。なお、この問合せコマンドはオペレーションイベントステータスレジスタをクリアします。

たとえば、取得された10進値が34の場合、これと等値の2進値は00100010です。この2進値に対して、オペレーションイベントステータスレジスタのビットB5およびB1がセットされます。

このレジスタのセットされたビットがどういう意味をもつかは、トランジションフィルタをどう設定するかによって決まります (:PTRansitionおよび:NTRansitionコマンドを参照してください)。イベントを正トランジション (PTR) にプログラムした場合、このレジスタの該当するビットは、イベントが発生を開始するとセットされます。たとえば、このレジスタのビットB10はスキャンが開始されるとセットされます。逆に、イベントを負トランジション (NTR) にプログラムした場合、ビットはイベントが完了するまでセットされません。たとえば、ビットB10はスキャンが完了するとセットされます。

以下に、オペレーションイベントステータスレジスタの各ビットについて説明します。

ビット0 — 使用せず

ビット1、Settling (Set) — このビットがセットされたということは、スキャンがリレーのセットリング期に入ったか (PTR)、またはセットリング期が終了した (NTR) ということです。

ビット2~4 — 使用せず

ビット5、Waiting for Trigger (Trig) — このビットがセットされたということは、Model 7001がトリガを待っているか (PTR)、またはトリガが発生した (NTR) ということです。

ビット6、Waiting for Arm (Arm) — このビットがセットされたということは、Model 7001がアームを待っているか (PTR)、またはアームが行なわれた (NTR) ということです。

ビット7~9 — 使用せず

ビット10、Idle — このビットがセットされたということは、Model 7001がアイドル状態にあるか (PTR)、またはModel 7001がスキャンを行なうためにアイドル状態から抜け出した (NTR) ということです。

ビット11~15 — 使用せず

ビット位置	(B15 - B11)	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
イベント	——	Idle	——	——	——	Arm	Trig	——	——	——	Set	——
10進重み	$2^{15} - 2^{11}$	1024 (2^{10})	—— (2^9)	—— (2^8)	—— (2^7)	64 (2^6)	32 (2^5)	—— (2^4)	—— (2^3)	—— (2^2)	2 (2^1)	—— (2^0)
値	——	0/1	——	——	——	0/1	0/1	——	——	——	0/1	——

値 : 1 = オペレーションイベントがセットされた。
0 = オペレーションイベントがクリアされた。

イベント : Idle = 7001のアイドル状態
Arm = アーム待ち
Trig = トリガ待ち
Set = セットリング

図5-17

オペレーションイベントステータスレジスタ

:CONDition?

:STATus:OPERation:CONDition?

フォーマット

:stat:oper:cond?

説明

この問合せコマンドは、オペレーションイベント条件レジスタの内容を読み取るのに使います。条件レジスタはオペレーションイベントステータスレジスタと同じですが、Model 7001の現在の状態を反映するように絶えず更新されるリアルタイムのレジスタです。

このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。

たとえば、取得された10進値が1024の場合、これと等値の2進値は0000010000000000です。この2進値に対して、ビットB10がセットされ、スキャンが進行中であることを表わします。なお、スキャンが完了すると、ビットB10はリセットされます。

:PTRansition <n>

:STATus:OPERation:PTRansition<n>

パラメータ

n=0 PTRレジスタをクリア
n=2 PTRレジスタのSetビット (B1) をセット

- n=32 PTRレジスタのTrigビット (B5) をセット
- n=64 PTRレジスタのArmビット (B6) をセット
- n=1024 PTRレジスタのIdleビット (B10) をセット

フォーマット :stat:oper:ptr n

デフォルト

電源投入	全ビットをセット (1)
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	全ビットをセット (1)

説明 このコマンドは、正トランジションレジスタ (PTR) をプログラムするのに使います。正トランジションは、条件レジスタの0から1への状態変化と定義されます。したがって、オペレーションイベントを正トランジションにプログラムした場合、オペレーションイベントステータスレジスタの該当するビットは、オペレーションイベント条件レジスタの対応するビットが0から1に変わるとセットされます。たとえば、Idleイベント (ビットB10) を正トランジションにプログラムした場合、オペレーションイベントステータスレジスタのIdleビット (B10) は、Model 7001がスキャンモードから抜け出ると (アームインジケータが消えると) セットされます。スキャンモードから抜け出ると、条件レジスタのビットB10は0から1に変わります (正トランジション)。レジスタ構造の詳細は5.6項を参照してください。

PTRレジスタを図5-18Aに示します。これには各ビットの10進重みが含まれています。セットしたいビットの10進重みの和がこのコマンドで送られるパラメータ (n) です。たとえば、Idle (B10) およびTrig (B5) イベントを正トランジションにプログラムするには、下記のコマンドを送信します。

stat:oper:ptr 1056

ここで、 Idle (ビットB10) = 10進値	1024
Trig (ビットB5) = 10進値	32
	n = 1056

正トランジションがオペレーションイベントステータスレジスタに及ぼす影響をまとめると、次の通りです。

オペレーションイベント	オペレーションイベントステータスレジスタへの正トランジションの影響
Settling	セットリング期が始まるとB1がセットされる。
Waiting for Trigger	Waiting for Trigger期が始まるとB5がセットされる。
Waiting for Arm	Waiting for Arm期が始まるとB6がセットされる。
Idle	Model 7001がアイドル状態になるとB10がセットされる。

PTRレジスタの内容は、:PTRansition?問合せコマンドを使って読み取ることができます。

:PTRansition?

:STATus:OPERation:PTRansition?

フォーマット

:stat:oper:ptr?

説明

この問合せコマンドは、正トランジション (PTR) レジスタ (図5-18A参照) の内容を読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。

たとえば、取得された10進値が1056の場合、これと等値の2進値は0000010000100000です。この2進値に対して、ビットB10およびB5がセットされます。

PTRレジスタは、:PTRansitionコマンドを使ってプログラムします。

:NTRansition <n>

:STATus:OPERation:NTRansition<n>

パラメータ

n=0 NTRレジスタをリセット
 n=2 NTRレジスタのSetビット (B1) をセット
 n=32 NTRレジスタのTrigビット (B5) をセット
 n=64 NTRレジスタのArmビット (B6) をセット
 n=1024 NTRレジスタのIdleビット (B10) をセット

フォーマット

:stat:oper:ntr n

デフォルト

電源投入	全ビットをクリア (0)
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	全ビットをクリア (0)

説明

このコマンドは、負トランジションレジスタ (NTR) をプログラムするのに使います。負トランジションは、条件レジスタの1から0への状態変化と定義されます。したがって、オペレーションイベントを負トランジションにプログラムした場合、オペレーションイベントステータスレジスタの該当するビットは、オペレーションイベント条件レジスタの対応するビットが1から0に変わるとセットされます。たとえば、Idleイベント (ビットB10) を負トランジションにプログラムした場合、オペレーションイベントステータスレジスタのIdleビット (B10) は、Model 7001がアイドル状態から抜け出るとセットされます。

NTRレジスタを図5-18Bに示します。これには各ビットの10進重みが含まれていません。セットしたいビットの10進重みの和がこのコマンドで送られるパラメータ (n) です。たとえば、Idle (B10) およびTrig (B5) イベントを負トランジションにプログラムするには、下記のコマンドを送信します。

stat:oper:ntr 1056

ここで、 Idle (ビットB10) = 10進値 1024
 Trig (ビットB5) = 10進値 32
 n = 1056

負トランジションがオペレーションイベントステータスレジスタに及ぼす影響をまとめると、次の通りです。

オペレーションイベント	オペレーションイベントステータスレジスタへの正トランジションの影響
Settling	セットリング期が終わるとB1がセットされる。
Waiting for Trigger	Waiting for Trigger期が終わるとB5がセットされる。
Waiting for Arm	Waiting for Arm期が終わるとB6がセットされる。
Idle	Model 7001がアイドル状態から抜け出るとB10がセットされる。

NTRレジスタの内容は、:NTRansition?問合せコマンドを使って読み取ることができます。

:NTRansition?

:STATus:OPERation:NTRansition?

フォーマット

:stat:oper:ntr?

説明

この問合せコマンドは、負トランジション (NTR) レジスタ (図5-18B参照) の内容を読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。

たとえば、取得された10進値が1056の場合、これと等値の2進値は0000010000100000です。この2進値に対して、ビットB10およびB5がセットされます。

NTRレジスタは、:NTRansitionコマンドを使ってプログラムします。

ビット位置	(B15 - B11)	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
イベント	———	Idle	———	———	———	Arm	Trig	———	———	———	Set	———
10進重み	$2^{15} - 2^{11}$	1024 (2^{10})	——— (2^9)	——— (2^8)	——— (2^7)	64 (2^6)	32 (2^5)	——— (2^4)	——— (2^3)	——— (2^2)	2 (2^1)	——— (2^0)
値	———	0/1	———	———	———	0/1	0/1	———	———	———	0/1	———

値 : 1 = 正トランジジョンをイネーブルする。
0 = 正トランジジョンをディスエーブルする。

A) 正トランジジョン (PTR) レジスタ

ビット位置	(B15 - B11)	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
イベント	———	Idle	———	———	———	Arm	Trig	———	———	———	Set	———
10進重み	$2^{15} - 2^{11}$	1024 (2^{10})	——— (2^9)	——— (2^8)	——— (2^7)	64 (2^6)	32 (2^5)	——— (2^4)	——— (2^3)	——— (2^2)	2 (2^1)	——— (2^0)
値	———	0/1	———	———	———	0/1	0/1	———	———	———	0/1	———

値 : 1 = 負トランジジョンをイネーブルする。
0 = 負トランジジョンをディスエーブルする。

B) 負トランジジョン (NTR) レジスタ

図5-18

オペレーションイベントトランジジョンフィルタ

:ENABle <n>

:STATus:OPERation:ENABle<n>

パラメータ

n=0	イネーブルレジスタをクリア
n=2	オペレーションイベントステータスイネーブルレジスタのビットB1 (Set) をセット
n=32	オペレーションイベントステータスイネーブルレジスタのビットB5 (Trig) をセット
n=64	オペレーションイベントステータスイネーブルレジスタのビットB6 (Arm) をセット
n=1024	オペレーションイベントステータスイネーブルレジスタのビットB10 (Idle) をセット

フォーマット

:stat:oper:enab n

デフォルト	電源投入	全ビットをクリア (0)
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	全ビットをクリア (0)

説明

このコマンドは、オペレーションイベントステータスイネーブルレジスタ（図5-19参照）の内容をセットするのに使います。このコマンドは、レジスタの目的の状態（0または1）を決める2進値と等値の10進値を付して送信します。なお、Model 7001では、レジスタの4ビット（B1、B5、B6、B10）だけしか使いません。

このイネーブルレジスタは、オペレーションイベントに対するマスクとして使われます。このレジスタのビットがクリア（0）されると、オペレーションイベントステータスレジスタの対応するビットがマスクされ、その結果、ステータスバイトレジスタのOperation Summary Bit (OSB) はセットできません。逆に、イネーブルレジスタのビットがセット（1）されると、オペレーションイベントステータスレジスタの対応するビットはアンマスクされます。オペレーションイベントステータスレジスタのアンマスクされたビットがセットされると、ステータスバイトレジスタのOSBビットがセットされます。

図5-19にはビットの10進重みが表示されています。セットしたいビットの10進重みの和が:ENABleコマンドのパラメータとして送信されます。たとえば、オペレーションイベントステータスイネーブルレジスタのArmビットおよびTrigビットをセットするには、下記のコマンドを送信します。

```
:stat:oper:enab 96
```

ここで、	Arm (ビットB6) = 10進値	64
	Trig (ビットB5) = 10進値	32
		96
		n = 96

オペレーションイベントビットについては、[:EVENTi]?コマンドを参照してください。オペレーションイベントステータスイネーブルレジスタは、次のコマンド (:ENABle?) を使って読み取ることができます。

ENABle?

```
:STATus:OPERation:ENABle?
```

フォーマット

```
:stat:oper:enab?
```

説明

この問合せコマンドは、オペレーションイベントステータスイネーブルレジスタ（図5-19参照）の内容を読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。

たとえば、取得された10進値が96の場合、これと等値の2進値は01100000です。この2進値に対して、オペレーションイベントステータスイネーブルレジスタのビットB5およびB6がセットされます。

オペレーションイベントステータスイネーブルレジスタは、:ENABleコマンド（前のコマンドを参照してください）を使ってプログラムします。

プログラミング例

以下のプログラミング例は、STATusサブシステムのコマンドを使ってオペレーションイベントレジスタをプログラムするものです。

```

10  OUTPUT 707;“:stat:oper:ptr 1120;ntr0;enab 1120”
20  OUTPUT 707;“:stat:oper:ptr?;enab?”
30  ENTER 707;A$
40  PRINT A$
50  END

```

行10 この行の3つのコマンドのうち、PTRansitionとNTRansitionは全イベントを正トランジションにプログラムし、ENABleは全イベントをイネーブルする。

行20 この行の3つの問合せコマンドのうち、PTRansition?とNTRansition?はトランジションフィルタレジスタを読み取り、ENABle?はイネーブルレジスタを読み取る。

行30 7001をトークにアドレッシングする。

行40 レジスタのステータスを要求された順 (1120、0、1120) に表示させる。

ビット位置	(B15 - B11)	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
イベント	——	Idle	——	——	——	Arm	Trig	——	——	——	Set	——
10進重み	$2^{15} - 2^{11}$	1024 (2^{10})	—— (2^9)	—— (2^8)	—— (2^7)	64 (2^6)	32 (2^5)	—— (2^4)	—— (2^3)	—— (2^2)	2 (2^1)	—— (2^0)
値	——	0/1	——	——	——	0/1	0/1	——	——	——	0/1	——

値 : 1 = オペレーションイベントをイネーブルする。
0 = オペレーションイベントをディスエーブル (マスク) する。

イベント : Idle = 7001のアイドル状態
Arm = Waiting for Arm
Trig = Waiting for Trigger
Set = Settling

図5-19

オペレーションイベントステータスイネーブルレジスタ

:ARM

:STATus:OPERation:ARM

このバスの以下のコマンドは、アームイベントレジスタをコントロールするのに使います。

[[:EVENT]?]

:STATus:OPERation:ARM[:EVENT]?

:EVENTを囲んでいる角括弧“[]”は、コマンドが暗黙に含まれるもので、使う必要はないことを表わします（SCPI定義による）。したがって、“フォーマット”に示されているように、疑問符（?）だけを使うことで、コマンドを簡略化できます。

フォーマット

:stat:oper:arm?

デフォルト

電源投入	全ビットをクリア (0)
*RST	無効
*CLS	全ビットをクリア (0)
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

説明

この問合せコマンドは、アームイベントステータスレジスタを読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。アームイベントステータスレジスタを図5-20に示します。このレジスタはSeq1ビット (B1) だけしか使いませんから、注意してください。

ビット位置	B15	(B14 - B2)	B1	B0
イベント	—	—	Seq1	—
10進重み	(2 ¹⁵)	(2 ¹⁴ - 2 ²)	2 (2 ¹)	(2 ⁰)
値	—	—	0/1	—

値 : 1 = イベントビットがセットされた。
0 = イベントビットがクリアされた。

イベント : Seq1 = Sequence1

図5-20**アームイベントステータスレジスタ**

たとえば、取得された値が2の場合、これと等値の2進値は0000000000000010です。この2進値に対して、アームイベントステータスレジスタのビットB1がセットされます。

セットされたビットB1がどういう意味をもつかは、トランジションフィルタをどう設定するかによって決まります (:PTRansitionおよび:NTRansitionコマンドを参照してください)。アームイベントを正トランジション (PTR) にプログラムした場合、このレジスタのビットB1は、Model 7001が最初にアームレイヤに入るとセットされます。

逆に、アームイベントを負トランジション (NTR) にプログラムした場合、ビットB1はイベントが完了するまでセットされません。

アームイベントステータスレジスタのビットのうち使用するのはSeq1ビット (B1) だけです。以下に、このビットについて説明します。

ビットB1、Sequence 1 (Seq1) — このビットがセットされたということは、Model 7001がSequence 1のアームレイヤに入ったか (PTR)、またはModel 7001がSequence 1のアームレイヤから出た (NTR) ということです。

:CONDition?

:STATus:OPERation:ARM:CONDition?

フォーマット

:stat:oper:arm:cond?

説明

この問合せコマンドは、アームイベント条件レジスタの内容を読み取るのに使います。このレジスタはアームイベントステータスレジスタと同じですが、Model 7001の現在の状態を反映するように絶えず更新されるリアルタイムのレジスタです。

このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。なお、Model 7001で使うのはビットB1(Seq1)だけです。

たとえば、取得された10進値が2の場合、これと等値の2進値は000000000000010です。この2進値に対して、ビットB1がセットされ、Model 7001がSequence 1のアームレイヤに入ったことを表わします。Model 7001がアームレイヤから出ると、ビットB1はリセットされます。

:PTRansition <n>

:STATus:OPERation:ARM:PTRansition<n>

パラメータ

n=0 PTRレジスタをリセット
n=2 PTRレジスタのビットB1 (Seq1)^{*}をセット

フォーマット

:stat:oper:arm:ptr n

デフォルト

電源投入	全ビットをセット (1)
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	全ビットをセット (1)

説明

このコマンドは、正トランジション (PTR) レジスタ (図5-21A参照) をプログラムするのに使います。正トランジションは、条件レジスタの0から1への状態変化と定義されます。したがって、アームイベントを正トランジションにプログラムした場合、アームイベントステータスレジスタのビットB1は、アームイベント条件レジスタのビットB1が0から1に変わるとセットされます。

この0から1へのトランジションは、Model 7001がSequence 1のアームレイヤに入ると生じます。

PTRレジスタを図5-21Aに示します。これには各ビットの10進重みが含まれています。1ビットだけしか使いませんから、ビットB1をセットするためのパラメータ (n) は10進値の2です。下記のコマンドは、Seq1 (B1) を正トランジションにプログラムするものです。

```
:stat:oper:arm:ptr 2
```

PTRレジスタの内容は、:PTRansition?問合せコマンドを使って読み取ることができます。

:PTRansition?

```
:STATus:OPERation:ARM:PTRansition?
```

フォーマット

```
:stat:oper:arm:ptr?
```

説明

この問合せコマンドは、正トランジション (PTR) レジスタ (図5-21A参照) の内容を読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。

たとえば、取得された10進値が2の場合、これと等値の2進値は0000000000000010です。この2進値に対して、ビットB1がセットされます。

PTRレジスタは、:PTRansitionコマンドを使ってプログラムします。

ビット位置	B15	(B14 - B2)	B1	B0
イベント	—	—	Seq1	—
10進重み	(2 ¹⁵)	(2 ¹⁴ - 2 ²)	2 (2 ¹)	(2 ⁰)
値	—	—	0/1	—

値 : 1 = 正トランジションをイネーブルする。
0 = 正トランジションをディスエーブルする。

A) 正トランジション (PTR) レジスタ

ビット位置	B15	(B14 - B2)	B1	B0
イベント	—	—	Seq1	—
10進重み	(2 ¹⁵)	(2 ¹⁴ - 2 ²)	2 (2 ¹)	(2 ⁰)
値	—	—	0/1	—

値 : 1 = 負トランジションをイネーブルする。
0 = 負トランジションをディスエーブルする。

B) 負トランジション (NTR) レジスタ

図5-21

アームイベントトランジションフィルタ

:NTRansition<n>

:STATus:OPERation:ARM:NTRansition<n>

パラメータ

n=0 NTRレジスタをリセット
n=2 NTRレジスタのビットB1 (Seq1) をセット

フォーマット

:stat:oper:arm:ntr n

デフォルト	電源投入	全ビットをクリア (0)
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	全ビットをクリア (0)

説明

このコマンドは、負トランジション (NTR) レジスタをプログラムするのに使います(図5-21B参照)。負トランジションは、条件レジスタの1から0への状態変化と定義されます。したがって、アームイベントを負トランジションにプログラムした場合、アームイベントステータスレジスタのビットB1は、アームイベント条件レジスタのビットB1が1から0に変わるとセットされます。この1から0へのトランジションは、Model 7001がSequenc 1のアームレイヤから出ると生じます。

NTRレジスタを図5-21Bに示します。これには各ビットの10進重みが含まれています。1ビットだけしか使いませんから、ビットB1をセットするためのパラメータ (n) は10進値の2です。下記のコマンドは、Seq1 (B1) を負トランジションにプログラムするものです。

```
:stat:oper:arm:ntr 2
```

NTRレジスタの内容は、:NTRansition?問合せコマンドを使って読み取ることができます。

:NTRansition?

```
:STATus:OPERation:ARM:NTRansition?
```

フォーマット

```
:stat:oper:arm:ntr?
```

説明

この問合せコマンドは、負トランジション (NTR) レジスタ (図5-21B参照) の内容を読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。

たとえば、取得された10進値が2の場合、これと等値の2進値は0000000000000010です。この2進値に対して、ビットB1がセットされます。

NTRレジスタは、:NTRansitionコマンドを使ってプログラムします。

:ENABLE <n>

```
:STATus:OPERation:ARM:ENABle<n>
```

パラメータ

n=0 イネーブルレジスタをクリア
n=2 アームイベントステータスイネーブルレジスタのビットB1(Seq1)をセット

フォーマット

```
:stat:oper:arm:enab n
```

デフォルト	電源投入	全ビットをクリア (0)
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	全ビットをクリア (0)

説明

このコマンドは、アームイベントステータスイネーブルレジスタ (図5-22参照) の内容をセットするのに使います。このコマンドは、レジスタのビットB1の目的の状態 (0または1) を決める2進値と等値の10進値を付して送信します。Model 7001ではレジスタのビットB1だけしか使いません。

このイネーブルレジスタは、アームイベントに対するマスクとして使われます。このレジスタのビットB1がクリア (0) されると、アームイベントステータスレジスタのビットB1がマスクされ、その結果、オペレーションイベント条件レジスタのビットB6 (Waiting for Arm) はセットできません。逆に、イネーブルレジスタのビットB1がセット (1) されると、アームイベントステータスレジスタのビットB1はアンマスクされます。アームイベントステータスレジスタのアンマスクされたビットがセットされると、オペレーションイベント条件レジスタのビットB6がセットされます。

図5-22にはビットの10進重みが表示されています。1ビットだけしか使いませんから、ビットB1をセットするためのパラメータ (n) は10進値の2です。下記のコマンドは、アームイベント (Seq1) をアンマスク (イネーブル) するものです。

```
:stat:oper:Arm:enab 2
```

アームイベントステータスイネーブルレジスタは、次のコマンド (:ENABLE?) を使って読み取ることができます。

:ENABLE?

```
:STATus:OPERation:ARM:ENABle?
```

フォーマット

```
:stat:oper:arm:enab?
```

説明

この問合せコマンドは、アームイベントステータスイネーブルレジスタ (図5-22参照) の内容を読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。

たとえば、取得された10進値が2の場合、これと等値の2進値は0000000000000010です。この2進値に対して、アームイベントステータスイネーブルレジスタのビットB1がセットされます。

アームイベントステータスイネーブルレジスタは、:ENABLEコマンド (前のコマンドを参照してください) を使ってプログラムします。

プログラミング例

以下のプログラミング例は、STATusサブシステムのコマンドを使ってアームイベントレジスタをプログラムするものです。

```
10 OUTPUT 707;":stat:oper:arm:ptr 2;ntr 0;enab 2"
20 OUTPUT 707;":stat:oper:arm:ptr?;ntr?;enab?"
```

```

30  ENTER 707;AS
40  PRINT AS
50  END

```

行10 この行の3つのコマンドのうち、:PTRansitionと:NTRansitionは全イベントを正
トランジションにプログラムし、ENABleは全イベントをイネーブルする。

行20 この行の3つのコマンドのうち、:PTRansition?と:NTRansition?はトランジ
ョンフィルタレジスタを読み取り、ENABle?はイネーブルレジスタを読み取
る。

行30 7001をトークにアドレッシングする。

行40 レジスタのステータスを要求された順 (2、0、2) に表示させる。

ビット位置	B15	(B14 - B2)	B1	B0
イベント	—	—	Seq1	—
10進重み	(2 ¹⁵)	(2 ¹⁴ - 2 ²)	2 (2 ¹)	(2 ⁰)
値	—	—	0/1	—

値 : 1 = アームイベントをイネーブルする。
0 = アームイベントをディスエーブル (マスク) する。

イベント : Seq1 = Sequence1

図5-22

アームイベントステータスイネーブルレジスタ

:SEQuence

:STATus:OPERation:ARM:SEQuence

このバスの以下のコマンドは、シーケンスイベントレジスタをコントロールするのに使います。

[:EVENT] ?

:STATus:OPERation:ARM:SEquence[:EVENT]?

:EVENTを囲んでいる角括弧 “ [] ” は、コマンドが暗黙に含まれるもので、使う必要はないことを表わします (SCPI定義による)。したがって、“フォーマット”に示されているように、疑問符 (?) だけを使うことで、コマンドを簡略化できます。

フォーマット

:stat:oper:arm:seq?

デフォルト

電源投入	全ビットをクリア (0)
*RST	無効
*CLS	全ビットをクリア (0)
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

説明

この問合せコマンドは、シーケンスイベントステータスレジスタを読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。シーケンスイベントステータスレジスタを図5-23に示します。このレジスタは2ビット (B1およびB2) だけしか使いませんから、注意してください。この問合せコマンドは、シーケンスイベントステータスレジスタをクリアします。

たとえば、取得された10進値が4の場合、これと等値の2進値は000000000000100です。この2進値に対して、シーケンスイベントステータスレジスタのビットB2がセットされます。

セットされたビットB2がどういう意味をもつかは、トランジションフィルタをどう設定するかによって決まります (:PTRansitionおよび:NTRansitionコマンドを参照してください)。シーケンスイベントを正トランジション (PTR) にプログラムした場合、このレジスタのビットB2は、Model 7001が最初にArm 1レイヤに入るとセットされます。逆に、シーケンスイベントを負トランジション (NTR) にプログラムした場合、ビットB2はイベントが完了するまでセットされません。

シーケンスイベントステータスレジスタのビットのうち使用するのはLay 1 (B1) と Lay 2 (B2) だけです。以下に、これらのビットについて説明します。

ビットB1、Layer1 (Lay1) — このビットがセットされたということは、Model 7001がArm1 レイヤに入ったか (PTR)、またはそのレイヤから出た (NTR) ということです。

ビットB2、Layer2 (Lay2) — このビットがセットされたということは、Model 7001がArm2 レイヤに入ったか (PTR)、またはそのレイヤから出た (NTR) ということです。

ビット位置	B15	(B14 - B3)	B2	B1	B0
イベント	—	—	Lay2	Lay1	—
10進値	(2 ¹⁵)	(2 ¹⁴ - 2 ³)	4 (2 ²)	2 (2 ¹)	(2 ⁰)
値	—	—	0/1	0/1	—

値 : 1 = イベントビットがセットされた。

0 = イベントビットがクリアされた。

イベント : Lay1 = Layer 1

Lay2 = Layer 2

図5-23

シーケンスイベントステータスレジスタ

:CONDition?

:STATus:OPERation:ARM:SEQuence:CONDition?

フォーマット

:stat:oper:arm:seq:cond?

説明

この問合せコマンドは、シーケンスイベント条件レジスタの内容を読み取るのに使います。このレジスタはシーケンスイベントステータスレジスタと同様ですが、Model 7001の現在の状態を反映するように絶えず更新されます。

このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。

たとえば、取得された10進値が2の場合、これと等値の2進値は0000000000000010です。この2進値に対して、ビットB1がセットされ、Model 7001がArm1レイヤに入ったことを表わします。なお、Model 7001がこのレイヤから出ると、ビットB1はリセットされます。

:PTRansition <n>

:STATus:OPERation:ARM:SEQuence:PTRansition<n>

パラメータ

n=0 PTRレジスタをリセット
 n=2 PTRレジスタのビットB1 (Lay1) をセット
 n=4 PTRレジスタのビットB2 (Lay2) をセット

フォーマット

:stat:oper:arm:seq:ptr n

デフォルト

電源投入	全ビットをセット (1)
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	全ビットをセット (1)

説明

このコマンドは、正トランジション (PTR) レジスタ (図5-24A参照) をプログラムするのに使います。正トランジションは、条件レジスタの0から1への状態変化と定義されます。したがって、シーケンスイベントを正トランジションにプログラムした場合、シーケンスイベントステータスレジスタの該当するビットは、シーケンスイベント条件レジスタの対応するビットが0から1に変わるとセットされます。この0から1へのトランジションは、Model 7001がArm1レイヤに入ると生じます。

PTRレジスタを図5-24Aに示します。これには各ビットの10進重みが含まれています。セットしたいビットの10進重みの和がこのコマンドで送られるパラメータ (n) です。たとえば、Lay1 (B1) とLay2 (B2) を両方とも正トランジションにプログラムするには、下記のコマンドを送信します。

:stat:oper:arm:seq:ptr 6

ここで、 Lay1 (ビットB1) = 10進値	2
Lay2 (ビットB2) = 10進値	4
	n=6

PTRレジスタの内容は、:PTRansition?問合せコマンドを使って読み取ることができます。

:PTRansition?

:STATus:OPERation:ARM:SEQuence:PTRansition?

フォーマット

:stat:oper:arm:seq:ptr?

説明

この問合せコマンドは、正トランジション (PTR) レジスタ (図5-24A参照) の内容を読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。

たとえば、取得された10進値が6の場合、これと等値の2進値は000000000000110です。この2進値に対して、ビットB1およびB2がセットされます。

PTRレジスタは、:PTRansitionコマンドを使ってプログラムします。

:NTRansition <n>

:STATus:OPERation:ARM:SEQuence:NTRansition<n>

パラメータ	n=0	NTRレジスタをリセット
	n=2	NTRレジスタのビットB1 (Lay1) をセット
	n=4	NTRレジスタのビットB2 (Lay2) をセット

フォーマット :stat:oper:arm:seq:ntr n

デフォルト	電源投入	全ビットをクリア (0)
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	全ビットをクリア (0)

説明

このコマンドは、負トランジション (NTR) レジスタをプログラムするのに使います。負トランジションは、条件レジスタの1から0への状態変化と定義されます。したがって、シーケンスイベントを負トランジションにプログラムした場合、シーケンスイベントステータスレジスタの該当するビットは、シーケンスイベント条件レジスタの対応するビットが1から0に変わるとセットされます。この1から0へのトランジションは、Model 7001がArm1 レイヤから出ると生じます。

NTRレジスタを図5-24Bに示します。これには各ビットの10進重みが含まれています。セットしたいビットの10進重みの和がこのコマンドで送られるパラメータ (n) です。たとえば、Lay1 (B1) とLay2 (B2) を両方も負トランジションにプログラムするには、下記のコマンドを送信します。

:stat:oper:arm:seq:ntr 6

ここで、	Lay1 (ビットB1)	= 10進値	2
	Lay2 (ビットB2)	= 10進値	4
			$\overline{n=6}$

NTRレジスタの内容は、:NTRansition?問合せコマンドを使って読み取ることができます。

:NTRansition?

:STATus:OPERation:ARM:SEQuence:NTRansition?

フォーマット :stat:oper:arm:seq:ntr?

説明

この問合せコマンドは、負トランジション (NTR) レジスタ (図5-24B参照) の内容を読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。

たとえば、取得された10進値が2の場合、これと等値の2進値は0000000000000010です。この2進値に対して、ビットB1がセットされます。

NTRレジスタは、:NTRansitionコマンドを使ってプログラムします。

ビット位置	B15	(B14 - B3)	B2	B1	B0
イベント	——	——	Lay2	Lay1	——
10進重み	(2 ¹⁵)	(2 ¹⁴ - 2 ³)	4 (2 ²)	2 (2 ¹)	(2 ⁰)
値	——	——	0/1	0/1	——

値 : 1 = 正トランジションをイネーブルする。
0 = 正トランジションをディスエーブルする。

A) 正トランジション (PTR) レジスタ

ビット位置	B15	(B14 - B3)	B2	B1	B0
イベント	——	——	Lay2	Lay1	——
10進重み	(2 ¹⁵)	(2 ¹⁴ - 2 ³)	4 (2 ²)	2 (2 ¹)	(2 ⁰)
値	——	——	0/1	0/1	——

値 : 1 = 負トランジションをイネーブルする。
0 = 負トランジションをディスエーブルする。

B) 負トランジション (NTR) レジスタ

図5-24

シーケンスイベントトランジションフィルタ

:ENABle <n>

:STATus:OPERation:ARM:SEQuence:ENABle<n>

パラメータ

n=0	イネーブルレジスタをクリア
n=2	シーケンスイベントステータスイネーブルレジスタのビットB1 (Lay1) をセット
n=4	シーケンスイベントステータスイネーブルレジスタのビットB2 (Lay2) をセット

フォーマット	:stat:oper:arm:seq:enab n	
デフォルト	電源投入	全ビットをクリア (0)
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	全ビットをクリア (0)

説明 このコマンドは、シーケンスイベントステータスイネーブルレジスタ（図5-25参照）の内容をセットするのに使います。このコマンドは、レジスタのビットB1およびB2の目的の状態（0または1）を決める2進値と等値の10進値を付して送信します。

このイネーブルレジスタは、シーケンスイベントに対するマスクとして使われます。このレジスタのビット（B1またはB2）がクリア（0）されると、シーケンスイベントステータスレジスタの対応するビットがマスク（ディスエーブル）され、その結果、アームイベント条件レジスタのSeq1ビット（In an arm layer of Sequence 1）はセットできません。逆に、シーケンスイベントステータスイネーブルレジスタがセット（1）されると、シーケンスイベントステータスレジスタの対応するビットはアンマスク（イネーブル）されます。アンマスクされたビットがセットされると、アームイベント条件レジスタのSeq1ビット（B1）がセットされます。

シーケンスイベントステータスイネーブルレジスタを図5-25に示します。これには各ビットの10進重みが含まれています。アンマスクしたいビットの10進重みの和が:ENABleコマンドで送られるパラメータ（n）です。たとえば、両方のシーケンスイベントビット（B1およびB2）をアンマスクしたければ、下記のコマンドを送信します。

```
:stat:oper:arm:seq:enab 6
```

ここで、	Lay1（ビットB1）	= 10進値	2
	Lay2（ビットB2）	= 10進値	4
			6
			n=6

シーケンスイベントステータスイネーブルレジスタは、次のコマンド (:ENABle?) を使って読み取ることができます。

:ENABle?

```
:STATus:OPERation:ARM:SEQuence:ENABle?
```

フォーマット	:stat:oper:arm:seq:enab?
--------	--------------------------

説明 この問合せコマンドは、シーケンスイベントステータスイネーブルレジスタ（図5-25参照）の内容を読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。

たとえば、取得された10進値が6の場合、これと等値の2進値は000000000000110です。この2進値に対して、シーケンスイベントステータスイネーブルレジスタのビットB1およびB2がセットされます。

シーケンスイベントステータスイネーブルレジスタは、:ENABLEcコマンド（前のコマンドを参照してください）を使ってプログラムします。

プログラミング例

以下のプログラミング例は、STATusサブシステムのコマンドを使ってシーケンスイベントレジスタをプログラムするものです。

```
10 OUTPUT 707;":stat:oper:arm:seq:ptr 6;ntr 0;enab 6"
20 OUTPUT 707;":stat:oper:arm:seq:ptr?;ntr?;enab?"
30 ENTER 707;AS
40 PRINT AS$
50 END
```

行10 この行の3つのコマンドのうち、PTRとNTRは全イベントを正トランジションにプログラムし、ENABLEは全イベントをイネーブルする。

行20 この行の3つの問合せコマンドのうち、PTR?とNTR?はトランジションフィルタレジスタを読み取り、ENABLE?はイネーブルレジスタを読み取る。

行30 7001をトークにアドレッシングする。

行40 レジスタのステータスを要求された順（6、0、6）に表示させる。

ビット位置	B15	(B14 - B3)	B2	B1	B0
イベント	—	—	Lay2	Lay1	—
10進重み	(2 ¹⁵)	(2 ¹⁴ - 2 ³)	4 (2 ²)	2 (2 ¹)	(2 ⁰)
値	—	—	0/1	0/1	—

値 : 1 = シーケンスイベントをイネーブルする。
0 = シーケンスイベントをディスエーブル(マスク)する。

イベント : Lay1 = Layer1
Lay2 = Layer2

図5-25

シーケンスイベントステータスイネーブル

:TRIGger

:STATus:OPERation:ARM:TRIGger

このパスの以下のコマンドは、トリガイイベントレジスタをコントロールするのに使います。

[:EVENT]?

:STATus:OPERation:ARM:TRIGger[:EVENT]?

:EVENTを囲んでいる角括弧“[]”は、コマンドが暗黙に含まれるもので、使う必要はないことを表わします（SCPI定義による）。したがって、“フォーマット”に示されているように、疑問符（?）だけを使うことで、コマンドを簡略化できます。

フォーマット

:stat:oper:trig?

デフォルト

電源投入	全ビットをクリア (0)
*RST	無効
*CLS	全ビットをクリア (0)
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

説明

この問合せコマンドは、トリガイイベントステータスレジスタを読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。トリガイイベントステータスレジスタを図5-26に示します。このレジスタはSeq1ビット (B1) だけしか使いませんから、注意してください。この問合せコマンドは、トリガイイベントステータスレジスタをクリアします。

たとえば、取得された10進値が2の場合、これと等値の2進値は000000000000010です。この2進値に対して、トリガイイベントステータスレジスタのビットB1がセットされます。

セットされたビットB1がどういう意味をもつかは、トランジションフィルタをどう設定するかによって決まります (:PTRansitionおよび:NTRansitionコマンドを参照してください)。トリガイイベントを正トランジション (PTR) にプログラムした場合、このレジスタのビットB1は、Model 7001が最初にSequence 1のトリガイレイヤに入るとセットされます。逆に、トリガイイベントを負トランジションにプログラムした場合、ビットB1はイベントが完了するまでセットされません。

トリガイイベントステータスレジスタのビットのうち使用するのはSeq1ビット (B1) だけです。以下に、このビットについて説明します。

ビットB1、Sequence 1 (Seq1) — このビットがセットされたということは、Model 7001がSequence 1 のトリガイレイヤに入ったか (PTR)、またはそのレイヤから出た (NTR) ということです。

ビット位置	B15	(B14 - B2)	B1	B0
イベント	—	—	Seq1	—
10進重み	(2 ¹⁵)	(2 ¹⁴ - 2 ²)	2 (2 ¹)	(2 ⁰)
値	—	—	0/1	—

値 : 1 = イベントビットがセットされた。
 0 = イベントビットがクリアされた。
 イベント : Seq 1 = Sequence 1

図5-26

トリガイベントステータスレジスタ

:CONDition?

:STATus:OPERation:ARM:TRIGger:CONDition?

フォーマット

:stat:oper:trig:cond?

説明

この問合せコマンドは、トリガイベント条件レジスタの内容を読み取るのに使います。このレジスタはトリガイベントステータスレジスタと同様ですが、Model 7001の現在の状態を反映するように絶えず更新されます。

このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。なお、Model 7001で使うのはビットB1 (Seq1) だけです。

たとえば、取得された10進値が2の場合、これと等値の2進値は0000000000000010です。この2進値に対して、ビットB1がセットされ、Model 7001がSequence 1のトリガレイヤに入ったことを表わします。Model 7001がトリガレイヤから出ると、ビットB1はリセットされます。

:PTRansition <n>

:STATus:OPERation:ARM:TRIGger:PTRansition<n>

パラメータ

n=0 PTRレジスタをリセット
 n=2 PTRレジスタのビットB1 (Seq1) をセット

フォーマット	:stat:oper:trig:ptr n	
デフォルト	電源投入	全ビットをセット (1)
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	全ビットをセット (1)
	:STATus:PRESet	全ビットをセット (1)

説明 このコマンドは、正トランジション (PTR) レジスタ (図5-27A参照) をプログラムするのに使います。正トランジションは、条件レジスタの0から1への状態変化と定義されます。したがって、トリガイイベントを正トランジションにプログラムした場合、レジスタのビットB1は、トリガイイベント条件レジスタのビットB1が0から1に変わるとセットされます。この0から1へのトランジションは、Model 7001がSequence 1のトリガレイヤに入ると生じます。

PTRレジスタを図5-27Aに示します。これには各ビットの10進重みが含まれています。1ビットだけしか使いませんから、ビットB1をセットするためのパラメータ (n) は2進値の2です。下記のコマンドは、Seq1 (B1) を正トランジションにプログラムするものです。

```
stat:oper:trig:ptr 2
```

PTRレジスタの内容は、:PTRansition?問合せコマンドを使って読み取ることができます。

:PTRansition?

```
:STATus:OPERation:ARM:TRIGger:PTRansition?
```

フォーマット	:stat:oper:trig:ptr?
---------------	----------------------

説明 この問合せコマンドは、正トランジション (PTR) レジスタ (図5-27A参照) の内容を読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。

たとえば、取得された10進値が2の場合、これと等値の2進値は000000000000010です。この2進値に対して、ビットB1がセットされます。

PTRレジスタは、:PTRansitionコマンドを使ってプログラムします。

:NTRansition <n>

```
:STATus:OPERation:ARM:TRIGger:NTRansition<n>
```

パラメータ	n=0	NTRレジスタをリセット
	n=2	NTRレジスタのビットB1 (Seq1) をセット

フォーマット	:stat:oper:trig:ntr n
---------------	-----------------------

デフォルト	電源投入	全ビットをクリア (0)
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	全ビットをクリア (0)

説明

このコマンドは、負トランジション (NTR) レジスタ (図5-27B参照) をプログラムするのに使います。負トランジションは、条件レジスタの1から0への状態変化と定義されます。したがって、トリガイベントを負トランジションにプログラムした場合、トリガイベントステータスレジスタのビットB1は、トリガイベント条件レジスタのビットB1が1から0に変わるとセットされます。この1から0へのトランジションは、Model 7001がSequence 1のトリガレイヤから出ると生じます。

NTRレジスタを図5-27Bに示します。これには各ビットの10進重みが含まれています。1ビットだけしか使いませんから、B1をセットするためのパラメータ (n) は10進値の2です。下記のコマンドは、Seq1 (B1) を負トランジションにプログラムするものです。

```
stat:oper:trig:ntr 2
```

NTRレジスタの内容は、:NTRansition?問合せコマンドを使って読み取ることができます。

:NTRansition?

```
:STATus:OPERation:ARM:TRIGger:NTRansition?
```

フォーマット

```
:stat:oper:trig:ntr?
```

説明

この問合せコマンドは、負トランジション (NTR) レジスタ (図5-27B参照) の内容を読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。

たとえば、取得された10進値が2の場合、これと等値の2進値は0000000000000010です。この2進値に対して、ビットB1がセットされます。

NTRレジスタは、:NTRansitionコマンドを使ってプログラムします。

ビット位置	B15	(B14 - B2)	B1	B0
イベント	—	—	Seq1	—
10進重み	(2 ¹⁵)	(2 ¹⁴ - 2 ²)	2 (2 ¹)	(2 ⁰)
値	—	—	0/1	—

値 : 1 = 正トランジジョンをイネーブルする。
0 = 正トランジジョンをディスエーブルする。

A) 正トランジジョン (PTR) レジスタ

ビット位置	B15	(B14 - B2)	B1	B0
イベント	—	—	Seq1	—
10進重み	(2 ¹⁵)	(2 ¹⁴ - 2 ²)	2 (2 ¹)	(2 ⁰)
値	—	—	0/1	—

値 : 1 = 負トランジジョンをイネーブルする。
0 = 負トランジジョンをディスエーブルする。

B) 負トランジジョン (NTR) レジスタ

図5-27

トリガイベントトランジジョンフィルタ

:ENABLE <n>

:STATus:OPERation:ARM:TRIGger:ENABle<n>

パラメータ

n=0 イネーブルレジスタをクリア
n=2 トリガイベントステータスイネーブルレジスタのビットB1(Seq1)をセット

フォーマット

:stat:trig:enable n

デフォルト	電源投入	全ビットをクリア (0)
	*RST	無効
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	無効
	:STATus:PRESet	全ビットをクリア (0)

説明

このコマンドは、トリガイベントステータスイネーブルレジスタ (図5-28参照) の内容をセットするのに使います。このコマンドは、レジスタのビットB1の目的の状態 (0または1) を決める2進値と等値の10進値を付して送信します。

このイネーブルレジスタは、トリガイベントに対するマスクとして使われます。ビットB1がクリア (0) されると、トリガイベントステータスレジスタのビットB1がマスクされ、その結果、オペレーションイベント条件レジスタのビットB5 (Waiting for Trigger) はセットできません。逆に、イネーブルレジスタのビットB1がセット (1) されると、トリガイベントステータスレジスタのビットB1はアンマスクされます。トリガイベントステータスレジスタのアンマスクされたビットがセットされると、オペレーションイベント条件レジスタのビットB5がセットされます。

トリガイベントステータスイネーブルレジスタを図5-28に示します。これには各ビットの10進重みが含まれています。1ビットだけしか使いませんから、ビットB1をセットするためのパラメータ (n) は10進値の2です。下記のコマンドは、トリガイベント (Seq1) をアンマスク (イネーブル) するものです。

```
:stat:oper:trig:enab 2
```

トリガイベントステータスイネーブルレジスタは、次のコマンド (:ENABLE?) を使って読み取ることができます。

:ENABLE?

```
:STATus:OPERation:ARM:TRIGger:ENABLE?
```

フォーマット

```
:stat:oper:trig:enab?
```

説明

この問合せコマンドは、トリガイベントステータスイネーブルレジスタ (図5-28参照) の内容を読み取るのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、10進値がコンピュータへ送られます。この10進値と等値の2進値により、レジスタのどのビットがセットされるかが決まります。

たとえば、取得された10進値が2の場合、これと等値の2進値は0000000000000010です。この2進値に対して、トリガイベントステータスイネーブルレジスタのビットB1がセットされます。

トリガイベントステータスイネーブルレジスタは、:ENABLEコマンド (前のコマンドを参照してください) を使ってプログラムします。

プログラミング例

以下のプログラミング例は、STATusサブシステムのコマンドを使ってトリガイベントレジスタをプログラムするものです。

```
10 OUTPUT 707;":stat:oper:trig:ptr 2;ntr 0;enab 2"
20 OUTPUT 707;":stat:oper:trig:ptr?;ntr?;enab?"
```



```

30  ENTER 707;A$
40  PRINT A$
50  END

```

行10 この行の3つのコマンドのうち、PTRとNTRは全イベントを正トランジションにプログラムし、ENABLEは全イベントをイネーブルする。
行20 この行の3つの間合せコマンドのうち、PTR?とNTR?はトランジションフィルタレジスタを読み取り、ENABLE?はイネーブルレジスタを読み取る。
行30 7001をトークにアドレッシングする。
行40 レジスタのステータスを要求された順 (2、0、2) に表示させる。

ビット位置	B15	(B14 - B2)	B1	B0
イベント	—	—	Seq1	—
10進重み	(2 ¹⁵)	(2 ¹⁴ - 2 ²)	2 (2 ¹)	(2 ⁰)
値	—	—	0/1	—

値 : 1 = トリガイベントをイネーブルする。
0 = トリガイベントをディスエーブル (マスク) する。
イベント : Seq 1 = Sequence 1

図5-28

トリガイベントステータスイネーブルレジスタ

:PRESet

:STATus:PRESet

フォーマット

:stat:pres

説明

このコマンドを送信すると、SCPIイベントレジスタは次のような影響を受けます。

1. 正トランジションフィルタレジスタの全ビットが1にセットされる。
2. 負トランジションフィルタレジスタの全ビットが0にリセットされる。
3. 下記のレジスタの全ビットが0にリセットされる。
 - A. オペレーションイベントイネーブルレジスタ。
 - B. 被疑イベントステータスイネーブルレジスタ。

4. 下記のレジスタの全ビットが*1にセットされる。
 - A. トリガイベントステータスイベントレジスタ。
 - B. アームイベントステータスイベントレジスタ。
 - C. シーケンスイベントステータスイネーブルレジスタ。

注：上記以外のレジスタはこのコマンドの影響は受けません。

プログラミング例

以下のプログラミング例は、STATusサブシステムの:PRESetコマンドの使い方を示すものです。

```
OUTPUT 707,;stat:pres
```

:QUEue

```
:STATus:QUEue
```

このコマンドパスは、エラー/ステータス待ち行列からエラーおよびステータスメッセージを読み取るためのものです。このパスで使うコマンドは以下の通りです。

[[:NEXT]]?

```
:STATus:QUEue[:NEXT]?
```

このコマンドを囲んでいる角括弧“[]”は、コマンドが暗黙に含まれるもので、“フォーマット”に示されているように送信する必要がないことを表わします。

フォーマット

```
:stat:que?
```

デフォルト

電源投入	エラー/ステータス待ち行列をクリア
*RST	無効
*CLS	エラー/ステータス待ち行列をクリア
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

説明

エラーおよびステータスメッセージが発生すると、それらはエラー/ステータス待ち行列に入れられます。この問合せコマンドは、これらのメッセージを読み取るのに使います。

エラー/ステータス待ち行列は、先入れ先出し (FIFO) 方式のレジスタです。待ち行列を読むと、その都度、“一番古い”メッセージが読み取られ、そのメッセージは待ち行列から削除されます。待ち行列には最大10メッセージが入ります。待ち行列が一杯になると、メッセージ“350, 'Queue Overflow”がレジスタの最後のメモリロケーションを占めます。電源を投入したとき、エラー/ステータス待ち行列は空です。エラー待ち行列が空になると、メッセージ“0, 'No error”がエラー待ち行列に入れられます。

待ち行列内のメッセージはそれぞれ前に番号が付されます。SCPI定義メッセージには負 (-) の番号が使われ、Keithley定義メッセージには正 (+) の番号が使われます。これらのメッセージは表5-3に示してあります。

このコマンドを送信し、Model 7001をトークにアドレッシングすると、待ち行列内の“一番古い”メッセージがコンピュータへ送られます。

注: STATus:QUEue?問合せコマンドは、:SYSTem:ERRor?問合せコマンドと同じ機能を果たします (SYSTemサブシステムを参照してください)。

プログラミング例

以下のプログラムは、エラー/ステータス待ち行列内の“一番古い”メッセージを読み取るものです。

```
5   DIM A$ [30]
10  OUTPUT 707;";stat:que?"
20  ENTER 707;A$
30  PRINT A$
40  END
```

行5 メッセージの長さに合わせてストリングサイズを大きくする。
 行10 待ち行列内の“一番古い”メッセージを要求する。
 行20 Model 7001をトークにアドレッシングする。
 行30 メッセージをCRT上に表示させる。

:ENABle <list>

:STATus:QUEue:ENABle<list>

パラメータ

list = (numlist)

ここで、numlistは、エラー/ステータス待ち行列に入れるためにイネーブルしたいメッセージの、指定されたリスト。

フォーマット

:stat:que:enab (numlist)

デフォルト

電源投入	リストをクリア
*RST	無効
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

説明

電源を投入すると、全てのエラーおよびステータスメッセージはイネーブルされ、発生の都度、エラー/ステータス待ち行列に入ります。このコマンドは、待ち行列に入りたいメッセージを指定するのに使います。このコマンドを使う場合、指定されないメッセージはディスエーブルされ、待ち行列に入れません。メッセージは番号によって指定されます (表5-3参照)。下例は、メッセージのnumlistを表現するための各種の形式を示すものです。

```
Numlist = -110           シングルメッセージ。
          = -110, -140, -222 カンマで区切られたメッセージ。
          = -110:-222      メッセージのレンジ (-110~-222)。
          = -110:-222, -230 カンマで区切られたレンジ入力とシングルメッセージ入力。
```

注: 全てのメッセージをエラー/ステータス待ち行列に入らないようにするには、下記のコマンドを送信します。

```
:stat:que:enab ()
```

:QUEStionable

:STATus:QUEStionable

:QUEStionableコマンドは、被疑イベントレジスタをコントロールするためのバスとして使うものです。Model 7001ではこれらのレジスタは使いませんから、本書では説明しません。これらのレジスタが存在し（SCPI規格に従って）、STATusサブシステムの他のコマンドと同様にアクセスできるということを覚えておいてください。

5.10.7 SYSTemサブシステム

SYSTemサブシステムには、表5-14に示す雑コマンドが含まれます。

表5-14
SYSTemコマンドサマリ

コマンド	説明
:SYSTem	
:ERRor?	エラー/ステータス待ち行列にメッセージを要求する。
:POSetup RST PRESet SAV0 SAV1 SAV2 SAV3 SAV4 SAV5 SAV6 SAV7 SAV8 SAV9	電源投入時セットアップを選択する。
:PRESet	Model 7001をデフォルト状態(表5-6参照)に戻す。
:VERsion?	SCPI規格の改訂水準を要求する。

注：大文字はコマンド短縮形を表わす。たとえば、“system:error?”を送信する代わりに、“syst:err?”を送信することができる。

:ERRor?

:SYSTem:ERRor?

フォーマット :syst:err?

デフォルト

電源投入	エラー/ステータス待ち行列をクリア
*RST	無効
*CLS	エラー/ステータス待ち行列をクリア
DCL、SDC	無効
:SYSTem:PRESet	無効
:STATus:PRESet	無効

説明

エラーおよびステータスメッセージが発生すると、それらはエラー/ステータス待ち行列に入れられます。この問合せコマンドは、これらのメッセージを読み取るのに使います。

エラー/ステータス待ち行列は、先入れ先出し (FIFO) 方式のレジスタです。待ち行列を読むと、その都度、“一番古い”メッセージが読み取られ、そのメッセージは待ち行列から削除されます。待ち行列には最大10メッセージが入ります。待ち行列が一杯になると、メッセージ “350, 'Queue Overflow” がレジスタの最後のメモリロケーションを占めます。電源を投入したとき、エラー/ステータス待ち行列は空です。エラー待ち行列が空になると、メッセージ “0, 'No error” がエラー待ち行列に入れられます。

待ち行列内のメッセージはそれぞれ前に番号が付されます。SCPI定義メッセージには負 (-) の番号が使われ、Keithley定義メッセージには正 (+) の番号が使われます。これらのメッセージは表5-3に示してあります。

このコマンドを送信し、Model 7001をトークにアドレッシングすると、待ち行列内の“一番古い”メッセージがコンピュータへ送られます。

注: :SYSTem:ERRor?問合せコマンドは、:STATus:QUEue?問合せコマンドと同じ機能を果たします (STATusサブシステムを参照してください)。

プログラミング例

以下のプログラムは、エラー/ステータス待ち行列内の“一番古い”メッセージを読み取るものです。

```
5   DIM AS [30]
10  OUTPUT 707;":sys:err?"
20  ENTER 707;AS
30  PRINT AS
40  END
```

行5 メッセージの長さに合わせてストリングサイズを大きくする。
 行10 待ち行列内の“一番古い”メッセージを要求する。
 行20 Model 7001をトークにアドレッシングする。
 行30 メッセージをCRT上に表示させる。

:POSetup RST | PRESet | SAV0 | SAV1 | SAV2 | SAV3 | SAV4 | SAV5 | SAV6 | SAV7 | SAV8 | SAV9

:SYSTem:POSeup RST|PRESet|SAV0|SAV1|SAV2|SAV3|SAV4|SAV5|SAV6|SAV7|SAV8|SAV9

パラメータ

RST	*RSTデフォルト状態
PRES	*SYSTem:PRESetデフォルト状態
SAV0-SAV9	ユーザ定義デフォルト状態

フォーマット

```
:sys:pos rst
:sys:pos pres
:sys:pos savX
ここで、X = 1~9
```

説明

このコマンドは、Model 7001の電源投入時デフォルトをプログラムするのに使います。RSTを選択した場合、Model 7001は電源を投入すると*RSTデフォルト状態をとります。PRESを選択した場合、Model 7001は電源を投入すると:SYSTem:PRESetデフォルト状態をとります。これらのコマンドに対してとられるデフォルト状態は表5-6に示してあります。

SAV0~SAV9を選択した場合、Model 7001は、電源を投入すると、*SAVコマンドによって指定されたメモリロケーションにセーブされたセットアップをとります。

プログラミング例

以下のステートメントは、Model 7001を電源投入時に*RSTデフォルト状態をとるようにプログラムするものです。

```
OUTPUT 707;":sys:pos rst"
```

:PRESet

:SYSTem:PRESet

フォーマット

:syst:pres

説明

このコマンドは、Model 7001をフロントパネルオペレーションに最適の状態に設定します。（この状態は、:syste:pos:presコマンドによって選択される状態と同じです）。このコマンドに対してとられるデフォルト値は表5-6に示してあります。

プログラミング例

以下のステートメントは、Model 7001を:syste:pres状態にプログラムするものです。

```
OUTPUT 707;":syste:pres"
```

:VERSion?

:SYSTem:VERSion?

フォーマット

:syst:vers?

説明

この問合せコマンドは、SCPI規格のバージョンを要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、バージョンメッセージがコンピュータへ送られます。

プログラミング例

以下のプログラムは、ソフトウェアバージョンを要求するものです。

```
10 OUTPUT 707;":syst:vers?"
20 ENTER 707;A$
30 PRINT AS
40 END
```

行10 バージョンレベルを要求する。

行20 Model 7001をトークにアドレッシングする。

行30 バージョンメッセージを表示させる。

5.10.8 トリガサブシステム

トリガサブシステムは、スキャンオペレーションの3つのレイヤを設定するための一連のコマンドおよびサブシステムで構成されています。これらのコマンドおよびサブシステムを表5-15に示します。

表5-15

トリガコマンドサマリ

コマンド	説明
:INITiate [:IMMEDIATE] :CONTinuous<cb>	サブシステムコマンドバス： 1トリガ（スキャン）サイクルを開始させる。 トリガシステムの連続起動をイネーブル（1またはON）またはディスエーブル（0またはOFF）する。
:ABORt	トリガシステムをリセットし、アイドル状態に入る。
:ARM[:SEQUence[1]] [:LAYer[1]] :IMMEDIATE :COUNt<n> INFinite :COUNt? :SOURce HOLD IMMEDIATE MANual BUS TLINk EXTernal :SOURce? :SIGNal :TCONfigure :DIRection SOURce ACCeptor :DIRection? :ASYNchronous :ILINe<n> :ILINe? :OLINe<n> :OLINe?	スキャンを設定するためのサブシステムコマンドバス： スキャンのアームレイヤをプログラムするためのバス： コントロールソースをループアラウンドする。 アーム回数（1~9999またはINFinite）をプログラムする。 プログラムされたアームカウントを要求する。 Arm Layer 1コントロールソースを選択する。 プログラムされたソース（イベント）を要求する。 コントロールソースをループアラウンドする。 トリガを設定するためのバス： バイパスをイネーブル（SOUR）またはディスエーブル（ACC）する。 プログラムされた方向を要求する。 非同期Trigger Linkモードを設定するためのバス： 入力ライン（1~6）を選択する。 プログラムされた入力ラインを要求する。 出力ライン（1~6）を選択する。 プログラムされた出力ラインを要求する。
:LAYer2 :IMMEDIATE :COUNt<n> INFinite :COUNt? :DELay<num> :DELay? :SOURce HOLD IMMEDIATE MANual BUS TLINk EXTernal TIMer :SOURce? :SIGNal :TIMer<num>	スキャンのスキャンレイヤをプログラムするためのバス： コントロールソースをループアラウンドする。 スキャン回数（1~9999またはINFinite）をプログラムする。 プログラムされたスキャンカウントを要求する。 デイレイ（0~99999.999秒）をプログラムする。 プログラムされたデイレイを要求する。 Arm Layer 2コントロールソースを選択する。 プログラムされたソース（イベント）を要求する。 スキャンコントロールソースを1回だけバイパスする。 タイマ時間間隔（0~99999.999秒）を設定する。

トリガコマンドサマリ

コマンド	説明
<pre> :TIMer? :TCONfigure :DIRection SOURce ACCeptor :DIRection? :ASYNchronous :ILINe<n> :ILINe? :OLINe<n> :OLINe? :TRIGger[:SEQuence[1]] :IMMediate :COUNt<n> INFinite :AUTo ON/OFF :COUNt? :DELAy<num> :DELAy? :SOURce HOLD IMMediate MANual BUS TLINk EXTernal TIMer :SOURce? :SIGNal :TIMer<num> :TIMer? :TCONfigure :PROTOcol ASYNchronous SSYNchronous :PROTOcol? :DIRection SOURce ACCeptor :DIRection? :ASYNchronous :ILINe<n> :ILINe? :OLINe<n> :OLINe? :SSYNchronous :LINe<n> :LIMe? </pre>	<p>プログラムされたタイマ時間間隔を要求する。</p> <p>Trigger Linkを設定するためのパス： バイパスをイネーブル (SOUR) またはディスエーブル (ACC) する。</p> <p>プログラムされた方向を要求する。</p> <p>非同期モードを設定するためのパス： 入力ライン (1~6) を選択する。 プログラムされた入力ラインを要求する。 出力ライン (1~6) を選択する。 プログラムされた出力ラインを要求する。</p> <p>スキヤンのチャンネルレイヤをプログラムするためのパス： チャンネルを即時にスキヤンする。(コントロールソースをループアラウンド)。 チャンネル数 (1~9999またはINFinite) をプログラムする。 scan-list-lengthを使う。</p> <p>プログラムされたチャンネルカウントを要求する。 ディレイ (0~99999.999秒) をプログラムする。 プログラムされたディレイを要求する。 Trigger Layerコントロールソースを選択する。</p> <p>プログラムされたソース (イベント) を要求する。 チャンネルコントロールソースを1回だけバイパスする。 タイマ時間間隔 (0~99999.999秒) を設定する。 プログラムされたタイマ時間間隔を要求する。 トリガを設定するためのパス： プロトコル (非同期または半同期Trigger Link) を選択する。</p> <p>プログラムされたプロトコルを要求する。 バイパスをイネーブル (SOUR) またはディスエーブル (ACC) する。</p> <p>プログラムされた方向を要求する。</p> <p>非同期Trigger Linkモードを設定するためのパス： 入力ライン (1~6) を選択する。 プログラムされた入力ラインを要求する。 出力ライン (1~6) を選択する。 プログラムされた出力ラインを要求する。</p> <p>半同期Trigger Linkモードを設定するためのパス： トリガライン (1~6) を選択する。 プログラムされたトリガラインを要求する。</p>

注：大文字はコマンド短縮形を表す。たとえば、":arm:layer2:source:immediate"を送信する代わりに、":arm:lay2:sour:imm"を送信することができる。

:INITiate

スキャナは、Model 7001がアイドル状態から抜け出すまでアームできません。このコマンドパスは、Model 7001をアイドル状態から抜け出させるためのものです。このパスのコマンドは以下の通りです。

[:IMMediate]

:INITiate[:IMMediate]

このコマンドを囲んでいる角括弧 “[] ” は、コマンドが暗黙に含まれるもので、送信する必要がないことを表わします。

フォーマット

:init

説明

このコマンドは、Model 7001をアイドル状態から抜け出させます。トリガ（スキャン）構造から出ると、Model 7001はアイドル状態に戻ります。

**:CONTInuous **

:INITiate:CONTInuous

パラメータ

b = 0またはOFF 連続起動をディスエーブル
b = 1またはONN 連続起動をイネーブル

フォーマット

:init:cont b

デフォルト

電源投入 セーブされているデフォルトを選択 (*SAV参照)
*RST 連続起動をディスエーブル
*CLS 無効
DCL、SDC 無効
:SYSTem:PRESet 連続起動をディスエーブル
:STATus:PRESet 無効

説明

連続起動を選択 (ON) すると、Model 7001はアイドル状態から抜け出し、シングルスキャンサイクルをプログラムされている通りに実行します。ただし、スキャンが完了しても、Model 7001はアイドル状態に戻らず、Arm Layer 1でアクティブ状態のままです。

プログラミング例

以下のプログラムステートメントは、:INITコマンドを使ってModel 7001をアイドル状態から抜け出させるものです。

```
OUTPUT 707;“:init”
```

:ABORt

フォーマット :abort

説明 連続起動がデイスエーブル (:init:cont off) されているとき、このコマンドはスキャンをアボートします。アボートされると、スキャンは速やかに終了し、Model 7001がアイドル状態に戻ります。

プログラミング例 以下のプログラムステートメントは、:ABORtコマンドを使ってModel 7001をアイドル状態に戻すものです。

```
OUTPUT 707;“:abort”
```

:ARM[:SEQuence[1]]

このコマンドパスは、スキャンのアームレイヤ (:LAYer1) およびスキャンレイヤ (LAYer2) を設定するためのものです。2番目のコマンドは角括弧 “ [] ” で囲まれています。これはコマンドが暗黙に含まれるもので、送信する必要がないことを表わします。このパスで使うコマンドは以下の通りです。

[:LAYer[1]]

```
:ARM[:SEQuence[1]][:LAYer[1]]
```

角括弧 “ [] ” は、このコマンドパスも暗黙に含まれるもので、送信する必要がないことを表わします。このパスの以下のコマンドは、スキャンのアームレイヤを設定するのに使います。

:IMMediate

```
:ARM[:SEQuence[1]][:LAYer[1]]:IMMediate
```

フォーマット :arm:imm

説明 このコマンドは、アームコントロールソースをバイパスするためのもので、プログラムされたイベントを待ちたくないときに使います。なお、このコマンドを送信するときは、Model 7001がアームイベントを待っている状態であればなりません。そうでないと、エラーが発生し、このコマンドは無視されます。

このコマンドを送信すると、アームコントロールソースはバイパスされます。そうすると、スキャンがアームされ、オペレーションはそのまま継続し、スキャンレイヤに入ります。

:IMMediateは機器セットアップコマンドではありません。このコマンドは、実行されると直ちにスキャンをアームしようとします。

:COUNT <n> | INFinite

:ARM[:SEQuence[1]][:LAYer[1]]:COUNT<n>|INFinite

パラメータ	n = 1~9999	有限アームカウント
	INF	無限アームカウント
フォーマット	:arm:coun n	
	:arm:coun INF	
デフォルト	電源投入	セーブされているデフォルトを選択 (*SAV参照)
	*RST	アームカウント = 1
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	アームカウント = 1
	:STATus:PRESet	無効

説明 このコマンドは、オペレーションがArm Layer 1に戻る回数をプログラムするのに使います。各スキャンサイクルの完了後、さらにアームを行なうようにプログラムされていれば、オペレーションはArm Layer 1までループバックします。アームカウントをINFiniteにプログラムした場合、オペレーションは連続的にArm Layer 1までループバックします。

:COUNT?

:ARM[:SEQuence[1]][:LAYer[1]]:COUNT?

フォーマット :arm:coun?

説明 この問合せコマンドは、プログラムされたアームカウントを要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、アームカウント値がコンピュータへ送られます。+9.9e37のアームカウント値は、アームカウントがINFiniteに設定されていることを表わします。

:SOURce HOLD | IMMEDIATE | MANual | BUS | TLINK | EXTERNAL

:ARM[:SEQuence[1]][:LAYer[1]]:SOURce HOLD|IMMEDIATE|MANual|BUS|TLINK|EXTERNAL

パラメータ	HOLD	オペレーションをアームレイヤでホールド
	IMM	即時アームを選択
	MAN	マニュアルアームイベントを選択
	BUS	アームイベントとしてGPIBトリガを選択
	TLINK	アームイベントとしてTrigger Linkを選択
	EXT	アームイベントとしてExternal Triggeringを選択
フォーマット	:arm:sour hold	
	:arm:sour man	
	:arm:sour imm	
	:arm:sour bus	

```
:arm:sour tlink
:arm:sour ext
```

デフォルト

電源投入	セーブされているデフォルトを選択 (*SAV参照)
*RST	ソース = IMMEDIATE
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	ソース = IMMEDIATE
:STATus:PRESet	無効

説明

このコマンドは、Arm Layer 1のコントロールソースを選択するのに使います。HOLDを選択した場合、オペレーションはArm Layer 1で停止し、無期限に待機します。HOLD状態のとき、前に説明したように:ARM:IMMコマンドを送信することによりスキャンを再開することができます。ただし、さらにアームを行なうようにプログラムされている場合は、次のスキャンサイクルの始めに (Arm Layer 1で)、再びHOLDが有効になります。

IMMを選択した場合 (:ARM:SOUR IMM=:ARM:IMMと混同しないでください)、オペレーションはそのまま直ちに次のレイヤへ進みます。

Arm Layer 1におけるオペレーションをコントロールするのに、特定のイベントを使うことができます。BUSを選択した場合、オペレーションは、 GPIBトリガ (GETまたは*TRG) が送信されると再開されます。TLINKを選択した場合、オペレーションは、Trigger Linkを介して入力トリガが受信されると再開されます。EXTを選択した場合、オペレーションは、外部トリガが受信されると再開されます。

:SOURce?

```
:ARM[:SEquence[1]][:LAYer[1]]:SOURce?
```

フォーマット

```
:arm:sour?
```

説明

この問合せコマンドは、プログラムされたアームコントロールソースを要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、ソースメッセージ (HOLD、MAN、IMM、BUS、TLINまたはEXT) がコンピュータへ送られます。

:SIGNal

```
:ARM[:SEquence[1]][:LAYer[1]]:SIGNal
```

フォーマット

```
:arm:sign
```

説明

このコマンドは、アームコントロールソースをバイパスするためのもので、プログラムされたアームイベントを待ちたくないときに使います。なお、このコマンドを送信するときは、Model 7001がアームイベントを待っている状態であればなりません。そうでないと、エラーが発生し、このコマンドは無視されます。

このコマンドを送信すると、アームコントロールソースはバイパスされます。そうすると、スキャンがアームされ、オペレーションはそのまま継続し、スキャンレイヤに入ります。

:SIGNalは機器セットアップコマンドではありません。このコマンドは、実行されると直ちにスキャンをアームしようとします。

:TCONfigure

:ARM[:SEquence[1]][:LAYer[1]]:TCONfigure

このコマンドパスは、スキャンのアームレイヤへのトリガを設定するためのものです。このパスで使うコマンドは以下の通りです。

:DIRection SOURce | ACCeptor

:ARM[:SEquence[1]][:LAYer[1]]:TCONfigure:DIRection SOURce|ACCeptor

パラメータ	SOUR	バイパスをイネーブル
	ACC	バイパスをディスエーブル
フォーマット	:arm:tcon:dir sour :arm:tcon:dir acc	
デフォルト	電源投入	セーブされているデフォルトを選択 (*SAV参照)
	*RST	Direction = Acceptor
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	Direction = Acceptor
	:STATus:PRESet	無効

説明

SOURを選択した場合、スキャンサイクルの始めに、アームレイヤを最初の通過時にバイパスします。プログラムされたイベントは必要ありません。Model 7001がアイドル状態から抜け出ると、直ちに、オペレーションはArm Layer 2に入ります。Model 7001はプログラムされたイベントを待ちません。このバイパスは、アームレイヤの最初の通過時にしか行なわれません。さらにアームを行なうようにプログラムされていれば、Model 7001は、プログラムされたアームイベントが発生するまでアームレイヤで待ちます。なお、SOURを選択すると、アームレイヤへの出力トリガがイネーブルされます。

ACCを選択した場合、バイパスはディスエーブルされます。つまり、オペレーションをArm Layer 2へ進ませるのに、プログラムされたアームイベントを必要とすることになります。なお、ACCを選択すると、アームレイヤへの出力トリガがイネーブルされます。

:DIRection?

:ARM[:SEquence[1]][:LAYer[1]]:TCONfigure:DIRection?

フォーマット :arm:tcon:dir?

説明

この問合せコマンドは、プログラムされた方向 (SOURceまたはACCeptor) を要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、方向メッセージ (SOURまたはACC) がコンピュータへ送られます。

:ASYNchronous

:ARM[:SEquence[1]][:LAYer[1]]:TCONfigure:ASYNchronous

このコマンドパスは、Trigger Link用として以下の非同期トリガラインを設定するためのものです。

:ILINe <n>

:ARM[:SEquence[1]][:LAYer[1]]:TCONfigure:ASYNchronous:ILINe<n>

パラメータ

n=1	入力ライン#1
n=2	入力ライン#2
n=3	入力ライン#3
n=4	入力ライン#4
n=5	入力ライン#5
n=6	入力ライン#6

フォーマット

:arm:tcon:asyn:ilin n

デフォルト

電源投入	セーブされているデフォルトを選択 (*SAV参照)
*RST	入力ライン = #1
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	入力ライン = #1
:STATus:PRESet	無効

説明

このコマンドは、6本のTrigger Linkラインのうちの1本を入力ラインとして選択するのに使います。なお、非同期Trigger Linkの入力と出力 (:OLINeコマンドを参照してください) は同じラインを共用できませんから、注意してください。

既に出力に使用されているラインに入力を指定すると、エラーが発生し、コマンドは無視されます。

:ILINe?

:ARM[:SEquence[1]][:LAYer[1]]:TCONfigure:ASYNchronous:ILINe?

フォーマット

:arm:tcon:asyn:ilin?

説明

この問合せコマンドは、どのTrigger Linkラインが入力トリガラインとして使用されているかを調べるのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、入力として指定されたライン番号 (1、2、3、4、5または6) がコンピュータへ送られます。

:OLINe <n>

:ARM[:SEquence[1]][:LAYer[1]]:TCONfigure:ASYNchronous:OLINe<n>

パラメータ

n=1 出力ライン#1

n=2 出力はライン#2
 n=3 出力はライン#3
 n=4 出力はライン#4
 n=5 出力はライン#5
 n=6 出力はライン#6

フォーマット :arm:tcon:asyn:olin n

デフォルト

電源投入	セーブされているデフォルトを選択 (*SAV参照)
*RST	出力ライン = #2
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	出力ライン = #2
:STATus:PRESet	無効

説明 このコマンドは、6本のTrigger Linkラインのうちの1本を出力ラインとして選択するのに使います。なお、非同期Trigger Linkの出力と入力 (:ILINeコマンドを参照してください) は同じラインを共用できませんから、注意してください。

既に出力に使用されているラインに出力を指定すると、エラーが発生し、コマンドは無視されます。

:OLINe?

:ARM[:SEQuence[1]][:LAYer[1]]:TCONfigure:ASYNchronous:OLINe?

フォーマット :arm:tcon:asyn:olin?

説明 この問合せコマンドは、どのTrigger Linkラインが出力トリガラインとして使用されているかを調べるのに使います。このコマンドを送信し、Model 7001をトークにアドレスッシングすると、出力として指定されたライン番号(1、2、3、4、5または6)がコンピュータへ送られます。現在、Trigger Link (TLIN) アームコントロールソースが選択されていない場合は、ライン番号ではなく、ゼロ(0)がコンピュータへ送られます。

プログラミング例

以下のプログラミング例は、:ARMサブシステムのLAYer1コマンドを使ってスキャンのアームレイヤをプログラムするものです。

```
10 OUTPUT 707;":arm:sour man;coun 2;tcon:dir sour"
20 OUTPUT 707;":arm:sour?;coun?;tcon:dir?"
30 ENTER 707;AS
40 PRINT AS
50 END
```

行10 この行の3つのコマンドのうち、:SOURCEはマニュアルアームコントロールソースを選択し、:COUNTは2のアームカウントを選択し、:DIRECTIONはソースバイパスをイネーブルする。

行20 この行の3つの問合せコマンドのうち、:SOURCE?はプログラムされたアームコントロールソースを要求し、:COUNT?はプログラムされたアームカウントを要求し、:DIRECTION?はソースバイパスのステータスを要求する。

行30 Model 7001をトークにアドレスッシングする。

行40 問合せメッセージを送信された順 (MAN、2、SOUR) に表示させる。

:LAYer2

:ARM[:SEQuence[1]]:LAYer2

このバスの以下のコマンドは、スキヤンのスキヤンレイヤを設定するのに使います。

:IMMediate

:ARM[:SEQuence[1]]:LAYer2:IMMediate

フォーマット

:arm:lay2:imm

説明

このコマンドは、スキヤンコントロールソースおよびスキヤンレイヤディレイ (:LAYer2:DELAy参照) をバイパスするためのもので、プログラムされたスキヤンイベントを待ちたくないときに使います。なお、このコマンドを送信するときは、Model 7001がスキヤンイベントを待っている状態であればなりません。そうでないと、エラーが発生し、このコマンドは無視されます。

このコマンドを送信すると、スキヤンコントロールソースおよびスキヤンディレイはバイパスされます。そうすると、スキヤンがイネーブルされ、オペレーションはそのまま継続し、チャンネルレイヤに入ります。

:IMMediateは機器セットアップコマンドではありません。このコマンドは、実行されると直ちにスキヤンをイネーブルしようとしています。

:COUNT <n> | INFinite

:ARM[:SEQuence[1]]:LAYer2:COUNT<n>|INFinite

パラメータ

n = 1~9999 有限スキヤンカウント
INF 無限スキヤンカウント

フォーマット

:arm:lay2:coun n
:arm:lay2:coun INF

デフォルト

電源投入 セーブされているデフォルトを選択 (*SAV参照)
*RST スキヤンカウント = 1
*CLS 無効
DCL、SDC 無効
:SYSTem:PRESet スキヤンカウント = INFinite
:STATus:PRESet 無効

説明

このコマンドは、スキヤンを繰り返すことができる回数をプログラムするのに使います。各チャンネルスキヤンの完了後、さらにアームを行なうようにプログラムされていれば、オペレーションはArm Layer 2までループバックします。スキヤンをINFiniteにプログラムした場合、オペレーションは連続的にArm Layer 2までループバックします。

:COUNT?

:ARM[:SEquence[1]]:LAYer2:COUNT?

フォーマット

:arm:lay2:coun?

説明

この問合せコマンドは、プログラムされたスキャンカウントを要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、スキャンカウント値がコンピュータへ送られます。+9.9e37のスキャンカウント値は、スキャンカウントがINFiniteに設定されていることを表わします。

:DElay <num>

:ARM[:SEquence[1]]:LAYer2:DElay<num>

パラメータ

num = 0~99999.999秒 デイレイ

フォーマット

:arm:lay2:del num

説明

このデイレイは、Arm Layer 2でオペレーションを遅延させるのに使います。スキャンイベントが検出されると、Model 7001は、DElayがタイムアウトになるのを待ってから、Trigger Layerへ進みます。このデイレイは、プログラムされた全てのスキャンに対して有効です。

:DElay?

:ARM[:SEquence[1]]:LAYer2:DElay?

フォーマット

:arm:lay2:del?

説明

この問合せコマンドは、Arm Layer 2の、プログラムされたDElayを要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、デイレイ (秒単位) がコンピュータへ送られます。

:SOURce HOLD | IMMEDIATE | MANual | BUS | TLINK | EXTernal

:ARM[:SEquence[1]]:LAYer2:SOURce HOLD|IMMEDIATE|MANual|BUS|TLINK|EXTernal

パラメータ

HOLD	Arm Layer 2でオペレーションをホールド
IMM	即時スキャンを選択
TIM	スキャンイベントとしてタイマを選択
MAN	マニュアルスキャンイベントを選択
BUS	スキャンイベントとしてGPIBトリガを選択
TLINK	スキャンイベントとしてTrigger Linkを選択
EXT	スキャンイベントとしてExternal Triggeringを選択

フォーマット	:arm:lay2:sour hold	
	:arm:lay2:sour man	
	:arm:lay2:sour imm	
	:arm:lay2:sour tim	
	:arm:lay2:sour bus	
	:arm:lay2:sour tlink	
	:arm:lay2:sour ext	
デフォルト	電源投入	セーブされているデフォルトを選択 (*SAV参照)
	*RST	ソース = IMMEDIATE
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	ソース = IMMEDIATE
	:STATus:PRESet	無効

説明 このコマンドは、Arm Layer 2のコントロールソースを選択するのに使います。HOLDを選択した場合、オペレーションはArm Layer 2で停止し、無期限に待機します。HOLD状態のとき、前に説明したように:ARM:LAY2:IMMコマンドを送信することによりスキャンを再開することができます。ただし、さらにアームを行なうようにプログラムされている場合は、次のスキャンサイクルの始めに (Arm Layer 2で)、再びHOLDが有効になります。

IMMを選択した場合 (:ARM:LAY2:SOUR IMMを:ARM:LAY2:IMMと混同しないでください)、スキャンはそのまま直ちに次のレイヤへ進みます。

スキャンをコントロールするのに、特定のイベントを使うことができます。TIMを選択した場合、タイマ時間間隔の始めと、それがタイムアウトになるごとに、スキャンイベントが発生します。たとえば、タイマを30秒の時間間隔にプログラムした場合、最初のスキャンイベントは即時に発生し、それ以後、30秒置きにスキャンイベントが発生します。タイマの時間間隔は、:LAYer2:TIMerコマンドを使って設定します。

BUSを選択した場合、オペレーションは、 GPIBトリガ (GETまたは*TRG) が送信されると再開されます。TLINKを選択した場合、オペレーションは、Trigger Linkを介して入力トリガが受信されると再開されます。EXTを選択した場合、オペレーションは、外部トリガが受信されると再開されます。

:SOURce?

:ARM[:SEquence[1]]:LAYer2:SOURce?

フォーマット :arm:lay2:sour?

説明 この問合せコマンドは、プログラムされたスキャンコントロールソースを要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、ソースメッセージ (HOLD、MAN、IMM、TIM、BUS、TLINまたはEXT) がコンピュータへ送られます。

:SIGNal

:ARM[:SEquence[1]]:LAYer2:SIGNal

フォーマット :arm:lay2:sign

説明

このコマンドは、スキャンコントロールソースをバイパスするためのもので、プログラムされたスキャンイベントを待ちたくないときに使います。ただし、ディレイ (:LAYer2:DELAy参照) はバイパスされません。このコマンドを送信するときは、Model 7001がスキャンイベントを待っている状態であればなりません。そうでないと、エラーが発生し、このコマンドは無視されます。

このコマンドを送信すると、スキャンコントロールソースはバイパスされます。スキャンディレイがプログラムされている場合、Model 7001は、ディレイがタイムアウトになるのを待ってから、チャンネルレイヤへ進みます。

:SIGNalは機器セットアップコマンドではありません。このコマンドは、実行されると直ちにコントロールソースをループアラウンドしようとします。

:TiMer <num>

:ARM[:SEQuence[1]]:LAYer2:TiMer<num>

パラメータ

num = 1~99999.999 秒単位の時間間隔

フォーマット

:arm:lay2:tim num

デフォルト

電源投入	セーブされているデフォルトを選択 (*SAV参照)
*RST	時間間隔 = 0.001秒
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	時間間隔 = 0.001秒
:STATus:PRESet	無効

説明

このコマンドは、Arm Layer 2のタイマの時間間隔を設定します。なお、タイマが有効なのは、コントロールソースとしてタイマを選択した場合だけです (:ARM:LAY2:SOURceを参照してください)。

:TiMer?

:ARM[:SEQuence[1]]:LAYer2:TiMer?

フォーマット

:arm:lay2:tim?

説明

この問合せコマンドは、Arm Layer 2のタイマの、プログラムされた時間間隔を要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、時間間隔 (秒単位) がコンピュータへ送られます。

:TCONfigure

:ARM[:SEquence[1]]:LAYer2:TCONfigure

このコマンドパスは、Arm Layer 2へのトリガを設定するためのものです。このパスで使うコマンドは以下の通りです。

:DIRection SOURce | ACCeptor

:ARM[:SEquence[1]]:LAYer2:TCONfigure:DIRection SOURce|ACCeptor

パラメータ	SOUR	バイパスをイネーブル
	ACC	バイパスをディスエーブル
フォーマット	:arm:lay2:tcon:dir sour	
	:arm:lay2:tcon:dir acc	
デフォルト	電源投入	セーブされているデフォルトを選択 (*SAV参照)
	*RST	Direction = Acceptor
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	Direction = Acceptor
	:STATus:PRESet	無効
説明	<p>SOURを選択した場合、スキャンサイクルの始めに、Arm Layer 2を最初の通過時にバイパスします。スキャンは、プログラムされたイベントを待ちません。このバイパスは、Arm Layer 2の最初の通過時にしか行なわれません。さらにスキャンを行なうようにプログラムされていれば、Model 7001は、プログラムされたスキャンイベントが発生するまでArm Layer 2で待ちます。なお、SOURを選択すると、Arm Layer 2への出力トリガがイネーブルされます。</p> <p>ACCを選択した場合、バイパスはディスエーブルされます。つまり、オペレーションをTrigger Layerへ進ませるのに、プログラムされたスキャンイベントを必要とすることになります。なお、ACCを選択すると、Arm Layer 2への出力トリガが抑制されます。</p>	

:DIRection?

:ARM[:SEquence[1]]:LAYer2:TCONfigure:DIRection?

フォーマット :arm:lay2:tcon:dir?

説明 この問合せコマンドは、プログラムされた方向 (SOURceまたはACCeptor) を要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、方向メッセージ (SOURまたはACC) がコンピュータへ送られます。

:ASYNchronous

:ARM[:SEQuence[1]]:LAYer2:TCONfigure:ASYNchronous

このコマンドパスは、Trigger Link用として以下の非同期トリガラインを設定するためのものです。

:ILINe <n>

:ARM[:SEQuence[1]]:LAYer2:TCONfigure:ASYNchronous:ILINe<n>

パラメータ

n=1 入力ライン#1
 n=2 入力ライン#2
 n=3 入力ライン#3
 n=4 入力ライン#4
 n=5 入力ライン#5
 n=6 入力ライン#6

フォーマット

:arm:lay2:tcon:asyn:ilin n

デフォルト

電源投入	セーブされているデフォルトを選択 (*SAV参照)
*RST	入力ライン = #1
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	入力ライン = #1
:STATus:PRESet	無効

説明

このコマンドは、6本のTrigger Linkラインのうちの1本を入力ラインとして選択するのに使います。なお、非同期Trigger Linkの入力と出力 (:OLINeコマンドを参照してください) は同じラインを共用できませんから、注意してください。

既に出力に使用されているラインに入力を指定すると、エラーが発生し、コマンドは無視されます。

:ILINe?

:ARM[:SEQuence[1]]:LAYer2:TCONfigure:ASYNchronous:ILINe?

フォーマット

:arm:lay2:tcon:asyn:ilin?

説明

この問合せコマンドは、どのTrigger Linkラインが入力トリガラインとして使用されているかを調べるのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、入力として指定されたライン番号 (1、2、3、4、5または6) がコンピュータへ送られます。なお、現在、Trigger Link (TLIN) アームコントロールソースが選択されていない場合は、ライン番号ではなくゼロ (0) がコンピュータへ送られます。

:OLINe <n>

:ARM[:SEquence[1]]:LAYer2:TCONfigure:ASYNchronous:OLINe<n>

パラメータ	n=1	出力はライン#1
	n=2	出力はライン#2
	n=3	出力はライン#3
	n=4	出力はライン#4
	n=5	出力はライン#5
	n=6	出力はライン#6

フォーマット :arm:lay2:tcon:asyn:olin n

デフォルト	電源投入	セーブされているデフォルトを選択 (*SAV参照)
	*RST	出力ライン = #2
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	出力ライン = #2
	:STATus:PRESet	無効

説明 このコマンドは、6本のTrigger Linkラインのうちの1本を出力ラインとして選択するのに使います。なお、非同期Trigger Linkの出力と入力 (:ILINeコマンドを参照してください) は同じラインを共用できませんから、注意してください。

既に出力に使用されているラインに出力を指定すると、エラーが発生し、コマンドは無視されます。

:OLINe?

:ARM[:SEquence[1]]:LAYer2:TCONfigure:ASYNchronous:OLINe?

フォーマット :arm:lay2:tcon:asyn:olin?

説明 この問合せコマンドは、どのTrigger Linkラインが出力トリガラインとして使用されているかを調べるのに使います。このコマンドを送信し、Model 7001をトークにアドレスリングすると、出力として指定されたライン番号 (1、2、3、4、5または6) がコンピュータへ送られます。

プログラミング例

以下のプログラミング例は、:ARMサブシステムのLAYer2コマンドを使ってスキヤンのスキヤンレイヤをプログラムするものです。

```
10 OUTPUT 707;":arm:lay2:sour tim;tim 20;coun 1;tcon:dir sour"
20 OUTPUT 707;":arm:lay2:sour?;tim?;coun?;tcon:dir?"
30 ENTER 707;AS
40 PRINT AS
50 END
```

行10 この行の4つのコマンドのうち、:SOURCEはスキヤンコントロールソースとしてタイマを選択し、TIMERはタイマを20秒に設定し、COUNTは1のスキヤンカウントを選択し、DIRECTIONはソースバイパスをイネーブルする。

- 行20 この行の4つの間合せコマンドのうち、:SOURce?はプログラムされたスキャンコントロールソースを要求し、TIMer?はタイマ時間間隔を要求し、:COUNT?はプログラムされたスキャンカウントを要求し、:DIRection?はソースバイパスのステータスを要求する。
- 行30 Model 7001をトークにアドレッシングする。
- 行40 間合せメッセージを送信された順(MAN、20.000、1、SOUR)に表示させる。

:TRIGger[:SEQuence[1]]

:SEQuence1を囲んでいる角括弧“[]”は、コマンドが暗黙に含まれるもので、送信する必要がないことを表わします。このパスの以下のコマンドは、スキャンのチャンネルレイヤ (:TRIGger)を設定するのに使います。

:IMMediate

TRIGger[:SEQuence[1]]:IMMediate

フォーマット

:trig:imm

説明

このコマンドは、チャンネルコントロールソースおよびTrigger Layerダイレイ (:TRIGger:DELAy参照)をバイパスするためのもので、プログラムされたチャンネルイベントを待ちたくないときに使います。なお、このコマンドを送信するときは、Model 7001がチャンネルイベントを待っている状態でなければなりません。そうでないと、エラーが発生し、このコマンドは無視されます。

このコマンドを送信すると、コントロールソースおよびダイレイがバイパスされ、次のチャンネルが閉じます。

:IMMediateは機器セットアップコマンドではありません。このコマンドは、実行されると直ちに次のチャンネルを閉じようとしています。

:COUNT <n> | INFinite

:TRIGger[:SEQuence[1]]:COUNT<n>|INFinite

アメータ

n = 1~9999 有限チャンネルカウント
INF 無限チャンネルカウント

フォーマット

:trig:coun n
:trig:coun INF

デフォルト

電源投入	セーブされているデフォルトを選択 (*SAV参照)
*RST	チャンネルカウント = 1
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	チャンネルカウント = scan-list-length
:STATus:PRESet	無効

説明

このコマンドは、スキャンするチャンネル数をプログラムするのに使います。通常は、チャンネル数 (n) は、スキャンリスト中のチャンネル数と同数にプログラムしてください。チャンネルカウントとしてscan-list-lengthを使う場合、スキャンリスト中の各チャンネルは、プログラムされたスキャンごとに1回閉じます。scan-list-lengthは、:ROUTe:SCAN:POINis?問合せコマンドを使って調べることができます (ROUTeサブシステムを参照してください)。

scan-list-lengthより小さいチャンネルカウントを選択した場合、最初のスキャンは、その数のチャンネルがスキャンされると終了します。さらにスキャンを行なうようにプログラムされていれば、最後にスキャンされたチャンネルからスキャンが継続されません。

scan-list-lengthより大きい有限チャンネルカウントを選択した場合、最初のスキャンで、スキャンリスト中の全チャンネルがスキャンされ、スキャンリストの先頭へループバックします。選択した数のチャンネルカウントに達するまでスキャンが繰り返されます。さらにスキャンを行なうようにプログラムされていれば、最後にスキャンされたチャンネルからスキャンが継続されます。

INFiniteチャンネルカウントを選択した場合、リスト中の全チャンネルがスキャンされるとスキャンリストの先頭へループバックします。このプロセスが無期限に繰り返されます。

:AUTO ON/OFF

:TRIGger[:SEQuence[1]]:COUNT<n>|INFinite:AUTO ON/OFF

パラメータ

ON チャンネルカウントとしてscan-list-lengthを使用
OFF チャンネルカウントとして<n> (1~9999) またはINF (無限) を使用

フォーマット

:trig:coun:auto on
:trig:coun:auto off

デフォルト

電源投入	セーブされている電源投入時デフォルト
*RST	OFF
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	ON
:STATus:PRESet	無効

説明

コマンド:trig:coun:auto onでは、チャンネルカウントをscan-list-lengthに設定します。コマンド:trig:coun:auto offを送信すると、チャンネルカウントは:trig:counのパラメータ (1~9999または無限) に設定されます。

:COUNT?

:TRIGger[:SEQuence[1]]:COUNT?

フォーマット

:trig:coun?

説明

この問合せコマンドは、プログラムされたチャンネルカウントを要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、チャンネルカウント値がコンピュータへ送られます。+9.9e37のチャンネルカウント値は、チャンネルカウントがINFiniteに設定されていることを表わします。

:DElay <num>

:TRIGger[:SEQuence[1]]:DElay<num>

パラメータ

num = 0~99999.999秒 デイレイ

フォーマット

:trig:del num

説明

このデイレイは、スキャン時の各チャンネルの閉鎖を遅延させるのに使います。チャンネルコントロールソースが発生すると、Model 7001は、DElayがタイムアウトになるのを待ってから、次のチャンネルを閉じます。このデイレイは、スキャンされる全チャンネルに対して有効です。

:DElay?

:TRIGger[:SEQuence[1]]:DElay?

フォーマット

:trig:del?

説明

この問合せコマンドは、Trigger Layerの、プログラムされたDElayを要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、デイレイ（秒単位）がコンピュータへ送られます。

:SOURce HOLD | IMMEDIATE | MANUAL | BUS | TLINK | EXTERNAL

:TRIGger[:SEQuence[1]]:SOURce HOLD|IMMEDIATE|MANUAL|BUS|TLINK|EXTERNAL

パラメータ

HOLD Trigger Layerでオペレーションをホールド
 IMM 即時チャンネルスキャンを選択
 TIM チャンネルイベントとしてタイマを選択
 MAN マニュアルチャンネルイベントを選択
 BUS チャンネルイベントとしてGPIBトリガを選択
 TLINK スチャンネルイベントとしてTrigger Linkを選択
 EXT チャンネルスキャンイベントとしてExternal Triggeringを選択

フォーマット

:trig:sour hold
 :trig:sour man
 :trig:sour imm
 :trig:sour tim
 :trig:sour bus
 :trig:sour tlink
 :trig:sour ext

デフォルト	電源投入	セーブされているデフォルトを選択 (*SAV参照)
	*RST	ソース = IMMEDIATE
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	ソース = MANual
	:STATus:PRESet	無効

説明 このコマンドは、チャンネルスキャンをコントロールするためのソースを選択するのに使います。HOLDを選択した場合、オペレーションはTrigger Layerで停止し、無期限に待機します。HOLD状態のとき、前に説明したように:TRIG:IMMコマンドを送信することによりチャンネルスキャンを再開することができます。なお、HOLDは次のチャンネルに対して再び有効になります。

IMMを選択した場合 (:TRIG:SOUR IMMをTRIG:IMMと混同しないでください)、各チャンネルは即時にスキャンされ、特定のイベントを待ちません。

チャンネルスキャンをコントロールするのに、特定のイベントを使うことができます。TIMを選択した場合、最初のチャンネルは即時にスキャンされ、その後、タイマがタイムアウトになるごとにチャンネルがスキャンされます。タイマの時間間隔は、:TRIGger:TIMerコマンドを使って設定します。

BUSを選択した場合、各チャンネルは、GPIBトリガ (GETまたは*TRG) が送信されるとスキャンされます。TLINKを選択した場合、各チャンネルは、Trigger Linkを介して入力トリガが受信されるとスキャンされます。EXTを選択した場合、各チャンネルは、外部トリガが受信されるとスキャンされます。

:SOURce?

:TRIGger[:SEquence[1]]:SOURce?

フォーマット

:trig:sour?

説明

この問合せコマンドは、プログラムされたチャンネルコントロールソースを要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、ソースメッセージ (HOLD、MAN、IMM、TIM、BUS、TLINまたはEXT) がコンピュータへ送られます。

:SIGNal

:TRIGger[:SEquence[1]]:SIGNal

フォーマット

:trig:sign

説明

このコマンドは、チャンネルコントロールソースをバイパスするためのもので、プログラムされたチャンネルイベントを待ちたくないときに使います。ただし、Trigger Layerダイレイ (:TRIGger:DELay参照) はバイパスされません。このコマンドを送信するときは、Model 7001がチャンネルイベントを待っている状態であればなりません。そうでないと、エラーが発生し、このコマンドは無視されます。

このコマンドを送信すると、チャンネルコントロールソースはバイパスされます。チャンネルダイレイがプログラムされている場合、Model 7001は、ダイレイがタイムアウトになるのを待ってから、次のチャンネルを閉じます。

注:SIGNalは機器セットアップコマンドではありません。このコマンドは、実行されると直ちに次のチャンネルを閉じようとしています。

:TImEr <num>

:TRIGger[:SEQuence[1]]:TImEr<num>

パラメータ	num = 1~99999.999 秒単位の時間間隔	
フォーマット	:trig:tim num	
デフォルト	電源投入	セーブされているデフォルトを選択 (*SAV参照)
	*RST	時間間隔 = 0.001秒
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	時間間隔 = 0.001秒
	:STATus:PRESet	無効

説明 このコマンドは、Trigger Layerのタイマの時間間隔を設定します。なお、タイマが有効なのは、コントロールソースとしてタイマを選択した場合だけです (:TRIGger:TImErを参照してください)。

:TImEr?

:TRIGger[:SEQuence[1]]:TImEr?

フォーマット :trig:tim?

説明 この問合せコマンドは、Trigger Layerタイマの、プログラムされた時間間隔を要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、時間間隔 (秒単位) がコンピュータへ送られます。

:TCONfigure

:TRIGger[:SEQuence[1]]:TCONfigure

このコマンドパスはトリガを設定するためのものです。このパスで使うコマンドは以下の通りです。

:PROTOcol ASYNchronous | SSYNchronous

:TRIGger[:SEQuence[1]]:TCONfigure:PROTOcol ASYNchronous|SSYNchronous

パラメータ	ASYNchronous	非同期Trigger Linkモード
	SSYNchronous	半同期Trigger Linkモード

フォーマット	:trig:tcon:prot asyn :trig:tcon:prot ssyn	
デフォルト	電源投入	セーブされているデフォルトを選択 (*SAV参照)
	*RST	Protocol = Asynchronous
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	Protocol = Asynchronous
	:STATus:PRESet	無効
説明	このコマンドは、Trigger Linkのプロトコルを選択するのに使います。非同期の場合、入力トリガと出力トリガに別々のトリガラインを使います。半同期の場合、入力トリガと出力トリガの両方に1本のトリガラインを使います。Trigger Linkの詳細は4.11項を参照してください。	

:PROTOcol?

:TRIGger[:SEQuence[1]]:TCOnfigure:PROTOcol?

フォーマット :trig:tcon:prot?

説明 この問合せコマンドは、プログラムされたTrigger Linkプロトコルを要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、プロトコルメッセージ (ASYNまたはSSYN) がコンピュータへ送られます。

:DIRection SOURce | ACCeptor

:TRIGger[:SEQuence[1]]:TCOnfigure:DIRection SOURcelACCeptor

パラメータ
 SOUR バイパスをネーブル
 ACC バイパスをディスエーブル

フォーマット
 :trig:tcon:dir sour
 :trig:tcon:dir acc

デフォルト
 電源投入 セーブされているデフォルトを選択 (*SAV参照)
 *RST Direction = Acceptor
 *CLS 無効
 DCL、SDC 無効
 :SYSTem:PRESet Direction = Source
 :STATus:PRESet 無効

説明 SOURを選択した場合、プログラムされたチャンネルソースイベントを最初にバイパスする必要があります。つまり、スキャンがアームされ、イネーブルされると、スキャンリスト中の最初のチャンネルは直ちに閉じます。このバイパスは、スキャンされる最初のチャンネルに対してのみ行なわれます。以後のチャンネルは、プログラムされたチャンネルイベントによってコントロールされます。

ACCを選択した場合、バイパスはディスエーブルされます。つまり、チャンネルスキャンは、プログラムされたチャンネルイベントによってコントロールされることとなります。

:DIRection?

:TRIGger[:SEQuence{1}]:TCONfigure:DIRection?

フォーマット

:trig:tcon:dir?

説明

この問合せコマンドは、プログラムされた方向 (SOURCEまたはACCEPTOR) を要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、方向メッセージ (SOURCEまたはACC) がコンピュータへ送られます。

:ASYNchronous

:TRIGger[:SEQuence{1}]:TCONfigure:ASYNchronous

このコマンドは、Trigger Link用として以下の非同期トリガラインを設定するためのものです。

:ILINe <n>

:TRIGger[:SEQuence{1}]:TCONfigure:ASYNchronous:ILINe<n>

パラメータ

n=1	入力ライン#1
n=2	入力ライン#2
n=3	入力ライン#3
n=4	入力ライン#4
n=5	入力ライン#5
n=6	入力ライン#6

フォーマット

:trig:tcon:asyn:ilin n

デフォルト

電源投入	セーブされているデフォルトを選択 (*SAV参照)
*RST	入力ライン = #1
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	入力ライン = #1
:STATus:PRESet	無効

説明

このコマンドは、6本のTrigger Linkラインのうちの1本を入力ラインとして選択するのに使います。なお、非同期Trigger Linkの入力と出力 (:OLINeコマンドを参照してください) は同じラインを共用できませんから、注意してください。

既に出力に使用されているラインに入力を指定すると、エラーが発生し、コマンドは無視されます。

:ILINe?

:TRIGger[:SEQuence[1]]:TCONfigure:ASYNchronous:ILINe?

フォーマット :trig:tcon:asyn:ilin?

説明 この問合せコマンドは、どのTrigger Linkラインが入力トリガラインとして使用されているかを調べるのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、入力として指定されたライン番号（1、2、3、4、5または6）がコンピュータへ送られます。

:OLINe <n>

:TRIGger[:SEQuence[1]]:TCONfigure:ASYNchronous:OLINe<n>

パラメータ	n=1	出力はライン#1
	n=2	出力はライン#2
	n=3	出力はライン#3
	n=4	出力はライン#4
	n=5	出力はライン#5
	n=6	出力はライン#6

フォーマット :trig:tcon:asyn:olin n

デフォルト	電源投入	セーブされているデフォルトを選択 (*SAV参照)
	*RST	出力ライン = #2
	*CLS	無効
	DCL、SDC	無効
	:SYSTem:PRESet	出力ライン = #2
	:STATus:PRESet	無効

説明 このコマンドは、6本のTrigger Linkラインのうちの1本を出力ラインとして選択するのに使います。なお、非同期Trigger Linkの出力と入力（:ILINeコマンドを参照してください）は同じラインを共用できませんから、注意してください。

既に出力に使用されているラインに出力を指定すると、エラーが発生し、コマンドは無視されます。

:OLINe?

:TRIGger[:SEQuence[1]]:TCONfigure:ASYNchronous:OLINe?

フォーマット :trig:tcon:asyn:olin?

説明 この問合せコマンドは、どのTrigger Linkラインが出力トリガラインとして使用されているかを調べるのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、出力として指定されたライン番号（1、2、3、4、5または6）がコンピュータへ送られます。

:SSYNchronous

:TRIGger[:SEQuence[1]]:TCONfigure:SSYNchronous

このコマンドパスは、半同期Trigger Linkモードのトリガラインを選択するためのものです。

:ILINe <n>

:TRIGger[:SEQuence[1]]:TCONfigure:SSYNchronous:ILINe<n>

パラメータ

n=1	トリガライン#1
n=2	トリガライン#2
n=3	トリガライン#3
n=4	トリガライン#4
n=5	トリガライン#5
n=6	トリガライン#6

フォーマット

:trig:tcon:ssyn:lin n

デフォルト

電源投入	セーブされているデフォルトを選択 (*SAV参照)
*RST	トリガライン = #1
*CLS	無効
DCL、SDC	無効
:SYSTem:PRESet	トリガライン = #1
:STATus:PRESet	無効

説明

このコマンドは、半同期Trigger Link用として6本のトリガラインのうちの1本を選択するのに使います。

:ILINe?

:TRIGger[:SEQuence[1]]:TCONfigure:SSYNchronous:ILINe?

フォーマット

:trig:tcon:ssyn:line?

説明

この問合せコマンドは、半同期Trigger Linkモードに選択されたライン番号を要求するのに使います。このコマンドを送信し、Model 7001をトークにアドレッシングすると、トリガライン番号がコンピュータへ送られます。

プログラミング例

以下のプログラミング例は、:TRIGgerサブシステムのコマンドを使ってスキャンのチャンネルレイヤをプログラムするものです。

```

10 OUTPUT 707;":trig:sour tim;tim 0.5;coun 15"
20 OUTPUT 707;":trig:sour?;tim?coun?"
30 ENTER 707;AS
40 PRINT AS
50 END

```


- 行10 この行の3つのコマンドのうち、:SOURceはチャンネルコントロールソースとしてタイマを選択し、TIMerはタイマを0.5秒に設定し、COUNTは15のチャンネルカウントを選択する。
- 行20 この行の3つの問合せコマンドのうち、:SOURce?はプログラムされたチャンネルコントロールソースを要求し、TIMer?はタイマ時間間隔を要求し、:COUNT?はプログラムされたチャンネルカウントを要求する。
- 行30 Model 7001をトークにアドレッシングする。
- 行40 問合せメッセージを送信された順 (TIM、0.500、15) に表示させる。

付録

付録

付録A — IEEE-488.2コモンコマンド

付録B — SCPIコマンドサブシステム

付録C — インタフェースファンクションコード

付録D — ASCII文字コードおよびIEEE-488マルチラインインタフェースコマンドメッセージ

付録E — コントローラプログラム

付録F — メニュー構造

付録G — IEEE-488バスの概要

付録H — IEEE-488への適合について

付録I — SCPIへの適合について

付録A

IEEE-488.2コモンコマンドおよび問合せ

表A-1

IEEE-488.2コモンコマンドおよび問合せ

ニモニク	名前	説明
*CLS	Clear status	全イベントレジスタおよびエラー待ち行列をクリアする。
*ESE<n>	Event status enable command	標準イベントステータスイネーブルレジスタの内容をセットする。
*ESE?	Event status enable query	標準イベントステータスイネーブルレジスタのプログラム値を要求する。
*ESR?	Event status register query	標準イベントステータスレジスタのプログラム値を要求し、それをクリアする。
*IDN?	Identification query	Model 7001のメーカー名、型番、シリアルナンバーおよびファームウェア改訂水準を要求する。
*OPC	Operation complete command	保留中の全コマンドが実行された後、標準イベントステータスレジスタのOperation Completeビットをセットする。
*OPC?	Operation complete query	選択された保留中の全装置オペレーションが実行された後、ASCII "1" を出力待ち行列に入れる。
*OPT?	Option identification query	各スロットごとに指定された型番を要求する。
*RCL<n>	Recall command	Model 7001を、指定されたメモリロケーションにストアされているセットアップコンフィギュレーションに戻す。
*RST	Reset command	Model 7001を*RSTデフォルトに戻す。
*SAV<n>	Save command	セットアップコンフィギュレーションを指定されたメモリロケーションにセーブする。

表A-1 (続き)

IEEE-488.2 コモンコマンドおよび問合せ

ニモニツク	名前	説明
*SRE<n>	Service request enable command	サービス要求イネーブルレジスタの内容をセットする。
*SRE?	Service request enable query	Model 7001がサービス要求イネーブルレジスタの値に戻る。
*STB?	Read status byte query	ステータスバイトレジスタの値を返す。
*TRG	Trigger command	このコマンドは、グループ実行トリガ (GET) コマンドと同じ効果をもつバストリガを発生させる。
*TST?	Self-test query	この問合せが送信されると、Model 7001がROMのチェックサムテストを実行し、結果を返す。
*WAI	Wait-to-continue command	前の全コマンドが実行されるまで待つ。

B

SCPIコモンコマンドサブシステム

表B-1
DISPlayコマンドサマリ

コマンド	説明
:DISPlay	
[:WINDow[1]] :TEXT :DATA<a> :STATe :STATe?	メッセージをディスプレイの上部に位置指定するためのパス： ユーザテキストメッセージを制御するためのパス： 最大20文字を使ってASCIIメッセージ“a”を定義する。 メッセージをイネーブル（1またはON）あるいはディスエーブル（0またはOFF）する。 メッセージステータス（イネーブル状態またはディスエーブル状態）を要求する。
:WINDow2 :TEXT :DATA<a> :STATe :STATe?	メッセージをディスプレイの下部に位置指定するためのパス： ユーザテキストメッセージを制御するためのパス： 最大35文字を使ってASCIIメッセージ“a”を定義する。 メッセージをイネーブル（1またはON）あるいはディスエーブル（0またはOFF）する。 メッセージステータス（イネーブル状態またはディスエーブル状態）を要求する。
:SMESsage :SMESsage?	表示されたステータスメッセージをイネーブル（1またはON）あるいはディスエーブル（0またはOFF）する。 ステータスメッセージディスプレイの状態を要求す

表B-2
OUTPutコマンドサマリ

コマンド	説明
:OUTPut :TTLn :LSEnse AHIGH ALLOW :LSEnse?	出力ライン“n”（1～4）を指定するためのパスおよびコマンド： 極性、すなわち、アクティブハイ(AHIGH)またはアクティブロー(ALLOW)を設定する。 指定された出力ライン（n）を要求する。

表B-3
ROUTEコマンドサマリ

コマンド	説明
[:ROUTE]	
:CLOSe<list> :STATe?	指定されたチャンネルを閉じるためのパスおよびコマンド： 閉じているチャンネルを要求する。
:OPEN<list> ALL	指定された（または全部の）チャンネルを開く。
:SCAN<list> :POINts?	チャンネルリストを指定するためのパスおよびコマンド： スキャンリスト中のチャンネルの数を要求する。
:SCAN?	プログラムされたスキャンリストを要求する。
:FCHannels<list>	閉じてはならないチャンネル（禁止チャンネル）を指定する。
:FCHannels?	制限チャンネルのリストを要求する。
:CONFigure:	コンフィギュレーションコマンドパス：
:SCHannel	Single Channelをイネーブル(1またはON)あるいはディスエーブル(0またはOFF)する。
:SCHannel?	Single Channelモードのステータスを要求する。
:CPAir	Channel Pairingをイネーブル(1またはON)あるいはディスエーブル(0またはOFF)する。
:CPAir?	Channel Paringのステータスを要求する。
:SLOT1	CARD 1を設定するためのパス：
:CTYPE<type>	スロットに装着されたカードタイプを指定する。
:CTYPE?	スロットに装着されたカードタイプを要求する。
:POLE<n>	ポールモード（1、2または4）を選択する。
:POLE?	プログラムされたポール数を要求する。
:STIme<num>	秒単位のリレーセットリング時間を（0～99999.999）を設定する。
:STIme?	プログラムされたりレーセットリング時間を要求する。
:SLOT2	CARD 2を設定するためのパス：
:CTYPE<type>	スロットに装着されたカードタイプを指定する。
:CTYPE?	スロットに装着されたカードタイプを要求する。
:POLE<n>	ポールモード（1、2または4）を選択する。
:POLE?	プログラムされたポール数を要求する。
:STIme<num>	秒単位のリレーセットリング時間を（0～99999.999）を設定する。
:STIme?	プログラムされたりレーセットリング時間を要求する。
:MEMory	メモリをプログラムするためのパス：
:SAVe M<num>	チャンネルパターンをメモリロケーション（1～100）にセーブする。
:RECall M<num>	メモリロケーション（1～100）からチャンネルパターンをリコールする。

表B-4
SENSeコマンドサマリ

コマンド	説明
:SENSe[1]	
:TTL1 :DATA?	入力ポートのためのコマンドパス。 デジタル入力ポートを読み取る。

表B-5
SOURceコマンドサマリ

コマンド	説明
:SOURce	デジタル出力ポートのビットをプログラムするためのコマンドパス：
:TTL1 [:LEVel] [:LEVel]?	出力ポートのビット1をプログラムするためのパス： セット（1またはON）あるいはリセット（0またはOFF）。 ビット1のステータスを要求する。
:TTL2 [:LEVel] [:LEVel]?	出力ポートのビット2をプログラムするためのパス： セット（1またはON）あるいはリセット（0またはOFF）。 ビット2のステータスを要求する。
:TTL3 [:LEVel] [:LEVel]?	出力ポートのビット3をプログラムするためのパス： セット（1またはON）あるいはリセット（0またはOFF）。 ビット3のステータスを要求する。
:TTL4 [:LEVel] [:LEVel]?	出力ポートのビット4をプログラムするためのパス： セット（1またはON）あるいはリセット（0またはOFF）。 ビット4のステータスを要求する。

表B-6
STATusコマンドサマリ

コマンド	説明
<pre> :STATus :OPERation [:EVENTi]? :CONDition? :PTRansition<n> :PTRansition? :NTRansition<n> :NTRansition? :ENABle<n> :ENABle? :ARM [:EVENTi]? :CONDition? :PTRansition<n> :PTRansition? :NTRansition<n> :NTRansition? :ENABle<n> :ENABle? :SEquence [:EVEVt]? :CONDition? :PTRansition<n> :PTRansition? :NTRansition<n> :NTRansition? :ENABle<n> :ENABle? :TRIGger [:EVENTi]? :CONDition? :PTRansition<n> :PTRansition? :NTRansition<n> :NTRansition? :ENABle<n> :ENABle? :PRESet :QUEue [:NEXT]? :ENABle<list> </pre>	<pre> オペレーションイベントレジスタをコントロールするためのパス： ステータスレジスタを読み取る。 条件レジスタを読み取る。 正トランシジョンレジスタをプログラムする (0~1122)。 正トランシジョンレジスタを読み取る。 負トランシジョンレジスタをプログラムする (0~1122)。 負トランシジョンレジスタを読み取る。 イネーブルレジスタをプログラムする (0~1122)。 イネーブルレジスタを読み取る。 アームイベントレジスタをコントロールするためのパス： ステータスレジスタを読み取る。 条件レジスタを読み取る。 正トランシジョンレジスタをプログラムする (0または2)。 正トランシジョンレジスタを読み取る。 負トランシジョンレジスタをプログラムする (0または2)。 負トランシジョンレジスタを読み取る。 イネーブルレジスタをプログラムする (0または2)。 イネーブルレジスタを読み取る。 シーケンスイベントレジスタをコントロールするためのパス： ステータスレジスタを読み取る。 条件レジスタを読み取る。 正トランシジョンレジスタをプログラムする (0~6)。 正トランシジョンレジスタを読み取る。 負トランシジョンレジスタをプログラムする (0~6)。 負トランシジョンレジスタを読み取る。 イネーブルレジスタをプログラムする (0~6)。 イネーブルレジスタを読み取る。 トリガイベントレジスタをコントロールするためのパス： ステータスレジスタを読み取る。 条件レジスタを読み取る。 正トランシジョンレジスタをプログラムする (0または2)。 正トランシジョンレジスタを読み取る。 負トランシジョンレジスタをプログラムする (0または2)。 負トランシジョンレジスタを読み取る。 イネーブルレジスタをプログラムする (0または2)。 イネーブルレジスタを読み取る。 ステータスレジスタをデフォルト状態に戻す。 エラー待ち行列にアクセスするためのパス： 一番新しいエラーメッセージを要求する。 待ち行列に入るエラーおよびステータスメッセージを指定する。 </pre>

表B-6 (続き)
STATusコマンドサマリ

コマンド	説明
:QUESTIONable [:EVENT]? :CONDition? :PTRansition<n> :PTRansition? :NTRansition<n> :NTRansition? :ENABLE<n> :ENABLE?	被疑イベントレジスタをコントロールするためのパス： ステータスレジスタを読み取る。 条件レジスタを読み取る。 正トランジションレジスタをプログラムする。 正トランジションレジスタを読み取る。 負トランジションレジスタをプログラムする。 負トランジションレジスタを読み取る。 イネーブルレジスタをプログラムする。 イネーブルレジスタを読み取る。

表B-7
SYSTemコマンドサマリ

コマンド	説明
:SYSTem :ERRor? :POSetup RST PRESet SAV10 SAV1 SAV2 SAV3 SAV4 SAV5 SAV6 SAV7 SAV8 SAV9 :POSetup? :PRESet :VERSion?	エラー/ステータス待ち行列にメッセージを要求する。 電源投入時セットアップを選択する。 電源投入セットアップを要求する。 Model 7001をデフォルト状態に戻す。 SCPI規格の改訂水準を要求する。

表B-8

トリガコマンドサマリ

コマンド	説明
<pre>:INITiate [:IMMediate] :CONTinuous</pre>	<p>サブシステムコマンドパス： 1トリガ（スキャン）サイクルを開始させる。 トリガシステムの連続起動をイネーブル（1またはON）またはディスエーブル（0またはOFF）する。</p>
<pre>:ABORt</pre>	<p>トリガシステムをリセットし、アイドル状態に入る。</p>
<pre>:ARM[:SEQuence[1]] [:LAYer[1]] :IMMediate :COUNt<n> INFinite :COUNt? :SOURce HOLD IMMediate MANual BUS TLINk EXTernal :SOURce? :SIGNal :TCONfigure :DIRection SOURce ACCeptor :DIRection? :ASYNchronous :ILINe<n> :ILINe? :OLINe<n> :OLINe? :LAYer2 :IMMediate :COUNt<n> INFinite :COUNt? :DELay<num> :DELay? :SOURce HOLD IMMediate MANual BUS TLINk EXTernal TIMer :SOURce? :SIGNal :TIMer<num> :TIMer?</pre>	<p>スキャンを設定するためのサブシステムコマンドパス： スキャンのアームレイヤをプログラムするためのパス： コントロールソースをループアラウンドする。 アーム回数（1～9999またはINFinite）をプログラムする。 プログラムされたアームカウントを要求する。 Arm Layer 1コントロールソースを選択する。</p> <p>プログラムされたソース（イベント）を要求する。 コントロールソースをループアラウンドする。 トリガを設定するためのパス： バイパスをイネーブル（SOUR）またはディスエーブル（ACC）する。 プログラムされた方向を要求する。 非同期Trigger Linkモードを設定するためのパス： 入力ライン（1～6）を選択する。 プログラムされた入力ラインを要求する。 出力ライン（1～6）を選択する。 プログラムされた出力ラインを要求する。</p> <p>スキャンのスキャンレイヤをプログラムするためのパス： コントロールソースをループアラウンドする。 スキャン回数（1～9999またはINFinite）をプログラムする。 プログラムされたスキャンカウントを要求する。 ディレイ（0～99999.999秒）をプログラムする。 プログラムされたディレイを要求する。 Arm Layer 2コントロールソースを選択する。</p> <p>プログラムされたソース（イベント）を要求する。 スキャンコントロールソースを1回だけバイパスする。 タイマ時間間隔（0～99999.999秒）を設定する。 プログラムされたタイマ時間間隔を要求する。</p>

表B-8 (続き)
トリガコマンドサマリ

コマンド	説明
<pre> :TCONfigure :DIRection SOURce ACceptor :DIRection? :ASYNchronous :ILINe<n> :ILINe? :OLINe<n> :OLINe? </pre>	<p>Trigger Linkを設定するためのパス： バイパスをイネーブル (SOUR) またはディスエーブル (ACC) する。 プログラムされた方向を要求する。</p> <p>非同期モードを設定するためのパス： 入力ライン (1~6) を選択する。 プログラムされた入力ラインを要求する。 出力ライン (1~6) を選択する。 プログラムされた出力ラインを要求する。</p>
<pre> :TRIGger[:SEQuence[1]] :IMMEDIATE :COUNT<n> INFinite :AUTO ON/OFF :COUNT? :DELay<num> :DELay? :SOURce HOLD IMMEDIATE MANual BUS TLINK EXTernal TIMer :SOURce? :SIGNAL :TIMER<num> :TIMER? :TCONfigure :PROTOcol ASYNchronous SSYNchronous :PROTOcol? :DIRection SOURce ACceptor :DIRection? :ASYNchronous :ILINe<n> :ILINe? :OLINe<n> :OLINe? :SSYNchronous :LINe<n> :LINe? </pre>	<p>スキヤンのチャンネルレイヤをプログラムするためのパス： チャンネルを即時にスキヤンする。(コントロールソースをループアラウンド)。 チャンネル数 (1~9999またはINFinite) をプログラムする。 scan-list-lengthを使う。 プログラムされたチャンネルカウントを要求する。 ディレイ (0~99999.999秒) をプログラムする。 プログラムされたディレイを要求する。 Trigger Layerコントロールソースを選択する。</p> <p>プログラムされたソース (イベント) を要求する。 チャンネルコントロールソースを1回だけバイパスする。 タイマ時間間隔 (0~99999.999秒) を設定する。 プログラムされたタイマ時間間隔を要求する。</p>
<pre> :TCONfigure :PROTOcol ASYNchronous SSYNchronous :PROTOcol? :DIRection SOURce ACceptor :DIRection? :ASYNchronous :ILINe<n> :ILINe? :OLINe<n> :OLINe? :SSYNchronous :LINe<n> :LINe? </pre>	<p>トリガを設定するためのパス： プロトコル (非同期または半同期Trigger Link) を選択する。</p> <p>プログラムされたプロトコルを要求する。 バイパスをイネーブル (SOUR) またはディスエーブル (ACC) する。 プログラムされた方向を要求する。</p> <p>非同期Trigger Linkモードを設定するためのパス： 入力ライン (1~6) を選択する。 プログラムされた入力ラインを要求する。 出力ライン (1~6) を選択する。 プログラムされた出力ラインを要求する。</p> <p>半同期Trigger Linkモードを設定するためのパス： トリガライン (1~6) を選択する。 プログラムされたトリガラインを要求する。</p>

付録C

インタフェースファンクションコード

インタフェースファンクションコードは、IEEE-488規格の一部をなすもので、機器の、各種のインタフェースファンクションをサポートする機能を定義します。本書中のここ以外に記載されているプログラミングコマンドと混同しないでください。Model 7001のインタフェースファンクションコードを表C-1に示します。これらのコードで定義されるModel 7001の機能は以下の通りです。

SH (ソースハンドシェイクファンクション) — SH1は、Model 7001の、データバスを通じてのメッセージ/データの転送を開始する機能を定義します。

AH (アクセプタハンドシェイクファンクション) — AH1は、Model 7001の、データバスを通じて送信されたメッセージ/データを正しく受信する機能を定義します。

T (トークファンクション) — Model 7001の、バスを通じてデータを他の装置へ送信する機能は、Tファンクションによって与えられます。Model 7001のトーカー機能 (T5) は、Model 7001がトークにアドレッシングされた後にのみ有効になります。

L (リスナファンクション) — Model 7001の、バスを通じて他の装置から装置依存データを受信する機能は、Lファンクションによって与えられます。Model 7001のリスナ機能 (L4) は、Model 7001がリスンにアドレッシングされた後にのみ有効になります。

SR (サービス要求ファンクション) — SR1は、Model 7001の、コントローラにサービスを要求する機能を定義します。

RL (リモート-ローカルファンクション) — RL1は、Model 7001の、リモートモードまたはローカルモードに入る機能を定義します。

PP (パラレルポールファンクション) — Model 7001は、パラレルポール機能を備えていません (PP0)。

DC (装置クリアファンクション) — DC1は、Model 7001の、クリア (イニシャライズ) される機能を定義します。

DT (装置トリガファンクション) — DT1は、Model 182の、読取りをトリガさせる機能を定義します。

C (コントローラファンクション) — Model 7001は、コントローラ機能を備えていません (C0)。

TE (拡張トーカーファンクション) — Model 7001は、拡張トーカー機能を備えていません (TE0)。

LE (拡張リスナファンクション) — Model 7001は、拡張リスナ機能を備えていません (LE0)。

E (バスドライバタイプ) — Model 7001は、オープンコレクタ型のバスドライバを備えています (E1)。

表C-1

Model 7001インタフェースファンクションコード

コード	インタフェースファンクション
SH1	ソースハンドシェーク機能
AH1	アクセプタハンドシェーク機能
T5	トーカ (ベーシックトーカ、シリアルボール、LAGではトークにアドレッシングされない)
L4	リスナ (ベーシックリスナ、TAGではリスンにアドレッシングされない)
SR1	サービス要求機能
RL1	リモート/ローカル機能
PP0	パラレルボール機能なし
DC1	装置クリア機能
DT1	装置トリガ機能
C0	コントローラ機能なし
E1	オープンコレクタ型バスドライバ
TE0	拡張トーカ機能なし
LE0	拡張リスナ機能なし

付録D

ASCII文字およびIEEE-488マルチライン インタフェースコマンドメッセージ

10進数	16進数	ASCII	IEEE-488 メッセージ*
0	00	NUL	
1	01	SOH	GTL
2	02	STX	
3	03	ETX	
4	04	EOT	SDC
5	05	ENQ	PPC
6	06	ACK	
7	07	BEL	
8	08	BS	GET
9	09	HT	TCT
10	0A	LF	
11	0B	VT	
12	0C	FF	
13	0D	CR	
14	0E	SO	
15	0F	SI	
16	10	DLE	
17	11	DC1	LLO
18	12	DC2	
19	13	DC3	
20	14	DC4	DCL
21	15	NAK	PPU
22	16	SYN	
23	17	ETB	
24	18	CAN	SPE
25	19	EM	SPD
26	1A	SUB	
27	1B	ESC	
28	1C	FS	
29	1D	GS	
30	1E	RS	
31	1F	US	

* ATNが真の状態で送信または受信されるメッセージ

10進数	16進数	ASCII	IEEE-488 メッセージ*
32	20	SP	MLA0
33	21	!	MLA1
34	22	"	MLA2
35	23	#	MLA3
36	24	\$	MLA4
37	25	%	MLA5
38	26	&	MLA6
39	27	'	MLA7
40	28	(MLA8
41	29)	MLA9
42	2A	*	MLA10
43	2B	+	MLA11
44	2C	,	MLA12
45	2D	-	MLA13
46	2E	.	MLA14
47	2F	/	MLA15
48	30	0	MLA16
49	31	1	MLA17
50	32	2	MLA18
51	33	3	MLA19
52	34	4	MLA20
53	35	5	MLA21
54	36	6	MLA22
55	37	7	MLA23
56	38	8	MLA24
57	39	9	MLA25
58	3A	:	MLA26
59	3B	;	MLA27
60	3C	<	MLA28
61	3D	=	MLA29
62	3E	>	MLA30
63	3F	?	UNL

* ATNが真の状態で送信または受信されるメッセージ。数字は、
 MLA (My Listen Address) となる一次アドレスを表わす。

10進数	16進数	ASCII	IEEE-488 メッセージ*
64	40	@	MTA0
65	41	A	MTA1
66	42	B	MTA2
67	43	C	MTA3
68	44	D	MTA4
69	45	E	MTA5
70	46	F	MTA6
71	47	G	MTA7
72	48	H	MTA8
73	49	I	MTA9
74	4A	J	MTA10
75	4B	K	MTA11
76	4C	L	MTA12
77	4D	M	MTA13
78	4E	N	MTA14
79	4F	O	MTA15
80	50	P	MTA16
81	51	Q	MTA17
82	52	R	MTA18
83	53	S	MTA19
84	54	T	MTA20
85	55	U	MTA21
86	56	V	MTA22
87	57	W	MTA23
88	58	X	MTA24
89	59	Y	MTA25
90	5A	Z	MTA26
91	5B	[MTA27
92	5C	\	MTA28
93	5D]	MTA29
94	5E	^	MTA30
95	5F	_	UNL

* ATNが真の状態で送信または受信されるメッセージ。数字は、MTA (My Talk Address) となる一次アドレスを表わす。

10進数	16進数	ASCII	IEEE-488 メッセージ*
96	60	〇	MSA 0,PPE
97	61	a	MSA 1,PPE
98	62	b	MSA 2,PPE
99	63	c	MSA 3,PPE
100	64	d	MSA 4,PPE
101	65	e	MSA 5,PPE
102	66	f	MSA 6,PPE
103	67	g	MSA 7,PPE
104	68	h	MSA 8,PPE
105	69	i	MSA 9,PPE
106	6A	j	MSA 10,PPE
107	6B	k	MSA 11,PPE
108	6C	l	MSA 12,PPE
109	6D	m	MSA 13,PPE
110	6E	n	MSA 14,PPE
111	6F	o	MSA 15,PPE
112	70	p	MSA 16,PPD
113	71	q	MSA 17,PPD
114	72	r	MSA 18,PPD
115	73	s	MSA 19,PPD
116	74	t	MSA 20,PPD
117	75	u	MSA 21,PPD
118	76	v	MSA 22,PPD
119	77	w	MSA 23,PPD
120	78	x	MSA 24,PPD
121	79	y	MSA 25,PPD
122	7A	z	MSA 26,PPD
123	7B	{	MSA 27,PPD
124	7C		MSA 28,PPD
125	7D	}	MSA 29,PPD
126	7E	~	MSA 30,PPD
127	7F	DEL	

* ATNが真の状態で送信または受信されるメッセージ。数字は、MSA (My Secondary Address) となる二次アドレスを表わす。

付録E

コントローラプログラム

以下のプログラムは、ユーザのための簡単な参考としてここに示すもので、特定の要求を満たすものではありません。それぞれのプログラムでは、Model 7001へコマンドメッセージを送信し、問合せコマンドに対するModel 7001の応答を取得、表示させることができます。

ここに含まれるのは以下のコントローラのためのプログラムです。

- Capital Equipment Corporation PC<>488インタフェース (Keithley Model PC-488-CEC IEEE-488インタフェースカード) を装備したIBM PC/XT/ATのためのBASICプログラム。
- IOtech Driver488ソフトウェアを装備したIBM PC/XT/ATのためのBASICプログラム。
- Hewlett-Packard Model 9000 Series 200/300コンピュータのためのBASICプログラム。
- ASYSTソフトウェアパッケージ (Keithley社で販売しています) で書かれた、Capital Equipment Corporation PC<>488インタフェースを装備したIBM PC/XT/ATのためのBASICプログラム。

Capital Equipment Corporation PC<>488インタフェース (Keithley PC-488-CEC)を装備した IBM PC/XT/AT

概要

ここでは、IBM PC/XT/ATコンピュータおよびCapital Equipment Corporation PC<>488インタフェースを有する Model 7001のコントロールについて一般事項を記載します。詳細は、「CEC PC<>488 Programing and Reference Manual」を参照してください。

システムのイニシャライズ

どのプログラムでも、最初のステップは、INITIALIZEルーチンを使ってシステムをイニシャライズすることです。このルーチンは、PC<>488の一次IEEE-488バスアドレスを設定し、使用するコントローラのタイプを指定し、バス上の他の装置へインタフェースクリア (IFC) を送信します。DOSを使ってIBMコンピュータをブートアップし、BASICを入力したら、どのプログラムの始めにもプログラム1のステートメントを使ってシステムをイニシャライズすることができます。

行30はPC<>488のアドレスを21に設定しますが、バス上の他の装置のアドレスと矛盾しない限り、21以外の有効アドレスを使うことができます。

インタフェースの基本プログラミング ステートメント

本項では、Model 7001のオペレーションに不可欠のPC<>488ステートメントについて説明します。プログラミングステートメントのうち一部を表E-1に示します。目的のアクションを実行するには、IBM BASIC CALLステートメントを使います。CALLルーチンを使うには、その前に、BASICがそのルーチンのオフセットアドレスを知っていなければなりません。表中、各プログラミングシーケンスの最初のステートメントは、CALLルーチンのオフセットアドレスを定義するものです。

プログラム例

以下のプログラムは、IBM PC/XT/ATコンピュータから Model 7001へコマンドメッセージを送信し、問合せに対する応答をCRT上に表示させるものです。コンピュータは、CECインタフェースカードおよびDOS 2.0オペレーティングシステム（または後続バージョン）を搭載していなければなりません。

手順

1. フロントパネル上のMENUキーを使って、Model 7001の一次IEEE-488アドレス (GPIO選択) を7に設定します。
2. 電源を切って、IBMコンピュータにインストールされているIEEE-488インタフェースカードにModel 7001を接続します。
3. コンピュータキーボードでBASICをタイプインし、インタプリティブASIC言語に入ります。
4. プログラム2の各行を入力します。このとき、各行の入力の都度、RETURNキーを押します。
5. プログラムを実行し、ENTER COMMAND MESSAGEプロンプトが表示されたら目的のコマンドストリングをタイプインします。たとえば、現在定義されているスキャンリストを要求するには、:SCAN?をタイプインし、RETURNキーを押します。コンピュータCRT上にスキャンリストが表示されます。

表E-1

バスコマンドの送信に必要なBASICステートメント

アクション	PC<>488ステートメント
装置7へコマンドストリング (CMD\$) を送信する。	SEND=9 ADDRESS%=7 CALL SEND (ADDRESS%, CMD\$, STATUS%)
装置7からストリングを取得する。	ENTER=21 ADDRESS%=7 RESPONSE\$=SPACES\$ (50) CALL ENTER (RESPONSE\$, LENGTH%, ADDRESS%, STATUS%)
装置7へGTLを送信する。	TRANSMIT=3 CMD\$="LISTEN7 GTL" CALL TRANSMIT (CMD\$, STATUS%)
装置7へSDCを送信する。	TRANSMIT=3 CMD\$="LISTEN 7 SDC" CALL TRANSMIT (CMD\$, STATUS%)
全装置へDCLを送信する。	TRANSMIT=3 CMD\$="LISTEN 7 DCL" CALL TRANSMIT (CMD\$, STATU%)
リモートイネーブルを送信する。	TRANSMIT=3 CMD\$="REN LISTEN 7" CALL TRANSMIT (CMD\$, STATUS%)
装置7をシリアルポールする。	SPOLL=12 ADDRESS%=7 CALL SPOLL% (ADDRESS%, POLL%, STATUS%)

```

10 DEF SEG=&HC400
20 INITIALIZE=0
30 MY.ADDRESS%=21
40 CONTROLLER%=0
50 CALL INITIALIZE (MY.ADDRESS%,
    CONTROLLER%)

```

```

! Memory address.
! Offset address of routine.
! Set PC<>488 address.
! Set for system control.
! Initialize system.

```

プログラム1

```

10 DEF SEG=#HC400           ' Memory address.
20 INITIALIZE=0             ' Define call offsets.
30 SEND=9:ENTER=21
40 MY.ADDRESS#=21          ' Set PC<>488 address.
50 INST.ADDRESS#=7        ' SET 7001 address.
60 CONTROLLER#=0          ' Set for system control.
70 CALL INITIALIZE (MY.ADDRESS%, ' Initialize system.
    CONTROLLER%)
80 LINE INPUT "ENTER COMMAND ' Prompt for command message.
    MESSAGE ";CMD$
90 CALL SEND (INST.ADDRESS%, ' Send command message.
    CMD$, STATUS%)
100 IF RIGHT$(CMD$, 1) <> "?" ' Test for query. If command is a
    THEN GO TO 80             query, goto 80.
110 RESPONSE$=SPACE$(80)   ' Make room for data.
120 CALL ENTER (RESPONSE$,LENGTH%, ' Get response to query from 7001.
    INST.ADDRESS%, STATUS%)
130 PRINT RESPONSE$       ' Display response to query.
140 GOTO 80                ' Repeat.
150 END

```

プログラム2

IOtech Driver488ソフトウェアを装備したIBM PC/XT/ATまたはPS/2

概要

Driver488ソフトウェアは、IBM PCおよびコンパチブルコンピュータのためのIEEE-488ドライバです。これをIEEE-488インタフェースボードと組み合わせると、このパッケージは、BASIC、Pascal、Fortran、Cなどの言語から装置をコントロールすることができます。Driver488ソフトウェアは下記のものと同コンパチブルです。

- IOtech GP488BまたはGP488/2ボード
- National Instruments GPIB-PC、PCII、PCIIAまたはPCIIIボード
- Capital Equipment #01000-00300ボード
- IBM DPIBアダプタ

以下のGW-BASICプログラムは、Model 7001へコマンドメッセージを送信し、問合せに対するModel 7001の応答をCRT上に表示させるものです。コンピュータは、IEEE-488インタフェースボード、DOS 3.0オペレーティングシステム（または後続バージョン）、およびDOS I/O機能付き言語を搭載していなければなりません。Driver488コントローラは、取扱説明書に従ってインストールし、設定しなければなりません。

手順

1. フロントパネル上のMENUキーとGPIBメニュー項目を使って、Model 7001の一次アドレスを7に設定します。
2. 電源を切って、IBMコンピュータにインストールされているIEEE-488インタフェースにModel 7001を接続します。
3. コンピュータキーボードでGWBASICをタイプインし、MicrosoftインタプリティブBASIC言語に入ります。
4. 以下の各行を入力します。このとき、各行の入力の都度、RETURNキーを押します。
5. プログラムを実行し、ENTER COMMAND MESSAGEプロンプトが表示されたら目的のコマンドストリングをタイプインします。たとえば、現在定義されているスキャンリストを要求するには、:SCAN?をタイプインし、RETURNキーを押します。コンピュータCRT上にスキャンリストが表示されます。

```

10 OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1      ' Open file for output.
20 IOCTL#1,"BREAK"                            ' Reset.
30 PRINT#1,"RESET"
40 OPEN "\DEV\IEEEIN" FOR INPUT AS #2        ' Open file for inupt.
50 PRINT#1,"FILL ERROR"                      ' Enable SEQUENCE error detect.
60 LINE INPUT "ENTER COMMAND MESSAGE ";A$    ' Prompt for and get commands.
70 IF LEN(A$)=0 THEN 140                     ' Check for null.
80 PRINT #1,"OUTPUT 07:"+A$                 ' Send commands.
90 IF RIGHT$(A$, 1) <> "?" THEN GOTO 60      ' Check for query.
100 PRINT #1,"ENTER 07"                      ' Address 7001 to talk.
110 LINE INPUT#2,B$                          ' Get response.
120 PRINT B$                                 ' Display response message.
130 GOTO 60                                  ' Repeat.
140 END

```


Hewlett-Packard Model 9000 Series 200/300

概要

以下のプログラムは、Hewlett-Packard Model 9000 Series 200/300コンピュータからModel 7001へコマンドメッセージを送信し、問合せに対する応答をコンピュータCRT上に表示させるものです。コンピュータは、HP BASIC 4.0を搭載していなければなりません。

```

10 DIM A$(50), B$(50)
20 REMOTE 707
30 LINPUT "ENTER COMMAND MESSAGE", A$
40 OUTPUT 707; A$
50 L=LEN(A$)
60 IF A$(L,L) <> "?" THEN GOTO 30
70 ENTER 707; B$
80 PRINT B$
90 GOTO 30
100 END

```

手順

1. フロントパネル上のMENUキーとGPIBメニュー項目を使って、Model 7001の一次IEEE-488アドレスを7に設定します。
2. 電源を切って、HPコンピュータにインストールされているIEEE-488インタフェースカードにModel 7001を接続します。
3. 以下のプログラム各行を入力します。このとき、各行の入力の都度、ENTER/RETURNキーを押します。
4. RUNキーを押し、ENTER COMMAND MESSAGEプロンプトが表示されたら目的のコマンドストリングをタイプインします。たとえば、現在定義されているスキャンリストを要求するには、:SCAN?をタイプインし、RETURNキーを押します。コンピュータCRT上にスキャンリストが表示されます。

```

! Place 7001 in remote.
! Prompt for command.
! Address 7001 to listen; send command.
! If command is not a query, goto 30.
! If query, address 7001 to talk.
! Display response to query command.
! Repeat

```

CEC PC<>488インタフェースおよび ASYSTソフトウェアを装備した IBM PC/XT/ATまたは コンパチブルコンピュータ

概要

ASYSTは、3つの統合モジュールの中に、解析ソフト、グラフ作成ソフトおよびデータ収集ソフトが収められています。ASYSTに最低限必要なものは次の通りです。

- IBM PC/XT/ATまたは100%コンパチブルコンピュータ (実モードのPS/2、386または486系コンピュータを含む)
- DOS 2.0 (できれば、DOS3.0または後続バージョン)
- Intel 8087,80287,または80387数値演算プロセッサ
- 640K RAM
- 25ピン・パラレルポート
- ハードディスクおよびフロッピー1枚
- IBM CGA、EGA、VGAまたは100%コンパチブルコンピュータ;Herculesモノクロームグラフィックス
- できれば、LIM (Lotus/Intel/Microsoft) 拡張メモリ

この例では、IEEE-488インタフェース用としてオプションのソフトウェアモジュールが必要です。また、コンピュータは、CECインタフェースカード (PC<>488 Model 300または310、または4x488) を搭載していなければなりません。

手順

1. フロントパネルメニューを使って、Model 7001の一次アドレスを07に設定します。
2. 電源を切って、IBMコンピュータにインストールされているIEEE-488インタフェースカードにModel 7001を接続します。
3. ASYSTソフトウェアをブートします。Main Configuration Menuに入ります。OKプロンプトから<F2>キーを押すか、またはメニューバーからCONFIGを選択します。
4. ASYSTを通常通り設定する必要があることに加え、GPIBを設定しなければなりません。Overlay Configurationメニューで、Available OverlayリストからGPIBを選択します。さらに、ポップアップメニューからGPIB MasterおよびType 1 NEC GPIBドライバを選択します。
5. GPIB Configurationメニューから、バス番号、Capital Equipment PC<>488 (NEC uPD7210型) のボード番号、一次アドレス (0)、16進/Oアドレス (2B8、メモリアドレスとも言う)、および使用されていない割込みラインを選択します。
6. 変更をセーブし、OKプロンプトまたはメニューバーに戻ります。
7. コマンドラインエディタを使ってプログラム例を入力します。OKプロンプトからEDIT 7001.DMOをタイプするか、またはメニューバーからEDITを選択し、7001.DMOをタイプします。プログラムをセーブし、エディタから抜け出します。
8. 指示されたら、Lキーを使ってプログラムをロードするか、またはOKプロンプトからLOAD 7001.DMOをタイプします。
9. プログラムが正しくロードされたら、OKプロンプトからMAINをタイプし、実行を開始させます。
10. "Enter command string." プロンプトが表示されたら目的のコマンドストリングをタイプインします。たとえば、マルチプレクサチャンネルを閉じるには、:CLOSE (@1!:1:1!5) をタイプインし、RETURNキーを押します。Model 7001がード1上の閉チャンネルを表示します。閉チャンネルを調べるには、:CLOSE:STAT?をタイプインし、RETURNキーを押します。コンピュータディスプレイ上に閉チャンネルリストが表示されます。
11. <Control-Break>をタイプし、7001.DMOのプログラムループから抜け出し、OKプロンプトに戻ります。プログラム定義語は、FORGET 7001をタイプすることによってASYSTデイクシヨナリから削除することができます。

```

BUS.INIT                \ Initialize bus
SEND.INTERFACE.CLEAR   \ Take control of bus
REMOTE.ENABLE.ON       \ Allow remote operation
SYNCHRONOUS.GPIB       \ Make sure ASYST is in sync. mode
?GPIB.DEVICES          \ List controller parameters
07 GPIB.DEVICE 7001     \ Set 7001 address and name
    EOI.ON              \ Enable end or identify
    EOS.ON              \ Enable end of string terminator
    10 EOS.CHARACTER   \ Use LF terminator
    1000 TIMEOUT       \ Set timeout for 1sec
?GPIB.DEVICE           \ List 7001 parameters
100 STRING COMMAND     \ Allocate 100 bytes for user input
100 STRING RESPONSE    \ Allocate 100 bytes for 7001 response

: GET.OUTPUT           \ Definition for 7001 output
  RESPONSE GPIB.READ   \ Get response
  CR RESPONSE "TYPE    \ Display response
;

: GET.INPUT            \ Definition for user input
  CR ." Enter command string: " \ Prompt user
  "INPUT COMMAND " :=    \ Get commands
  " ?" COMMAND "WITHIN  \ Test for query
  IF 7001               \ If query, make 7001 current device
    COMMAND GPIB.WRITE  \ Send commands to 7001
    GET.OUTPUT          \ Get response from 7001
  ELSE 7001             \ If not query, make 7001 current device
    COMMAND GPIB.WRITE  \ Send commands to 7001
  THEN
;

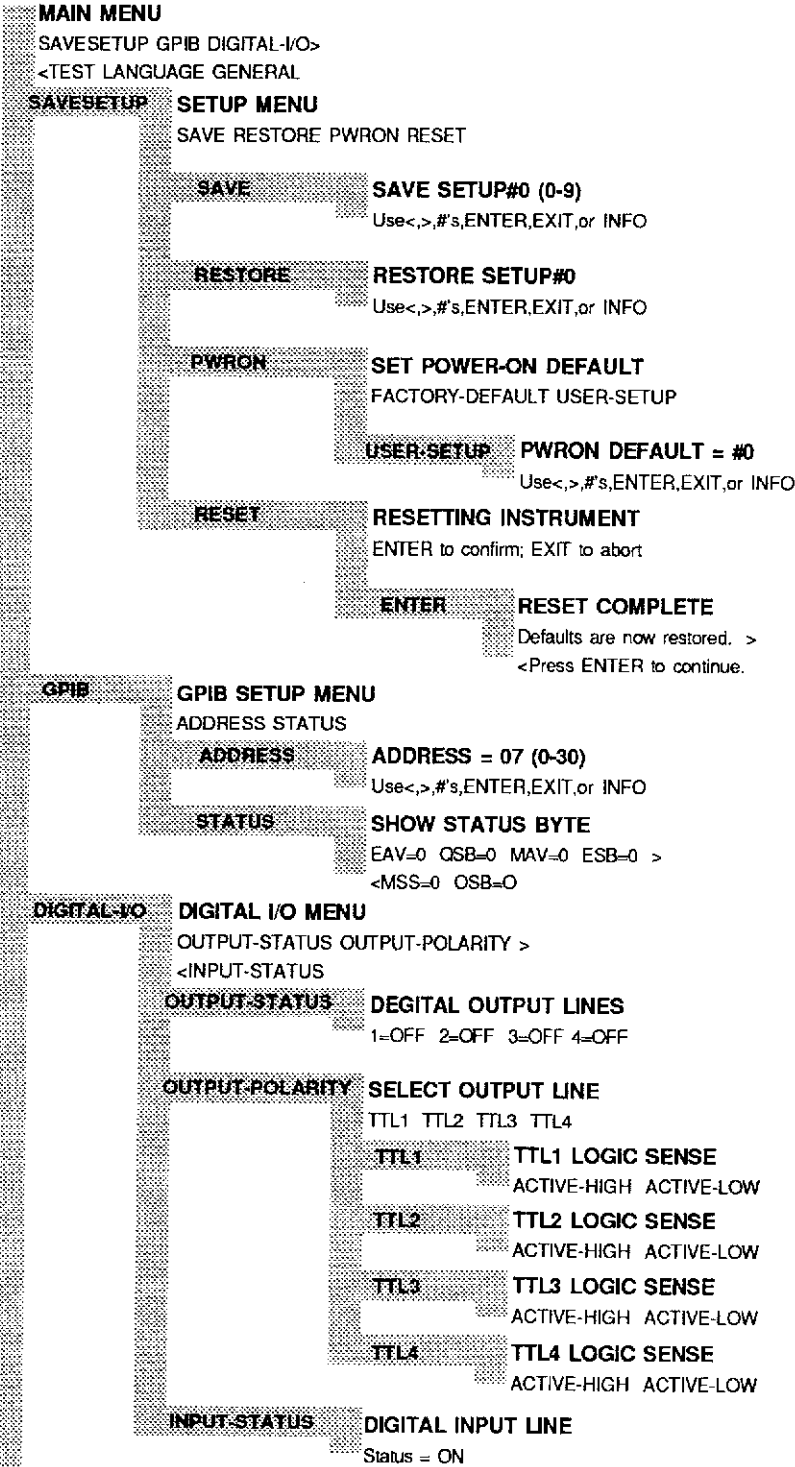
: MAIN                 \ Definition for main program
  BEGIN                \ Start loop
  ME                    \ Make controller current device
  GET.INPUT            \ Get commands from user
  AGAIN                \ Return for more commands
;                       \ <Control-Break> to exit

```

付録F

メニュー構造

MENU



INFO: MAIN MENU

ここから一般動作条件を選択する。

INFO: SETUP MENU

ここでメインフレームパラメータが設定、セーブ、復元される。

INFO: SAVE SETUP

ここでメインフレームパラメータがストアされる。セットアップ番号を入力する。

INFO: RECALL SETUP

ここでメインフレームパラメータが復元される。セットアップ番号を入力する。

INFO: PWRON DEFAULT

出荷時デフォルトを選択するか、またはユーザセットアップ番号を入力する。

INFO: PWRON DEFAULT

ストアされているセットアップを選択する。

INFO: FACTORY RESET

メインフレームパラメータを元の出荷時デフォルトにリセットする。

INFO: GPIB ADDRESS

目的のGPIBアドレスを入力する。

INFO: STATUS BYTE

ステータスがビット順 (0-7) に表示される。ENTERを押して操作継続。

INFO: DIG-I/O MENU

ここでメインフレームデジタルパラメータを設定する。

INFO: DIGITAL OUTPUTS

任意の数字キーを押してラインのON/OFFを切り替える。

INFO: POLARITY

アクティブハイまたはアクティブローを選択する。

INFO: POLARITY

アクティブハイまたはアクティブローを選択する。

INFO: POLARITY

アクティブハイまたはアクティブローを選択する。

INFO: POLARITY

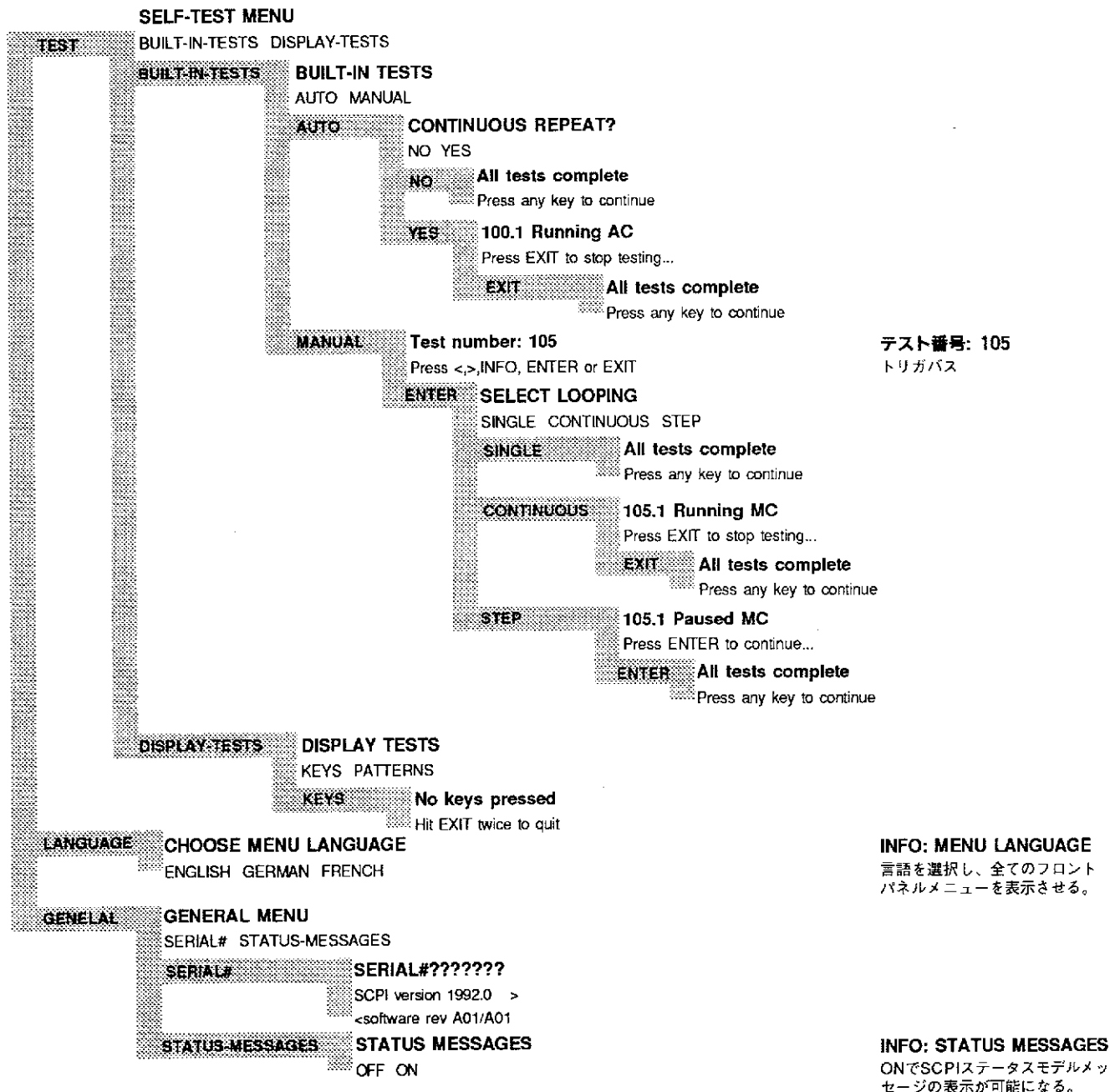
アクティブハイまたはアクティブローを選択する。

INFO: POLARITY

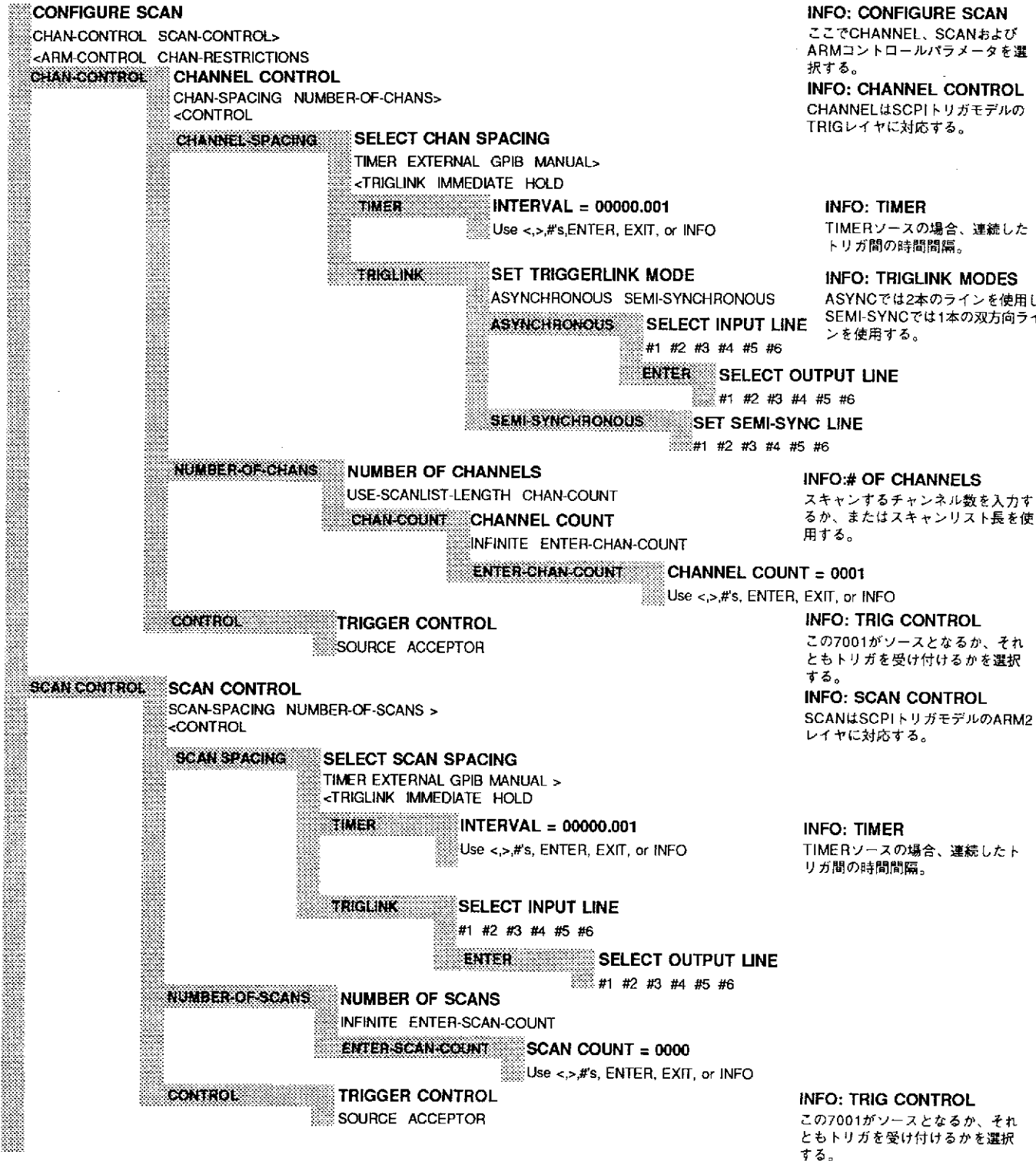
アクティブハイまたはアクティブローを選択する。

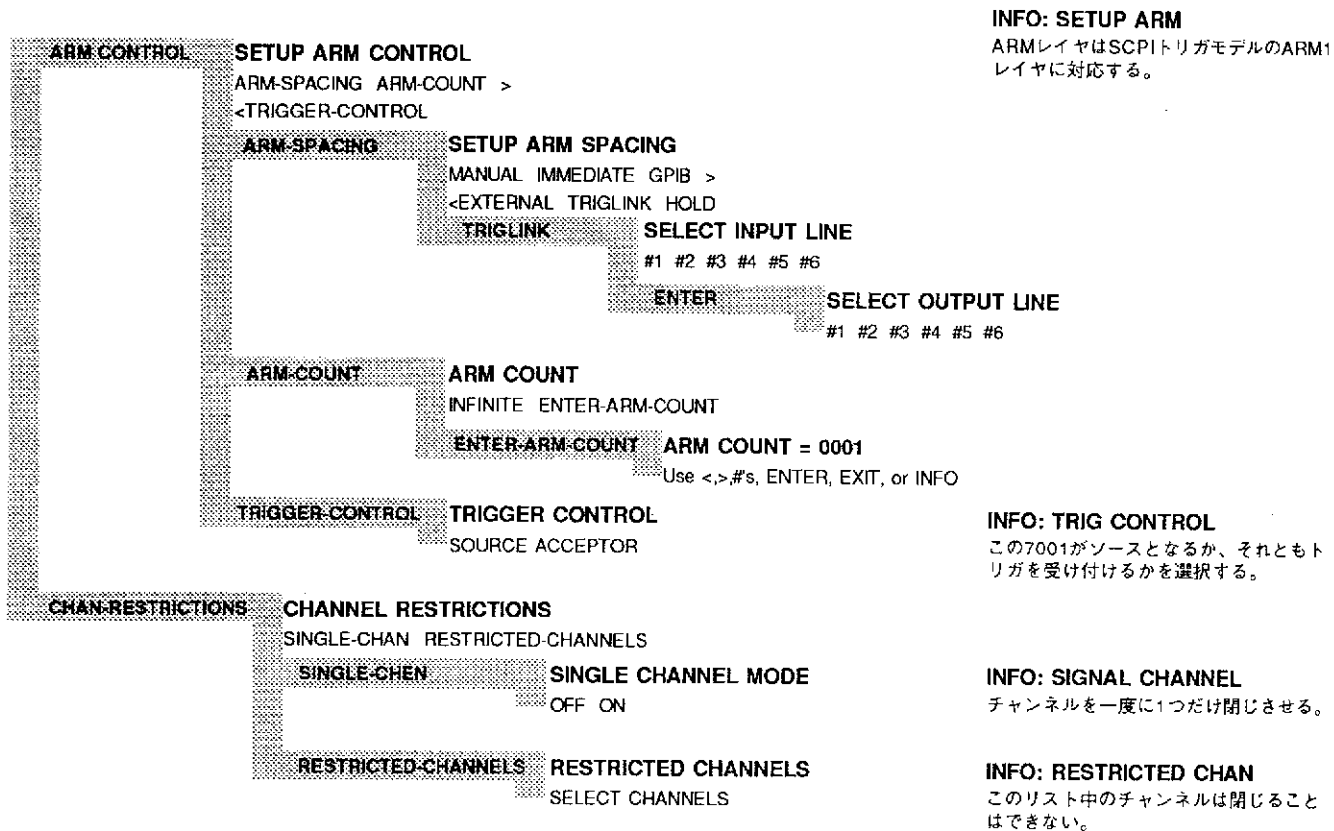
INFO: DIGITAL INPUT

デジタル入力の状態。



SCAN CONFIGURATION





CARD CONFIGURATION

```

CARD CONFIG MENU
TYPE #-OF-POLES CARD-PAIR >
< DELAY READ-I/O-CARD

TYPE          SET CARD TYPE
              SLOT-1 SLOT-2

              SLOT-1  SLOT-1 CARD: 9990
              Use <,>, ENTER, EXIT, or INFO

              SLOT-2  SLOT-2 CARD: 9991
              Use <,>, ENTER, EXIT, or INFO

#-OF-POLES    SELECT #OF POLES
              SLOT-1  SLOT-2

              SLOT-1  # OF POLES SLOT-1
              1-POLE 2-POLE 4-POLE

              SLOT-2  # OF POLES SLOT-2
              1-POLE 2-POLE 4-POLE

CARD-PAIR     SET CARD PAIR
              OFF ON

DELAY         SET DELAY FOR:
              SLOT-1 SLOT-2

              SLOT-1  (1)DELAY = 00000.000
              Use <,>.#s, ENTER, EXIT, or INFO

              SLOT-2  (2)DELAY = 00000.000
              Use <,>.#s, ENTER, EXIT, or INFO

READ-I/O-CARD No I/O cards
              I/O CARD STATUS:Use EXIT to quit
    
```

INFO: CARD CONFIGURE

このメニューからカードパラメータを選択する。

INFO: SET CARD TYPE

カード型番を入力する。

INFO: SET CARD TYPE

カード型番を入力する。

INFO: SET CARD TYPE

カード型番を入力する。

INFO: NUMBER OF POLES

ポールは接点の数。

INFO: NUMBER OF POLES

ポールは接点の数。

INFO: NUMBER OF POLES

ポールは接点の数。

INFO: CARD PAIRING

両カードとも同じリレーを閉じさせる。

INFO:SET DELAY

チャンネルが閉じてからチャンネルレディパルス発生までのユーザ指定ディレイを入力する。

INFO:SET DELAY

チャンネルが閉じてからチャンネルレディパルス発生までのユーザ指定ディレイを入力する。

INFO:SET DELAY

チャンネルが閉じてからチャンネルレディパルス発生までのユーザ指定ディレイを入力する。

INFO: I/O STATUS

I/Oカードのステータスが表示される。

付録G

IEEE-488バスの概要

はじめに

IEEE-488バスは、基本的には、2つまたはそれ以上の電子装置を結ぶ通信システムにすぎません。装置は計器の場合もコンピュータの場合もあります。コンピュータは、バス上で使用される場合、全装置間の通信交換を管理する役割を果たし、コントローラと言われます。コントローラによる管理の内容は、どの装置がトークし、どの装置がリスンするかを決めることです。トークとして装置は情報を出力し、リスンとして装置は情報を受信します。装置の管理を容易にするために、各装置には固有のアドレス番号が付されます。

バス上では、一度に1台の装置しかトークできず、これはコントローラによってトークにアドレッシングされます。トークしている装置はアクティブトークと言われます。トークに対してリスンする必要のある装置は、コントローラによってリスンにアドレッシングされます。各リスナはアクティブリスナと言われます。リスンする必要のない装置はアンリスンを命じられます。アンリスン命令を出す理由は、バス時間がリスンに取られることを考慮してバスの情報転送速度を最適化するためです。

コントロールラインの使用により、トークからリスナへの情報転送プロセスにハンドシェイクシーケンスが生じます。このハンドシェイクシーケンスのおかげで情報転送の信頼性が保証されます。アクティブコントローラ（トーク）とリスナの間の基本的なハンドシェイクシーケンスは次の通りです。

1. リスナがリスン可能な状態にあることを知らせる。

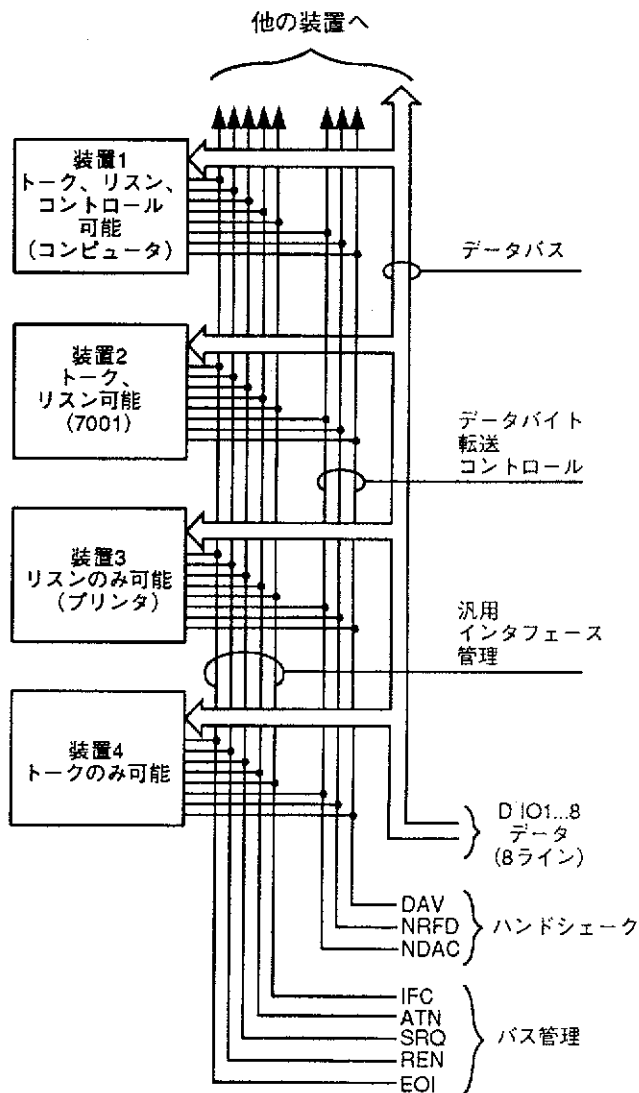
2. トークがデータのバイトをバス上に載せ、リスナにとってデータが利用可能であることを知らせる。
3. リスナは、データが利用可能であることを認知し、データを受信し、データを受信されたことを知らせる。
4. トークは、データを受信されたことを認知し、データの送信を停止し、データが送信されていないことを知らせる。
5. リスナは、バス上にデータがないことを認知し、データの次のバイトを受け取ることが可能な状態にあることを知らせる。

バスについて

IEEE-488バスは、しばしば GPIB（汎用インタフェースバス）とも言われ、過剰のバスラインを使わずにデータ転送を最適化するための並列転送媒体として開発されたものです。この目的に沿って、バスには8本のデータラインしかなく、これらをデータにも大多数のコマンドにも使います。5本のバス管理ラインと3本のハンドシェイクラインがバスシグナルラインを補完します。

オペレーションをコントロールするための標準的なセットアップを図G-1に示します。一般に、システムは、1台のコントローラと、コマンドが送られる他の多数の装置を含んでいます。装置オペレーションは、コントローラ、トーク、リスナの3形態に分類されます。コントローラは、その名の通り、バス上の機器をコントロールします。トークはデータを送信し、リスナはデータを受信します。装置は、タイプに応じて、トークにしかかなれないもの、リスナにしかかなれないもの、トークにもリスナにもなれるものがあります。

コントローラは、システムコントローラと基本コントローラの2種類があります。どちらも他の機器をコントロールできますが、システムコントローラはシステムに関して絶対的な権限をもっています。複数のコントローラを有するシステムでは、所定の時点で1台のコントローラしかアクティブになれません。コントロールをあるコントローラから別のコントローラへ渡すために、ある種のプロトコルを使います。



図G-1 IEEE-488バスコンフィギュレーション

IEEE-488バスはコントローラを含めて15装置に制限されず。したがって、この限度内の任意の数のトーカーとリスナがバス上に同時に存在することができます。複数の装置がリスンするように命じられることはありますが、バス上には1台のアクティブトーカーしか存在できません。そうでないと、通信がスクランブルします。

装置をトークまたはリスン状態にするには、適切なトークまたはリスンコマンドを送信します。これらのトークおよびリスンコマンドは、機器の一次アドレスから導出されます。一次アドレスは0から31までの任意の値をとることができます。原則として、リヤパネル上のDIPスイッチによって設定するか、またはModel 7001のフロントパネルからプログラムします。バスを通じて送り出される実際のリスンアドレス値は、一次アドレスと\$20のORをとることによって得られます。たとえば、一次アドレスが10進数7 (\$07)であれば、実際のリスンアドレスは\$27 (\$27 = \$07 + \$20)です。同様に、トークアドレスは、一次アドレスと\$40のORをとることによって得られます。上の例を引けば、10進数7の一次アドレスから導出されるトークアドレスは\$47 (\$47 = \$07 + \$40)ということになります。

IEEE-488規格には、二次アドレッシングと呼ばれる別のアドレッシングモードも定められています。二次アドレッシングの範囲は\$60~\$7Fです。ただし、Model 7001を含めて多くの装置は二次アドレッシングをしません。

装置がトークまたはリスンにアドレッシングされると、それに応じたバストランザクションが行なわれます。たとえば、Model 7001をトークにアドレッシングすると、Model 7001はそのデータストリングを一度に1バイトずつバス上に載せます。コントローラがその情報を読み取ります。そして、該当するソフトウェアにより情報を目的のロケーションへ送ることができます。

バスライン

IEEE-488バスのシグナルラインは、データライン、管理ライン、ハンドシェイクラインの3種類に分けられます。データラインはバスデータとコマンドを扱い、管理ラインおよびハンドシェイクラインはデータ転送とオペレーションが正しく行なわれるようにします。各バスラインはアクティブローで、ほぼ0ボルトが論理値1 (真)を表わします。以下に、これらのラインの動作について説明します。

データライン

IEEE-488バスでは、データを一度に1バイトずつ転送する8本のデータラインを使います。これら8本のデータラインはDIO1（データ入出力）～DIO8（データ入出力）で、データとマルチラインコマンドの両方を送信するのに使われ、双方向です。これらのデータラインはLowを真論理として動作します。

バス管理ライン

5本のバス管理ラインは、インタフェースを正しく制御、管理する役割をします。これらのラインは、ユニラインコマンドを送信するのに使います。

ATN (Attention) — ATNラインは、管理ラインの中でもより重要なラインの一つです。つまり、このラインの状態によって、データバス上の情報がどう解釈されるかが決まるからです。

IFC (Interface Clear) — IFCは、その名の通り、バスからの機器のクリアをコントロールします。

REN (Remote Enable) — RENラインは、バス上の機器をリモートモードにするのに使います。

EOI (End or Identify) — EOIは、通常、マルチバイトデータ転送シーケンスの終了を示すのに使います。

SRQ (Service Request) — このラインは、装置がコントローラにサービスを要求するときに使います。

ハンドシェークライン

バスハンドシェークラインは、インタロックされたシーケンスで動作します。この方式では、転送速度に関係なく信頼性の高いデータ伝送が可能です。一般には、データ転送の速度は、バス上の最も遅いアクティブ装置によって決まります。

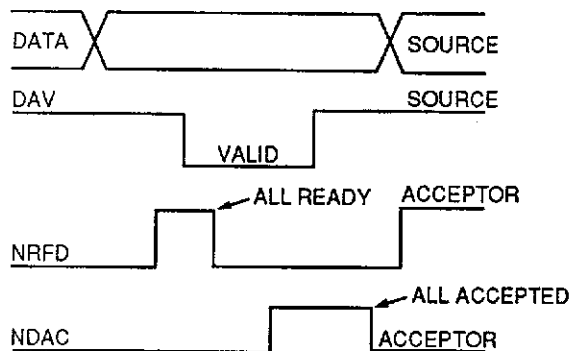
3本のハンドシェークラインのうち1本はソース（情報を送信するトーカー）によってコントロールされ、残りの2本は受信側装置（情報を受信するリスナ（1台または複数台））によってコントロールされます。3本のハンドシェークラインは次の通りです。

DAV (DATA VALID) — DAVラインの状態はソースによってコントロールされます。DAVは、受信側装置に対して、データバス情報が有効かどうかを知らせます。

NRFD (Not Ready For Data) — NRFDラインの状態はアクセプタ〔受信側装置〕によってコントロールされます。NRFDは、送信側装置に対して、受信側装置がレディ状態になるまでバイト転送シーケンスを保留するよう指示します。

NDAC (Not Data Accepted) — NDACラインも受信側装置によってコントロールされます。NDACは、ソースに対して、受信側装置がデータバイトを受信したかどうかを知らせます。

1データバイトを転送するための完全なハンドシェークシーケンスを図G-2に示します。データがデータライン上に載せられると、ソースは、NRFDがHighになっていること、すなわち、全てのアクティブ装置がレディ状態にあることを確かめます。このとき、NDACは、前のデータ転送の結果としてLowになっていなければなりません。これらの条件が満たされない場合、ソースは、NDACとNRFDが適正な状態になるまで待たなければなりません。ソースがコントローラの場合、NRFDとNDACは、ATNが真に設定されてから少なくとも10nsの間、安定していなければなりません。バスがハングアップする可能性を考慮して、多くのコントローラは、何らかの理由で転送シーケンスが停止した場合にメッセージを表示させるタイムアウトルーチンを備えています。



図G-2
IEEE-488ハンドシェークシーケンス

全てのNDACとNRFDが正しく設定されると、ソースはDAVをLowに設定し、これで、データライン上のバイトが有効になったことを受信側装置に知らせます。そして、全装置がデータを受信すると、NRFDがLowになり、NDACがHighになります。各装置はそれぞれ固有のレートでNDACを解放しますが、NDACが解放されてHighになるのは全装置がデータバイトを受信してからです。

上記のシーケンスは、データ、トークおよびリスンアドレス、マルチラインコマンドを転送するためのものです。以下に説明するように、データバスがデータ、アドレスまたはコマンドを含むかどうかは、ATNラインの状態によって決まります。

バスコマンド

Model 7001には、IEEE-488インタフェースを通じて多数の特殊バスコマンドを送ることができます。ここでは、以下の3種類に分けられるバスコマンドの目的について簡単に説明します。

1. ユニラインコマンド — 対応するバスラインを真に設定することによって送信される。たとえば、REN (Remote Enable) をアサートするのであれば、RENラインをLow (真) に設定する。
2. マルチラインコマンド — ATNラインが真 (Low) の状態でデータラインを通じて送信される汎用バスコマンド。
3. コモンコマンド — 全装置に共通のコマンド。ATNがHigh (偽) の状態で送信される。
4. SCPIコマンド — バス上の各装置に固有のコマンド。ATNがHigh (偽) の状態で送信される。

これらのバスコマンドとそれぞれの目的を表G-1に示します。

ユニラインコマンド

ATN、IFCおよびRENはコントローラによってのみアサートされます。SQRは外部装置によってアサートされます。EOIは、データ転送の方向に応じて、コントローラによってアサートされることも他の装置によってアサートされることもあります。以下に各コマンドについて説明します。各コマンドは、対応するバスラインを真に設定することによって送信されます。

REN (Remote Enable) — RENは、バス上の機器をリモートオペレーションにセットアップするために送信されます。RENが真になると、装置はローカルモードから解除されます。装置のコンフィギュレーションにもよりますが、LOCALボタン以外の全てのフロントパネルコントロール類 (装置がこのような装備になっている場合) は、RENが真になるとロックアウトされることがあります。原則として、RENは、バスを通じて機器をプログラムしようとする前に送信する必要があります。

EOI (End or Identify) — EOIは、マルチバイト転送シーケンスの最後のバイトを明確に識別するのに使います。これにより、いろいろな長さのデータワードを容易に伝送することができます。

IFC (Interface Clear) — IFCは、インタフェースをクリアし、全装置をトークおよびリスナアイドル状態に戻すのに使います。

ATN (Attention) — コントローラは、アドレスおよびマルチラインコマンドを送信すると同時にATNを送信します。

SRQ (Service Request) — SQRは、装置がコントローラにサービスを要求するとき、その装置によってアサートされます。

表G-1

IEEE-488バスコマンドサマリ

コマンドタイプ	コマンド	ATN ラインの 状態	コメント
ユニラインコマンド	REN (Remote Enable)	X	装置をリモートオペレーションにセットアップする。 伝送の終了を示す。 インタフェースをクリアする。 データバスの内容を定義する。 外部装置によってコントロールされる。
	EOI	X	
	IFC (Interface Clear)	X	
	ATN (Attention)	Low	
	SRQ	X	
ユニバーサルマルチ ラインコマンド	LLO (Local Lockout)	Low	ローカルオペレーションをロックアウトする。 装置をデフォルト状態に戻す。 シリアルポーリングをイネーブルする。 シリアルポーリングをディスエーブルする。
	DCL (Device Clear)	Low	
	SPE (Serial Poll Enable)	Low	
	SPD (Serial Poll Disable)	Low	
アドレスによる コマンド	SDC (Selective Device Clear)	Low	ユニットをデフォルト状態に戻す。 装置をローカルに戻す。
	GTL (Go To Local)	Low	
アドレスによらない コマンド	UNL (Unlisten)	Low	全リスナをバスから解除する。 全トークをバスから解除する。
	UNT (Untalk)	Low	
コモンコマンド	—	High	IEEE-488.2コンパチブル機器を共通のオペレーションに プログラムする。
SCPIコマンド	—	High	SCPIコンパチブル機器を特定のオペレーションに プログラムする。

ユニバーサルマルチラインコマンド

ユニバーサルコマンドは、アドレッシングを必要としないマルチラインコマンドです。この種のコマンドを実行するように装備された全ての装置は、コマンドが送信されると、これらを同時に実行します。全てのマルチラインコマンドの場合と同様、これらのコマンドは、ATNが真の状態です。

LLO (Local Lockout) — LLOは、LOCALキーをロックアウトし、さらにそれに伴って全てのフロントパネルコントロール類をロックアウトするために機器へ送信されます。

DCL (Device Clear) — DCLは、機器をデフォルト状態に戻すのに使います。通常、機器は電源投入時の状態に戻ります。

SPE (Serial Poll Enable) — SPEは、シリアルポーリングシーケンスの最初のステップで、どの装置がサービスを要求したかを確認するのに使います。

SPD (Serial Poll Disable) — SPDは、コントローラがバス上の全装置をシリアルポールモードから解除するのに使います。通常、シリアルポールシーケンスの最後のコマンドです。

アドレスによるマルチラインコマンド

アドレスによるコマンドは、装置リスナアドレスを先行させなければならないマルチラインコマンドで、そうしてからでなければ、その機器は該コマンドに回答しません。コマンドとそれに先行するアドレスはどちらもATNが真の状態です。

SDC (Selective Device Clear) — SDCコマンドは、実質上はDCLコマンドと同じ機能を果たしますが、アドレッシングされた装置だけが回答します。原則として、機器は、SDCコマンドに回答すると電源投入時デフォルト状態に戻ります。

GTL (Go to Local) — GTLコマンドは、機器をリモートモードから解除するのに使います。機器によっては、LLOコマンドでロックアウトされていた場合、GTLでもフロントパネルコントロール類をアンロックできます。

GET (Group Exec Trigger) — GETコマンドは、装置コンフィギュレーションに依存する特定のアクション（たとえば、読取り）を実行するために装置をトリガするのに使います。GETはアドレスによるコマンドですが、多くの装置はアドレッシングなしでGETに応答します。

アドレスによるコマンド

アドレスによるコマンドには、2つの一次コマンドグループと1つの二次アドレスグループがあります。これらのコマンドをアサートするとき、ATNは真です。これらのコマンドは次の通りです。

LAG (Listen Address Group) — リスンコマンドは機器の一次アドレスから導出されます。装置をリスンにアドレッシングするのに使います。実際のコマンドバイトは、一次アドレスと\$20のORをとることによって得られます。

TAG (Talk Address Group) — トークコマンドは、一次アドレスから\$40とのORをとることによって導出されます。装置をトークにアドレッシングするのに使います。

SCG (Secondary Command Group) — このグループのコマンドは、付加的なアドレッシング機能を果たすものです。多くの装置（Model 7001を含む）はこれらのコマンドは使いません。

アドレスによらないコマンド

アドレスによらないコマンドは以下の2つあり、コントローラがバスからトーカまたはリスナを解除するのに使います。

UNL (Unlisten) — リスナは、このUNLコマンドによってリスナアイドル状態にされます。

UNT (Untalk) — 指定されていたトーカは、このUNTコマンドによってトーカアイドル状態にされます。

コモンコマンド

コモンコマンドはバス上の全装置に共通のコマンドです。これらのコマンドはIEEE-488.2規格に定められています。

原則として、これらのコマンドは、装置に対してリセットなどの共通のオペレーション実行するよう命じる1つまたはそれ以上のASCII文字として送信されます。これらのコマンドを送信するときATNは偽ですから、IEEE-488バスはこれらのコマンドをデータとして扱います。

SCPIコマンド

SCPIコマンドはバス上のそれぞれ装置に固有のコマンドです。これらのコマンドは機器メーカーによって定められ、Standard Commands for Programmable Instruments (SCPI) ConsortiumのSCPI規格に準拠しています。

原則として、これらのコマンドは、装置に対してレンジの設定やリレーの閉鎖などの特定のオペレーションを実行するよう命じる1つまたはそれ以上のASCII文字として送信されます。これらのコマンドを送信するときATNは偽ですから、IEEE-488バスはこれらのコマンドをデータとして扱います。

コマンドコード

データラインを使う各種のコマンドのコマンドコードを図G-3に示します。また、各種のコマンドの16進値と10進値を表G-2に示します。

図G-3
コマンドコード

D ₇	D ₆	D ₅	D ₄	Bits		COMMAND		COMMAND		COMMAND		PRIMARY ADDRESS		PRIMARY ADDRESS		PRIMARY ADDRESS		PRIMARY ADDRESS																						
D ₃	D ₂	D ₁	D ₀	D ₃	D ₂	D ₁	D ₀	D ₃	D ₂	D ₁	D ₀	D ₃	D ₂	D ₁	D ₀	D ₃	D ₂	D ₁	D ₀																					
				0(A)		0(B)		1(A)		1(B)		2(A)		2(B)		3(A)		3(B)		4(A)		4(B)		5(A)		5(B)		6(A)		6(B)		7(A)		7(B)						
0	0	0	0	0	0	0	0	DLE	GTL	NUL	SP	@	0	P	16	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				
0	0	0	1	0	1	0	1	DC1		SOH		A	1	Q	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0	0	1	0	0	1	0	2	DC2		STX	"	B	2	R	18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
0	0	1	1	0	1	1	3	DC3		ETX	#	C	3	S	19	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
0	1	0	0	0	0	0	4	DC4	SDC	EOT	\$	D	4	T	20	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
0	1	0	0	0	0	0	5	ENQ	PPC*	ENQ	%	E	5	U	21	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	
0	1	1	0	0	0	0	6	ACK		ACK	&	F	6	V	22	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
0	1	1	1	0	0	0	7	BEL		BEL	.	G	7	W	23	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
1	0	0	0	0	0	0	8	BS	GET	BS	(H	8	X	24	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
1	0	0	1	0	0	0	9	HT	TCT*	EM)	I	9	Y	25	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	0	1	0	0	0	0	10	LF		SUB	.	J	10	Z	26	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
1	0	1	1	0	0	0	11	VT		ESC	*	K	11	[27	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
1	1	0	0	0	0	0	12	FF		FS	,	L	12	\	28	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11
1	1	1	0	0	0	1	13	CR		GS	-	M	13]	29	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
1	1	1	1	0	0	1	14	SO		RS	_	N	14	^	30	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13
1	1	1	1	1	0	1	15	SI		US	/	O	15	~	UNL	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
1	1	1	1	1	1	1	15																																	

アドレスによる コマンド グループ (ACG)	ユニバーサル コマンド グループ (UCG)	リスン アドレス グループ (LAG)	トーク アドレス グループ (TAG)	二次 コマンド グループ (SDG)
一次 コマンド グループ (PCG)				

*PPC (PARALLEL POLL CONFIGURE), PPU
(PARALLEL POLL UNCONFIGURE), および
TCT (TAKE CONTROL) は MODEL 7001では使わない。
注: D₆ = D₁₀1...D₇ = D₁₀8
X = DON'T CARE

表G-2

16進および10進コマンドコード

コマンド	16進値	10進値
GTL	01	1
SDC	04	4
GET	08	8
LLO	11	17
DCL	14	20
SPE	18	24
SPD	19	25
LAG	20-3F	32-63
TAG	40-5F	64-95
SCG	6-7F	96-127
UNL	3F	63
UNT	5F	95

代表的なコマンドシーケンス

各種のマルチラインコマンドの場合、コマンドを正しく送信するために特定のバスシーケンスが生じなければなりません。特に、Model 7001がアドレスによるコマンドに応答するためには、正しいリスンアドレスをModel 7001へ送信しなければなりません。アドレスによるマルチラインコマンドを送信するための代表的なバスシーケンスを表G-3に示します。このケースでは、SDCコマンドがModel 7001へ送信されています。通常は、他にアクティブリスナが存在しないようにするためにシーケンスの一部としてUNLが送信されます。なお、リスンコマンドとSDCコマンドバイト自体を送信するとき、どちらの場合もATNは真です。

代表的なコモンコマンドシーケンスを表G-4に示します。このケースでは、Model 7001がアドレスリングされているときATNは真ですが、コモンコマンドストリングを送信しているときはHighに設定されます。

IEEEコマンドグループ

Model 7001によってサポートされているコマンドグループを表G-5に示します。なお、コモンコマンドとSCPIコマンドは表中には含まれていません。

表G-3

アドレスによるコマンドを送信するための代表的なシーケンス

ステップ	コマンド	ATNの状態	データバス		
			ASCII	16進値	10進値
1	UNL	Lowに設定	?	3F	63
2	LAG*	Lowのまま	0	27	39
3	SDC	Lowのまま	EOT	04	4
4		Highに戻る			

*一次アドレス = 7とする。

表G-4

代表的なコモンコマンドシーケンス

ステップ	コマンド	ATNの状態	データバス		
			ASCII	16進値	10進値
1	UNL	Lowに設定	?	3F	63
2	LAG*	Lowのまま	0	27	39
3	Data	Highに設定	*	2A	42
4	Data	Highのまま	R	52	82
5	Data	Highのまま	S	53	83
6	Data	Highのまま	T	54	84

*一次アドレス = 7とする。

表G-5

IEEEコマンドグループ

ハンドシェークコマンドグループ	
	NDAC = NOT DATA ACCEPTED
	NRFD = NOT READY FOR DATA
	DAV = DATA VALID
ユニバーサルコマンドグループ	
	ATN = ATTENTION
	DCL = DEVICE CLEAR
	IFC = INTERFACE CLEAR
	REN = REMOTE ENABLE
	SPD = SERIAL POLL DISABLE
	SPE = SERIAL POLL ENABLE
アドレスコマンドグループ	
LISTEN	LAG = LISTEN ADDRESS GROUP
	MLA = MY LISTEN ADDRESS
	UNL = UNLISTEN
TALK	TAG = TALK ADDRESS GROUP
	MTA = MY TALK ADDRESS
	UNT = UNTALK
	OTA = OTHER TALK ADDRESS

表G-5 (続き)

IEEEコマンドグループ

アドレスコマンドグループ	
	ACG = ADDRESS COMMAND GROUP
	GTL = GO TO LOCAL
	SDC = SELECTIVE DEVICE CLEAR
ステータスコマンドグループ	
	RQS = REQUEST SERVICE
	SRQ = SERIAL POLL REQUEST
	STB = STATUS BYTE
	EOI = END

付録H

IEEE-488への適合について

概要

IEEE-488.2規格では、Model 7001が規格をどう実施するかについての明確な情報が要求されます。IEEE-488.2規格 (Std. 488.2-1987) の4.9項にはドキュメンテーション要求事項が示されています。表H-1は、これらの要求事項をまとめて示すとともに、それぞれ該当する情報を示すか、本書中のその情報の参照箇所を示すものです。

表H-1

IEEE-488ドキュメンテーション要求事項

	要求事項	情報または参照箇所
(1)	IEEE-488インタフェースファンクションコード。	付録C参照。
(2)	アドレスが0~30の範囲外に設定されたときの7001の動作。	エラーメッセージが発生。前のアドレスが保持される。
(3)	有効アドレスが入力されたときの7001の動作。	エラーメッセージなし。ディスプレイがGIB SETUP MENUに戻る。
(4)	電源投入時のセットアップ状態。	*RST = 5.9.10項参照。 :SYST:PRESおよび :SYST:POS = 5.10.7項参照。

表H-1

IEEE-488ドキュメンテーション要求事項

	要求事項	情報または参照箇所
(5)	メッセージ交換オプション。	
(a)	入力バッファサイズ。	256バイト。
(b)	複数の応答メッセージを返す問合せ。	なし。
(c)	パーズされると応答を生じる問合せ。	全ての問い合わせ (コモンコマンドおよびSCPI)。
(d)	読み取られると応答を生じる問い合わせ。	なし。
(e)	連結コマンド。	:TRIG:COUNTは:TRIG:COUNT:AUTOに連結される。Autoをイネーブルすると、カウントがスキャンリスト長に設定される。
(6)	SCPIコマンドに必要なファンクショナルエレメント。	SCPIコマンドサブシステムの各表に記載されている (付録B参照)。
(7)	ブロックデータのバッファサイズ制限。	ブロック表示メッセージについて最大20文字。

表H-1

IEEE-488ドキュメンテーション要求事項

	要求事項	情報または参照箇所
(8)	シンタックス上での制約。	5.10項 (SCPIコマンドサブシステム) に記載されている。
(9)	各問い合わせコマンドごとの対応シンタックス。	5.9 (コモンコマンド) および5.10項 (SCPIコマンドサブシステム) に記載されている。
(10)	規格の規則に従わない装置間メッセージ転送。	なし。
(11)	ブロックデータ応答サイズ。	なし。
(12)	7001で使われるコモンコマンド。	付録A参照。
(13)	校正問い合わせ情報。	該当せず。
(14)	*DDT用のトリガマクロ。	該当せず。
(15)	マクロ情報。	該当せず。
(16)	*IDN (識別) に対する応答。	5.9.5項参照。

表H-1

IEEE-488ドキュメンテーション要求事項

	要求事項	情報または参照箇所
(17)	*PUDおよびPUD?用の記憶領域。	該当せず。
(18)	*RDTおよび*RDT?の資源記述。	該当せず。
(19)	*RST,*RCLおよび*SAV効果。	*RST = 5.9.10項参照。 *RCL = 5.9.9項参照。 *SAV = 5.9.11項参照。
(20)	*TST情報。	第7節参照。
(21)	ステータスレジスタ構造。	5.6項参照。
(22)	シーケンシャルまたはオーバーラップコマンド。	:INITおよび:INIT:CONT ONをのぞき全てシーケンシャル。
(23)	オペレーション完了メッセージ。	*OPC = 5.9.6項参照。 *OPC? = 5.9.7項参照。 *WAI = 5.9.17項参照。

付録I

SCPIへの適合について

概要

Model 7001はSCPIバージョン1992.0に準拠しています。

表I-1

Model 7001で使われるSCPI認定コマンドのシンタックス

コマンド	説明
<p>:DISPlay</p> <p>[:WINDow[1]]</p> <p>:TEXT</p> <p>:DATA<a></p> <p>:STATe</p> <p>:STATe?</p> <p>:WINDow2</p> <p>:TEXT</p> <p>:DATA<a></p> <p>:STATe</p> <p>:STATe?</p>	<p>メッセージをディスプレイの上部に位置指定するためのパス： ユーザテキストメッセージを制御するためのパス： 最大20文字を使ってASCIIメッセージ“a”を定義する。 メッセージをイネーブル(1またはON)あるいはディスエーブル(0またはOFF)する。 メッセージステータス(イネーブル状態またはディスエーブル状態)を要求する。</p> <p>メッセージをディスプレイの下部に位置指定するためのパス： ユーザテキストメッセージを制御するためのパス： 最大35文字を使ってASCIIメッセージ“a”を定義する。 メッセージをイネーブル(1またはON)あるいはディスエーブル(0またはOFF)する。 メッセージステータス(イネーブル状態またはディスエーブル状態)を要求する。</p>
<p>[:ROUTe]</p> <p>:CLOS<list></p> <p>:STATe?</p> <p>:OPEN<list> ALL</p> <p>:SCAN<list></p> <p>:POINTs?</p>	<p>指定されたチャンネルを閉じるためのパスおよびコマンド： 閉じているチャンネルを要求する。</p> <p>指定された（または全部の）チャンネルを開く。</p> <p>チャンネルリストを指定するためのパスおよびコマンド： スキャンリスト中のチャンネルの数を要求する。</p>
<p>:STATus</p> <p>:OPERation</p> <p>[:EVENTi]?</p> <p>:CONDition?</p> <p>:PTRansition<n></p> <p>:PTRansition?</p> <p>:NTRansition<n></p> <p>:NTRansition?</p> <p>:ENABle<n></p> <p>:ENABle?</p> <p>:ARM</p> <p>[:EVENTi]?</p> <p>:CONDition?</p> <p>:PTRansition<n></p> <p>:PTRansition?</p> <p>:NTRansition<n></p> <p>:NTRansition?</p> <p>:ENABle<n></p> <p>:ENABle?</p>	<p>オペレーションイベントレジスタをコントロールするためのパス： ステータスレジスタを読み取る。 条件レジスタを読み取る。 正トランジションレジスタをプログラムする（0～1122）。 正トランジションレジスタを読み取る。 負トランジションレジスタをプログラムする（0～1122）。 負トランジションレジスタを読み取る。 イネーブルレジスタをプログラムする（0～1122）。 イネーブルレジスタを読み取る。</p> <p>アームイベントレジスタをコントロールするためのパス： ステータスレジスタを読み取る。 条件レジスタを読み取る。 正トランジションレジスタをプログラムする（0または2）。 正トランジションレジスタを読み取る。 負トランジションレジスタをプログラムする（0または2）。 負トランジションレジスタを読み取る。 イネーブルレジスタをプログラムする（0または2）。 イネーブルレジスタを読み取る。</p>

表1-1

Model 7001で使われるSCPI認定コマンドのシンタックス

コマンド	説明
:SEQUence [:EVEVt]? :CONDition? :PTRansition<n> :PTRansition? :NTRansition<n> :NTRansition? :ENABle<n> :ENABle? :TRIGger [:EVENTt]? :CONDition? :PTRansition<n> :PTRansition? :NTRansition<n> :NTRansition? :ENABle<n> :ENABle? :PRESet :QUEue [:NEXT]? :ENABle<list> :QUESTionable [:EVENTt]? :CONDition? :PTRansition<n> :PTRansition? :NTRansition<n> :NTRansition? :ENABle<n> :ENABle?	シーケンスイベントレジスタをコントロールするためのパス： ステータスレジスタを読み取る。 条件レジスタを読み取る。 正トランジションレジスタをプログラムする (0~6)。 正トランジションレジスタを読み取る。 負トランジションレジスタをプログラムする (0~6)。 負トランジションレジスタを読み取る。 イネーブルレジスタをプログラムする (0~6)。 イネーブルレジスタを読み取る。 トリガイベントレジスタをコントロールするためのパス： ステータスレジスタを読み取る。 条件レジスタを読み取る。 正トランジションレジスタをプログラムする (0または2)。 正トランジションレジスタを読み取る。 負トランジションレジスタをプログラムする (0または2)。 負トランジションレジスタを読み取る。 イネーブルレジスタをプログラムする (0または2)。 イネーブルレジスタを読み取る。 ステータスレジスタをデフォルト状態に戻す。 エラー待ち行列にアクセスするためのパス： 一番新しいエラーメッセージを要求する。 待ち行列に入るエラーおよびステータスメッセージを指定する。 被疑イベントレジスタをコントロールするためのパス： ステータスレジスタを読み取る。 条件レジスタを読み取る。 正トランジションレジスタをプログラムする。 正トランジションレジスタを読み取る。 負トランジションレジスタをプログラムする。 負トランジションレジスタを読み取る。 イネーブルレジスタをプログラムする。 イネーブルレジスタを読み取る。
:SYSTEM :ERRor? :PRESet :VERSion?	エラー/ステータス待ち行列にメッセージを要求する。 Model 7001をデフォルト状態に戻す。 SCPI規格の改訂水準を要求する。

表I-1

Model 7001で使われるSCPI認定コマンドのシンタックス

コマンド	説明
<p>Trigger commands</p> <p>:INTiate [:IMMediate] :CONTinuous</p> <p>:ABORt</p> <p>:ARM[:SEquence[1]] [:LAYer[1]] :IMMediate :COUNT<n> INFInite :COUNT? :SOURce HOLD IMMediate MANual BUS EXTernal :SOURce? :SIGNal :LAYer2 :IMMediate :COUNT<n> INFInite :COUNT? :DELAy<num> :DELAy? :SOURce HOLD IMMediate MANual BUS EXTernal TIMer :SOURce? :SIGNal :TIMer<num> :TIMer?</p> <p>:TRIGger[:SEquence[1]] :IMMediate :COUNT<n> INFInite :AUTo ON/OFF :COUNT? :DELAy<num> :DELAy? :SOURce HOLD IMMediate MANual BUS EXTernal TIMer :SOURce? :SIGNal :TIMer<num> :TIMer?</p>	<p>サブシステムコマンドパス： 1トリガ（スキャン）サイクルを開始させる。 トリガシステムの連続起動をイネーブル（1またはON） またはディスエーブル（0またはOFF）する。</p> <p>トリガシステムをリセットし、アイドル状態に入る。</p> <p>スキャンを設定するためのサブシステムコマンドパス： スキャンのアームレイヤをプログラムするためのパス： コントロールソースをループアラウンドする。 アーム回数（1～9999またはINFInite）をプログラムする。 プログラムされたアームカウントを要求する。 Arm Layer 1コントロールソースを選択する。</p> <p>プログラムされたソース（イベント）を要求する。 コントロールソースをループアラウンドする。</p> <p>スキャンのスキャンレイヤをプログラムするためのパス： コントロールソースをループアラウンドする。 スキャン回数（1～9999またはINFInite）をプログラムする。 プログラムされたスキャンカウントを要求する。 ダイレイ（0～99999.999秒）をプログラムする。 プログラムされたダイレイを要求する。 Arm Layer 2コントロールソースを選択する。</p> <p>プログラムされたソース（イベント）を要求する。 スキャンコントロールソースを1回だけバイパスする。 タイマ時間間隔（0～99999.999秒）を設定する。 プログラムされたタイマ時間間隔を要求する。</p> <p>スキャンのチャンネルレイヤをプログラムするためのパス： チャンネルを即時にスキャンする。（コントロールソースをループ アラウンド）。 チャンネル数（1～9999またはINFInite）をプログラムする。 scan-list-lengthを使う。 プログラムされたチャンネルカウントを要求する。 ダイレイ（0～99999.999秒）をプログラムする。 プログラムされたダイレイを要求する。 Trigger Layerコントロールソースを選択する。</p> <p>プログラムされたソース（イベント）を要求する。 チャンネルコントロールソースを1回だけバイパスする。 タイマ時間間隔（0～99999.999秒）を設定する。 プログラムされたタイマ時間間隔を要求する。</p>

表I-2

Model 7001で使われる非SCPIコマンド

コマンド	説明
:DISPlay :SMESsage :SMESsage?	表示されたステータスメッセージをイネーブル(1またはON)またはディスエーブル(0またはOFF)する。 ステータスメッセージディスプレイの状態を要求する。
:OUTPut :TTLn :LSEnse AHIGH ALLOW :LSEnse?	出力ライン “n” (1~4)を指定するためのパスおよびコマンド： 極性、すなわち、アクティブハイ (AHIGH)、またはアクティブロー (ALLOW) を設定する。 指定された出力ライン (n) を要求する。
[:ROUte] :SCAN? :FCHannel<list> :FCHannels? :CONFigure: :SCHannel :SCHannel? :CPAir :CPAir? :SLOT1 :CTYPE<type> :CTYPE? :POLE<n> :POLE? :STIME<num> :STIME? :SLOT2 :CTYPE<type> :CTYPE? :POLE<n> :POLE? :STIME<num> :STIME? :MEMory :SAVe M<num> :RECall M<num>	プログラムされたスキャンリストを要求する。 閉じてはならないチャンネル (禁止チャンネル) を指定する。 制限チャンネルのリストを要求する。 コンフィギュレーションコマンドパス： Single Channelをイネーブル(1またはON)あるいはディスエーブル(0またはOFF)する。 Single Channelモードのステータスを要求する。 Channel Pairingをイネーブル(1またはON)あるいはディスエーブル(0またはOFF)する。 Channel Paringのステータスを要求する。 CARD 1を設するためのパス： スロットに装着されたカードタイプを指定する。 スロットに装着されたカードタイプを要求する。 ポールモード (1、2または4) を選択する。 プログラムされたポール数を要求する。 秒単位のリレーセットリング時間を (0~99999.999) を設定する。 プログラムされたりレーセットリング時間を要求する。 CARD 2を設定するためのパス： スロットに装着されたカードタイプを指定する。 スロットに装着されたカードタイプを要求する。 ポールモード (1、2または4) を選択する。 プログラムされたポール数を要求する。 秒単位のリレーセットリング時間を (0~99999.999) を設定する。 プログラムされたりレーセットリング時間を要求する。 メモリをプログラムするためのパス： チャンネルパターンをメモリロケーション (1~100) にセーブする。 メモリロケーション (1~100) からチャンネルパターンをリコールする。

表I-2

Model 7001で使われる非SCPIコマンド

コマンド	説明
:SENSe[1] :TTL1 :DATA?	入力ポートのためのコマンドバス。 デジタル入力ポートを読み取る。
:SOURce :TTL1 [:LEVel] [:LEVel]? :TTL2 [:LEVel] [:LEVel]? :TTL3 [:LEVel] [:LEVel]? :TTL4 [:LEVel] [:LEVel]?	デジタル出力ポートのビットをプログラムするためのコマンドバス： 出力ポートのビット1をプログラムするためのバス： セット (1またはON) あるいはリセット (0またはOFF)。 ビット1のステータスを要求する。 出力ポートのビット2をプログラムするためのバス： セット (1またはON) あるいはリセット (0またはOFF)。 ビット2のステータスを要求する。 出力ポートのビット3をプログラムするためのバス： セット (1またはON) あるいはリセット (0またはOFF)。 ビット3のステータスを要求する。 出力ポートのビット4をプログラムするためのバス： セット (1またはON) あるいはリセット (0またはOFF)。 ビット4のステータスを要求する。
:SYSTEM :POSetup RST PRESet SAV0 SAV1 SAV2 SAV3 SAV4 SAV5 SAV6 SAV7 SAV8 SAV9 :POSetup?	電源投入時セットアップを選択する。 電源投入時セットアップを要求する。
Trigger commands: :ARN[:SEquence[1]] [:LAYer[1]] :SOURce TLINk :TCONfigure :DIRection SOURce ACCeptor :DIRection? :ASYNchronous :ILINe<n> :ILINe? :OLINe<n> :OLINe?	スキャンを設定するためのサブシステムコマンドバス： スキャンのアームレイヤをプログラムするためのバス： Arm Layer 1コントロールソースを選択する。 トリガを設定するためのバス： バイパスをイネーブル (SOUR) またはディスエーブル (ACC) する。 プログラムされた方向を要求する。 非同期Trigger Linkモードを設定するためのバス： 入力ライン (1~6) を選択する。 プログラムされた入力ラインを要求する。 出力ライン (1~6) を選択する。 プログラムされた出力ラインを要求する。

表1-2

Model 7001で使われる非SCPIコマンド

コマンド	説明
LAYer2 :SOURce TLINk :TCONfigure :DIRection SOURce ACCeptor :DIRection? :ASYNchronous :ILINe<n> :ILINe? :OLINe<n> :OLINe?	スキャンのスキャンレイヤをプログラムするためのパス。 Arm Layer2コントロールソースを選択する。 Trigger Linkを設定するためのパス： バイパスをイネーブル (SOUR) またはディスエーブル (ACC) する。 プログラムされた方向を要求する。 非同期モードを設定するためのパス： 入力ライン (1~6) を選択する。 プログラムされた入力ラインを要求する。 出力ライン (1~6) を選択する。 プログラムされた出力ラインを要求する。
:TRIGger[:SEQuence[1]] :SOURceTLINk :TCONfigure :PROTOcol ASYNchronous SSYNchronous :PROTOcol? :DIRection SOURce ACCeptor :DIRection? :ASYNchronous :ILINe<n> :ILINe? :OLINe<n> :OLINe? :SSYNchronous :LINe<n> :LINe?	スキャンのチャンネルレイヤをプログラムするためのパス。 Trigger Layerコントロールソースを選択する。 トリガを設定するためのパス： プロトコル (非同期または半同期Trigger Link) を選択する。 プログラムされたプロトコルを要求する。 バイパスをイネーブル (SOUR) またはディスエーブル (ACC) する。 プログラムされた方向を要求する。 非同期Trigger Linkモードを設定するためのパス： 入力ライン (1~6) を選択する。 プログラムされた入力ラインを要求する。 出力ライン (1~6) を選択する。 プログラムされた出力ラインを要求する。 半同期Trigger Linkモードを設定するためのパス： トリガライン (1~6) を選択する。 プログラムされたトリガラインを要求する。

Specifications are subject to change without notice.

All Keithley trademarks and trade names are the property of Keithley Instruments, Inc. All other trademarks and trade names are the property of their respective companies.



Keithley Instruments, Inc.

28775 Aurora Road • Cleveland, Ohio 44139 • 440-248-0400 • Fax: 440-248-6168
1-888-KEITHLEY (534-8453) • www.keithley.com

Sales Offices: BELGIUM:

Bergensesteenweg 709 • B-1600 Sint-Pieters-Leeuw • 02-363 00 40 • Fax: 02/363 00 64

CHINA:

Yuan Chen Xin Building, Room 705 • 12 Yumin Road, Dewai, Madian • Beijing 100029 • 8610-6202-2886 • Fax: 8610-6202-2892

FINLAND:

Tietäjäsentie 2 • 02130 Espoo • Phone: 09-54 75 08 10 • Fax: 09-25 10 51 00

FRANCE:

3, allée des Garays • 91127 Palaiseau Cédex • 01-64 53 20 20 • Fax: 01-60 11 77 26

GERMANY:

Landsberger Strasse 65 • 82110 Germering • 089/84 93 07-40 • Fax: 089/84 93 07-34

GREAT BRITAIN:

Unit 2 Commerce Park, Brunel Road • Theale • Berkshire RG7 4AB • 0118 929 7500 • Fax: 0118 929 7519

INDIA:

Flat 2B, Willocrissa • 14, Rest House Crescent • Bangalore 560 001 • 91-80-509-1320/21 • Fax: 91-80-509-1322

ITALY:

Viale San Gimignano, 38 • 20146 Milano • 02-48 39 16 01 • Fax: 02-48 30 22 74

JAPAN:

New Pier Takeshiba North Tower 13F • 11-1, Kaigan 1-chome • Minato-ku, Tokyo 105-0022 • 81-3-5733-7555 • Fax: 81-3-5733-7556

KOREA:

FL., URI Building • 2-14 Yangjae-Dong • Seocho-Gu, Seoul 137-130 • 82-2-574-7778 • Fax: 82-2-574-7838

NETHERLANDS:

Postbus 559 • 4200 AN Gorinchem • 0183-635333 • Fax: 0183-630821

SWEDEN:

c/o Regus Business Centre • Frosundaviks Allé 15, 4tr • 169 70 Solna • 08-509 04 679 • Fax: 08-655 26 10

SWITZERLAND:

Kriesbachstrasse 4 • 8600 Dübendorf • 01-821 94 44 • Fax: 01-820 30 81

TAIWAN:

1FL., 85 Po Ai Street • Hsinchu, Taiwan, R.O.C. • 886-3-572-9077 • Fax: 886-3-572-9031