

SEMI E122-0703 (Reapproved 1109) テスト装置の特定装置モデル (TSEM) のスタンダード

本スタンダードは、global Information & Control Committee で技術的に承認されている。現版は 2009 年 9 月 4 日、global Audits and Reviews Subcommittee にて発行が承認された。2009 年 10 月に www.semi.org で、そして 2009 年 11 月に CD-ROM で入手可能となる。初版は 2003 年 3 月発行、前版は 2007 年 3 月に発行された。

注意: 本文書は、2003 年 3 月版 (0303) にて削除された SEMI E30.3 を置き換えたものである。

1 目的

1.1 本文書では、テスト装置の特定装置モデル (TSEM) を確立する。TSEM は、このクラスの装置に適用される装置特性および挙動から構成される。これらの特性と挙動は実装される必要がある。本文書の目的は、テスト装置を自動化された半導体工場に統合し易くすることである。本文書は、ファクトリオートメーションコントローラの観点から見たテスト装置の動作モデルを定義することによって、その目的を達成している。この定義は、スタンダードなホストインタフェースおよび装置の動作挙動を提供する。

2 範囲

2.1 本文書では、ホスト通信リンクから見た装置のビューを定義するが、装置の内部動作は定義しない。本文書には、このクラスのすべての装置の挙動の基礎としての独自のプロセッシング状態モデルが含まれる。

2.2 本文書は、プロセッシング状態モデル、収集イベント、アラームの文書化、リモートコマンド、変数項目、プロセスプログラム管理の処理の領域で、テスト装置の要求条件および能力を拡張している。

注意: このスタンダードはその使用に関連したすべての安全問題を取り扱うことを意図していない。このスタンダードの利用者は、その責任において、適切な安全および健康上実施すべき事柄を確立し、また使用前に法規制やその他の制限への適用性を判断するものである。

3 制限

3.1 通信

3.1.1 すべての TSEM 準拠の装置は、SEMI E30 の通信状態モデルに従う必要がある。さらに、TSEM 準拠の装置は、高速メッセージサービス (SEMI E37) 通信スタンダードをサポートし、TCP/IP でメッセージを送信してこのクラスの装置から入手可能な監視用のデータ量を最大にしなければならない。本仕様は、ホストアプリケーションと通信するテスト装置の挙動のみを取り扱う。

3.2 仮想テスター

3.2.1 本文書の § 7 では、仮想テスターの概念について説明する。本文書に記載されている要求条件は、各仮想テスターに、他の仮想テスターとは完全に独立して実装されなければならない。本文書の目的を達するために、「テスター」とは、§ 7 で定義される仮想テスターであると見なすことができる。物理テスターへの参照は、明示的に注記される。

4 参照スタンダードおよび文書

4.1 SEMI スタンダード

SEMI E30 — Generic Model for Communications and Control of Manufacturing Equipment (GEM)

SEMI E37 — High-Speed SECS Message Services (HSMS) Generic Services

SEMI E37.1 — High-Speed SECS Message Services Single Selected-Session Mode (HSMS-SS)

注意: 別途指示のない場合、引用されたすべての文書は最新の出版バージョンである。

5 用語

5.1 定義

5.1.1 校正器具 (calibration fixture) — システムの校正を実行するために必要な電気機械的な器具。校正器具は、異なる部品およびシリアル番号の複数の構成要素で構成できる。

5.1.2 クラス (class) — クラスはテスト結果の最も粗い見方 (ビュー) を表現する。各プロセスプログラムには、最低限、2 つのクラスが定義されていなければならない。1 つのクラスは良品ユニットを表現し、他方は不良品ユニットを表現する。

5.1.3 データログ (datalog) — テストプログラムの実行中に収集される個別のテスト測定結果のコレクション。

5.1.4 診断器具 (diagnostic fixture) — システムの診断を実行するために必要な電気機械的な器具。診断器具は、異なる部品およびシリアル番号の複数の構成要素で構成できる。

5.1.5 実行エリア (execution area) — 現行のプロセスプログラムの指示を実行する部分。

5.1.6 ハードビン (hard-bin) — ハードビンは、テスト結果の代表的な見方 (ビュー) を表現する。プロセスプログラム内で、各ハードビンは単一のクラスに関連付けられる。一般的に複数のハードビンが特定のクラスに関連付けられる。

5.1.7 マルチサイトテスト (multi-site testing) — テストプログラムを 1 回実行して複数のユニットをテストすること。各ユニットは、独自のテスト結果を有する。

5.1.8 ソフトビン (soft-bin) — ソフトビンは、テスト結果の最も詳細な見方 (ビュー) を表現する。プロセスプログラム内で、各ソフトビンは単一のハードビンに関連付けられる。一般的に複数のソフトビンが特定のハードビンに関連付けられる。

5.1.9 ステーションコントローラ (station controller (SC)) — ステーションコントローラは、テストシステムおよびユニットハンドリング装置 (ウェーハプローバ、パッケージハンドラ等) の動作を調整するソフトウェアから構成される。テストシステムのコンピュータまたは何らかの別のコンピュータ上に存在できる。1 つのステーションコントローラは、1 つまたは複数の仮想テスターを担当できる。

5.1.10 システム校正 (system calibration) — テストシステムをテストシステムの製造元のシステム仕様にするために必要なテストシステムプロセス。

5.1.11 テストボード (test-board) — テストされるユニットとテスターのリソースを一時的に電氣的に接触させることを可能にするために必要な電気機械的インタフェース。テストボードは複数の構成要素から構成できる。

5.1.12 テストヘッド (test-head) — テスターのリソースの 1 つ。テストヘッドは、ユニットとテスター間の電気機械的インタフェースである。

5.1.13 テストサイト (test-site) — 一時に 1 つのユニットを位置決めしてテストするテストボード上の位置。

5.1.14 テスターエグゼクティブ (tester executive) — テストプログラムの実行を制御するテスタープログラム。

5.1.15 テスト装置 (test equipment) — 一般的には、製造プロセスの間にパッケージングされたデバイスおよび／あるいはウェーハダイに電氣的なテストを行うためのメカニズムおよびコントロールを統合したもので構成される装置クラス。

5.1.16 ユニット (unit) — 電氣的にテストされる機能集積回路 (またはチップ)。

5.1.17 仮想テスター (virtual tester) — 仮想テスターは、ユニットやユニットの集合をテストするためにハードウェアのリソースを使用するテストプログラムに対して、テストシステム内で使用可能な物理ハードウェアの部分の動的な割り当てを記述する論理的概念である。

6 状態モデル

6.1 この目的は、装置の予測される動作状態を表現するために必要な装置固有のプロセス状態モデルおよびその他の状態モデルを定義して、ローカルオペレータに代わり、ホストによるトラッキングと制御を可能にすることである。

6.2 状態モデルの要求条件

6.2.1 本文書のプロセス状態モデルは、TSEM準拠のテスターを実装するために必要である。状態モデルは、プロセス状態モデルダイアグラム、プロセス状態定義、プロセス状態遷移表から構成される。状態モデルは、ホストから見た仮想テスターを表現するが、必ずしも実際のテスターの動作を表現するものではない。TSEM状態モデルのすべての遷移は、これらの遷移の要求条件を満たす実際の装置イベントに順次マッピングされる。特定の実装において、テスターがある状態に入ると、別の状態に遷移するためにTSEM状態モデルが必要とするすべての条件が既に満たされていることがある。この状況においては、テスターは、追加の動作がなくても必要な遷移をおこなう。

6.2.2 装置によっては、追加の状態を含める必要があることがある。しかし、追加の状態はTSEM定義の状態遷移を変更してはならない。TSEM状態間で期待されるすべての遷移が起こらなくてはならない。

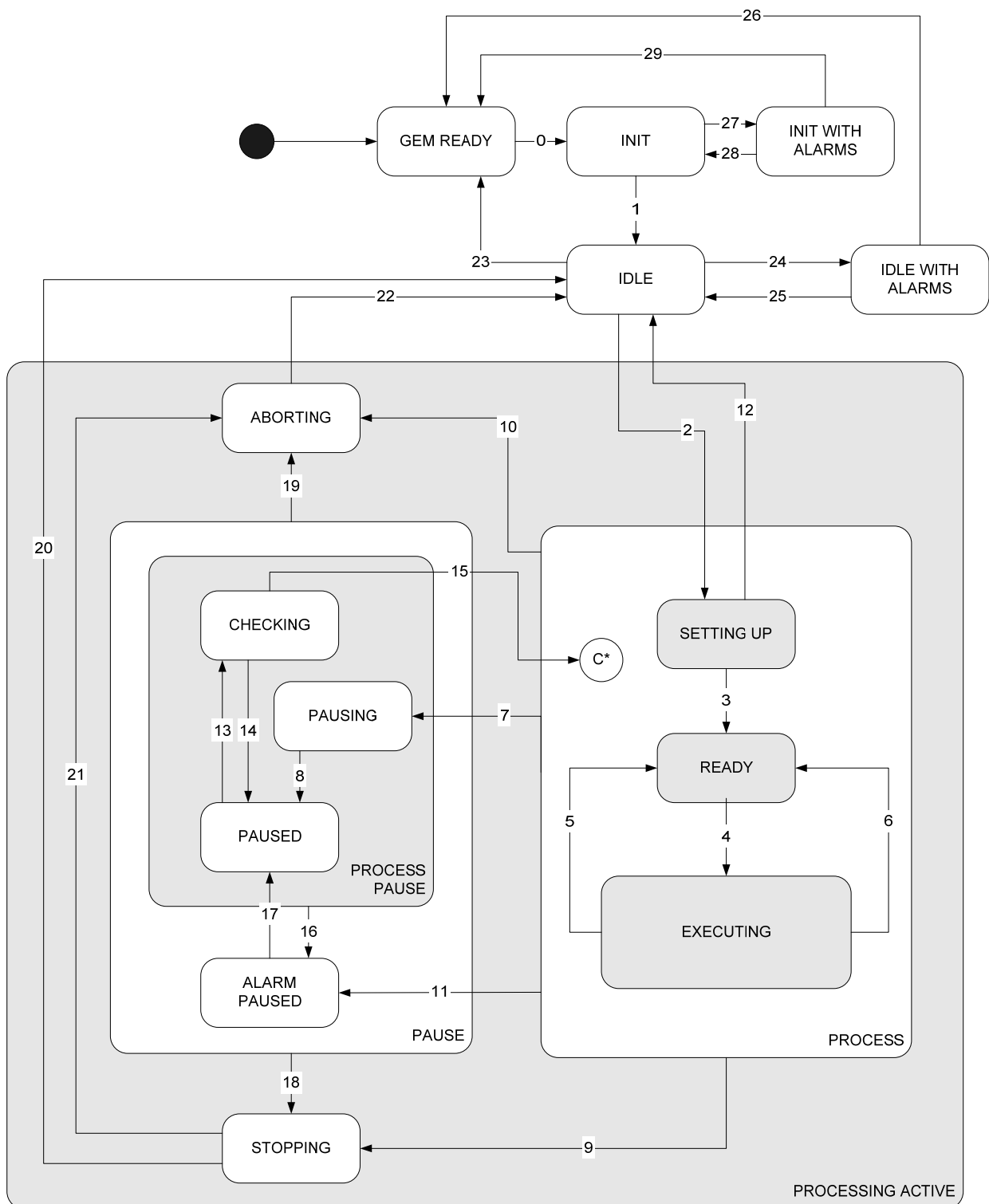


図 1
TSEM プロセス状態モデル

6.3 テスタープロセス状態の説明

6.3.1 **ABORTING (PROCESSING ACTIVEの下位状態)** — テスターがABORTコマンドを受信した。すべての動作は直ちに中断される。可能な場合テスターは、テスター自体と材料を「安全」な状態に移行させる適切な動作を実行中である。

6.3.2 **ALARM PAUSED (PAUSEの下位状態)** — PROCESSまたはPROCESS PAUSE状態でアラームが発生し、テスターはアラームがクリアされるまで待機中である。

6.3.3 **CHECKING (PROCESS PAUSEの下位状態)** — テスターは、プロセスプログラムに対しておこなわれた更新が有効であるか検証する。プロセスプログラムは、新しいプロセスプログラムの定義により結果を更新する。これは、SETTING UPで実行される手順と類似の手順である。プロセスプログラムの更新がない場合、これはパススルー(通過)の状態である。検証に成功すると、プロセスモデル条件表に基づいてプロセス状態が遷移する。

6.3.4 **EXECUTING (PROCESSの下位状態)** — テスターが材料を処理中である。

6.3.5 **IDLE** — テスターエグゼクティブがロードされ、テスターはコマンド待ちの状態である。IDLEでは、ALARMSやエラー状態がない。この状態が有効なアラームに入ると(ABORTING, STOPPING, SETTING UPからの異常終了の結果等により)、この状態は通過してIDLE WITH ALARMSに移行する。

6.3.6 **IDLE WITH ALARMS** — IDLE状態でアラームが発生し、テスターはすべてのアラームがクリアされるまで待機中である。アラーム条件をクリアするにはオペレータが必要となることもある。システムは、オペレータの介入が必要なアラームも含めすべてのアラームがクリアされるまでこの状態のままでなければならない。

6.3.7 **INIT** — テスターは初期化中である。テスターエグゼクティブをロード中である。

6.3.8 **INIT WITH ALARMS** — 初期化プロセス中にアラームが発生した。

6.3.9 **PAUSE (PROCESSING ACTIVEの下位状態)** — この状態は、プロセスプログラムが一時停止する下位状態の親である。

6.3.10 **PAUSED (PROCESS PAUSEの下位状態)** — PROCESS状態が中断され、テスターはコマンド待ちの状態である(RESUME, STOP, ABORT)。この状態では、オペレータは現在のプロセスプログラムの選択の条件を変更できる。

6.3.11 **PAUSING (PROCESS PAUSEの下位状態)** — 現在のユニットが存在すれば、それが完了すると現在の状態は中断され、テスターは「安全」な状態に移行する。

6.3.12 **PROCESS (PROCESSING ACTIVEの下位状態)** — この状態は、プロセスプログラムの準備および実行を参照する下位状態の親である。

6.3.13 **PROCESS PAUSE (PAUSEの下位状態)** — この状態は、テスター動作の一時停止を参照する下位状態の親状態である。PAUSE状態では、アラーム条件はない。

6.3.14 **PROCESSING ACTIVE** — この状態は、プロセスプログラムのアイデンティティを知る下位状態の親状態である。

6.3.15 **READY (PROCESSの下位状態)** — プロセスプログラムがロードされる。テスターはプロセスを開始する準備が整い、STARTコマンドを待っている。

6.3.16 **SETTING UP (PROCESSの下位状態)** — 選択されたプロセスプログラムをロード中である。テスターはプロセスを開始できるように条件を満たしつつある。これには、プロセスプログラムの初期化と有効化およびプロセスプログラム固有の校正が含まれる。これは、テスターによって独立に実行できる、あるいはオペレータやホストとの会話が必要な場合がある。

6.3.17 **STOPPING (PROCESSING ACTIVEの下位状態)** — テスターはプロセスを停止するように命令され、次の機会に停止しなければならない。材料、データ、制御システム等に関する必要なすべてのクリーンアップがこの状態内で完了する。

6.3.18 GEM READY — GEMセッションを実行中であり、ホストからSTART EXECコマンドを受け入れる準備が整っている。

6.4 TSEMプロセス状態遷移表

表 1 プロセス状態遷移表

#	現在の状態	トリガー	新状態	動作	コメント
0	GEM READY	GEMセッションがSTART EXEC コマンドを受信した。	INIT	この物理テスターの最初のSTART EXEC であれば、テスターエグゼクティブを起動する。	最初のテスターエグゼクティブがすべてのハードウェアの初期化を開始する。
1	INIT	アラームやエラー条件なしですべてのテスター初期化が完了する。	IDLE	なし	テスターエグゼクティブは実行中。
2	IDLE	プロセスプログラムが選択される。	SETTING UP	テスターに依存する。	セットアップがコミットされ、仮想テスターが定義された。
3	SETTING UP	すべてのセットアップ動作が完了し、START コマンドを受信する準備が整う。	READY	なし	選択されたプロセスプログラムが実行できる。この仮想テスターに必要なすべてのハードウェアが割り当てられる。テストプログラムがロードされる。
4	READY	テスターが有効なソースから START コマンドを受信する。	EXECUTING	テストサイトでユニットのテストを開始する。	有効なソースの定義は、アプリケーションに依存する。
5	EXECUTING	現在のユニットのプロセスが正常に終了した。	READY	テスターはテストデータの終了を処理する。	テストプログラム実行の「正常」終了。
6	EXECUTING	現在のユニットのプロセスが異常に終了した。	READY		テストプログラム実行の「異常」終了等。
7	PROCESS	テスターが有効なソースから PAUSE コマンドを受信した。	PAUSING	現在のユニットが完了した時点で現在の状態は中断される。	有効なソースの定義は、アプリケーションに依存する。
8	PAUSING	テスターが現在のユニットのプロセスを完了した。	PAUSED	テスターはコマンド待ちの状態である (RESUME, STOP, ABORT)。	なし
9	PROCESS	テスターが有効なソースから STOP コマンドを受信した。	STOPPING	テスターは、STOP を実行するために必要な動作を実行する。	有効なソースの定義は、アプリケーションに依存する。
10	PROCESS	テスターが有効なソースから ABORT コマンドを受信した。	ABORTING	テスターは、ABORT を実行するために必要な動作を実行する。	有効なソースの定義は、アプリケーションに依存する。
11	PROCESS	アラームが発生する。	ALARM PAUSED	PROCESS 動作を中断し、すべてのアラームがクリアされるまで待機する。	
12	SETTING UP	プロセスプログラムのロードに失敗したことを示す遷移を装置が起動した。	IDLE	ハードウェアリソースが解放される。プロセスプロセッサのロード失敗によりアラームとなった場合は、この遷移の後直ちに遷移 24 により IDLE WITH ALARMS に遷移する。	プロセスプログラムはロードされない。

#	現在の状態	トリガー	新状態	動作	コメント
13	PAUSED	テスターが RESUME コマンドを受信する。	CHECKING	RESUME コマンドが変数プロセスプログラムパラメータを格納している場合は、プロセスプログラムパラメータの検証が開始される。	RESUME プログラムにパラメータがない場合、CHECKING 状態はパススルーである。
14	CHECKING	新しいパラメータの設定でエラーが検出された。	PAUSED	テスターは、オペレータまたはホストによってパラメータが是正されるまで待機する。	有効なソースの定義は、アプリケーションに依存する。
15	CHECKING	パラメータチェックが成功裡に完了する。	条件表に基づく状態	テスターは PROCESS の実行を再開する。	これは、PROCESS 状態への条件付きの再遷移である (表 2 を参照)。パラメータがない場合、CHECKING 状態はパススルーである。
16	PROCESS PAUSE	アラームがセットされる。	ALARM PAUSED	テスターは、すべてのアラームがクリアされるまで待機する。	なし
17	ALARM PAUSED	すべてのアラームがクリアされる。	PAUSED	すべてのアラームと中断条件がクリアされたことをオペレータが確認した。 テスターは有効なコマンドを待っている。	なし
18	PAUSE	テスターが有効なソースから STOP コマンドを受信した。	STOPPING	テスターは、STOP を実行するために必要な動作を実行する。	有効なソースの定義は、アプリケーションに依存する。
19	PAUSE	テスターが有効なソースから ABORT コマンドを受信した。	ABORTING	テスターは、ABORT を実行するために必要な動作を実行する。	有効なソースの定義は、アプリケーションに依存する。
20	STOPPING	テスターが STOP を実行するために必要なすべての動作を完了する。	IDLE	この仮想テスター用のハードウェアリソースが解放される。	
21	STOPPING	テスターが有効なソースから ABORT コマンドを受信した。	ABORTING	テスターは、ABORT を実行するために必要な動作を実行する。	有効なソースの定義は、アプリケーションに依存する。
22	ABORTING	テスターが ABORT を実行するために必要な動作を完了した。	IDLE	この仮想テスター用のハードウェアリソースが解放される。 可能であれば安全でない状態が解決された。 すべてのアラームと中断条件がクリアされたことをオペレータが確認した。	
23	IDLE	テスターが有効なソースから STOP EXEC コマンドを受信する。	GEM READY	すべてのテスターからテスターエグゼクティブがアンロードされる。	有効なソースの定義は、アプリケーションに依存する。
24	IDLE	アラームがセットされる。	IDLE with ALARMS	テスターは、すべてのアラームがクリアされるまで待機する。	IDLE 状態に入ったときに ALARM が既に存在している場合、テスターは直ちに IDLE から IDLE WITH ALARMS に遷移する。
25	IDLE WITH ALARMS	装置のすべてのアラームがクリアされた。	IDLE	すべてのアラームと中断条件がクリアされたことをオペレータが確認した。	IDLE 状態にアラームはない。

#	現在の状態	トリガー	新状態	動作	コメント
26	IDLE WITH ALARMS	有効なソースからの STOP EXEC コマンドによりテスターエグゼクティブが停止した。	GEM READY	すべてのアラームと中断条件がクリアされたことをオペレータが確認した。 すべてのテスターからテスターエグゼクティブがアンロードされる。	有効なソースの定義は、アプリケーションに依存する。
27	INIT	テスターエグゼクティブのロード中にエラーが発生した。	INIT WITH ALARMS		なし
28	INIT WITH ALARMS	アラームがクリアされた。	INIT	すべてのアラームと中断条件がクリアされたことをオペレータが確認した。	なし
29	INIT WITH ALARMS	有効なソースからの STOP EXEC コマンドによりテスターエグゼクティブが停止した。	GEM READY	すべてのアラームと中断条件がクリアされたことをオペレータが確認した。 すべてのテスターからテスターエグゼクティブがアンロードされる。	有効なソースの定義は、アプリケーションに依存する。

6.5 プロセスモデル条件表

表 2 プロセス再開条件

条件	次の状態
チェックにより、プロセスプログラム条件が変更されたことを決定する。	SETTING UP
前の状態が EXECUTING であり、プロセスプログラム条件が変更されていないこと。	READY
前の状態が READY であり、プロセスプログラム条件が変更されていないこと。	READY
前の状態が SETTING UP であった。	SETTING UP

7 仮想テスター

7.1 一般定義

7.1.1 仮想テスターは、本文書で示す状態モデルに従って独立したテスターとして動作できるテストプログラムの単一のロードインスタンス(そのインスタンスに関連付けられるテストサイト、テストシステムのリソース、テストヘッドの数に関わらず)として定義される。

7.1.2 別のあるいは共有されたテストシステムのハードウェアリソースを使用する同じテストプログラムの別のインスタンスをロードすると、別の仮想テスターが確立される。そのテストプログラムの複数のコピーは、それぞれ独立している。ここで言うプログラムの独立とは、ある仮想テスタープログラムが実行されるために、別の仮想テスタープログラムの存在を期待したり、頼らないということである。

7.1.3 単一のテストプログラム専用でなければならない 1 つのテストヘッドの単一のテストシステムについては、物理テスターと仮想テスターは同じであり、単一の GEM 通信インタフェースが必要である。各テストヘッドが一意的なプロセスプログラムを自立的に実行できる 2 つのテストヘッドのテストシステムでは、2 つの GEM 通信セッションにより、2 つの仮想テスターが同時に動作できる。テストヘッドをそれぞれが一意的なプログラムを自立的に実行できる複数のセグメントに分割できる場合、これらのセグメントのそれぞれについて 1 つの仮想テスターおよびそれに合致する 1 つの GEM 通信セッションが存在する。

7.1.4 随時動作する仮想テスターの数は、テストシステムの構成方法(ハードウェアとソフトウェア)によって異なり、1 つから特定のテストシステムの最大能力の範囲となる。GEM 通信セッションの数は、テスターがサポートする仮想テスターの数と等しい。

7.2 GEM通信セッション

7.2.1 GEM通信ホスト接続からは、1つのプログラムと仮想テスターしか存在せず、ホストのGEM通信接続は他の仮想テスターのことは知らず、見えない。独立性は、電源や共有ソフトウェアライブラリ等のように、仮想テスター間で共有するリソースからテスターエグゼクティブやテストハードウェアを除外しない。

7.2.2 各仮想テスターに対して、テストシステムは独立したGEMクライアント接続を提供しなければならない。テストシステムコントローラは、複数のクライアント接続を管理する責任がある。

7.2.3 各仮想テスターに対して、ステーションコントローラは、単一のテスターGEMクライアントへの一意なホスト接続を確立する。テストシステムコントローラは、複数のホスト接続を管理する責任がある。

7.2.4 テストシステムは、設定された最大数の許容可能で一意なGEMホスト/クライアント接続を有する。これらのGEM通信接続の接続およびポートIDの最大数(n)は、ステーションコントローラが知っている定数であり、そのコントローラの起動後にテストシステムコントローラから入手できる。

7.2.5 テストシステムが許容するGEM通信セッションの最大数も、テスターリソースの数と割り当てに関わらず、テストシステムが許容する仮想テスターの最大数である。

7.2.6 各仮想テスターに対して、仮想テスターホスト/クライアント接続は、独立したGEM通信セッションである。次の階層構造の関係を確立できる。

7.2.7 GEM通信セッション(1 ... n)は、仮想テスター(1 ... n)を確立するテスターハードウェアリソース(x ... y)を使用するプログラムロードインスタンス(1 ... n)を許容する。この関係では、割り当ての順序は示されない。ステーションコントローラは、プログラムをロードするために使用可能な任意のGEM通信ポートを選択し、仮想テスターを確立できる。

7.2.8 テストシステムの物理的限界に到らない限り、複数のステーションコントローラが単一の物理テストシステムに対応し、各ステーションコントローラが独立の仮想テスターを確立するようにできる。

7.3 仮想テスターおよび物理テスターに対するGEM通信の同時性

7.3.1 GEM通信クライアントインタフェースは、常時、関連する仮想テスターの実際のリアルタイムの状態を表現しなければならない。

7.3.2 ローカルオペレータの操作等、GEM通信外部からのイベントにより、状態遷移を発生させることができる。

7.3.3 物理テスターの(任意のソースからの)イベントに対する仮想テスターの同期が必要である。

7.3.4 物理テスター同期に対するこの仮想テスターの方法および有効性は、TSEMの範囲外であり、テスターベンダーの責任である。

7.4 GEM通信セッションおよびアラーム

7.4.1 複数の仮想テスターに影響するアラーム状態は、影響を受けるすべての仮想テスターに通信しなければならない。

7.5 状態モデルの意味合い

7.5.1 テストシステムのワークステーションの電源が投入され、ワークステーションのオペレーティングシステム(OS)のロードに成功すると、(n)個のGEM通信ポートIDが確立され、ステーションコントローラが使用できるようになる。各GEM通信ポートは、可能性のある仮想テスターを代表する。各GEM通信ポートは、完全に独立したプロセス状態モデルがあり、それによってGEM READY状態に初期化される。

7.5.2 ステーションコントローラは、GEM通信セッションを使用可能な任意のGEM通信ポートに接続し、確率でできる。ステーションコントローラがGEM通信セッションを確立すると、リモートコマンドSTART EXECによってテスターエグゼクティブがロードされ、ロードに成功すると、そのGEM通信セッションに対するプロセス状態モデルまたは仮想テスターはIDLE状態に遷移する。この遷移は、他のGEM通信セッションとは完全に独立している。他のGEMセッションは、元の状態のままである。