**実行順序の制御と表示**

*実行順序*モデルについては、Simulink順序である®は、シミュレーション中にブロックの出力メソッドを呼び出します。Simulinkは、モデルの更新中にこの順序を決定します。これは、[ **モデリング** ]タブの[ **モデルの更新** ]をクリックして開始でき ます。Simulinkは、シミュレーション中にモデルも更新します。

実行順序を設定することはできませんが、非仮想ブロックに優先順位を割り当てて、対応するブロック線図内の他のブロックに対する実行順序を示すことができます。Simulinkは、データの依存関係との競合がない限り、ブロックの優先順位設定を尊重しようとします。設定した優先順位の結果を確認したり、モデルをデバッグしたりするには、非バーチャルブロックとサブシステムの実行順序を表示して確認します。

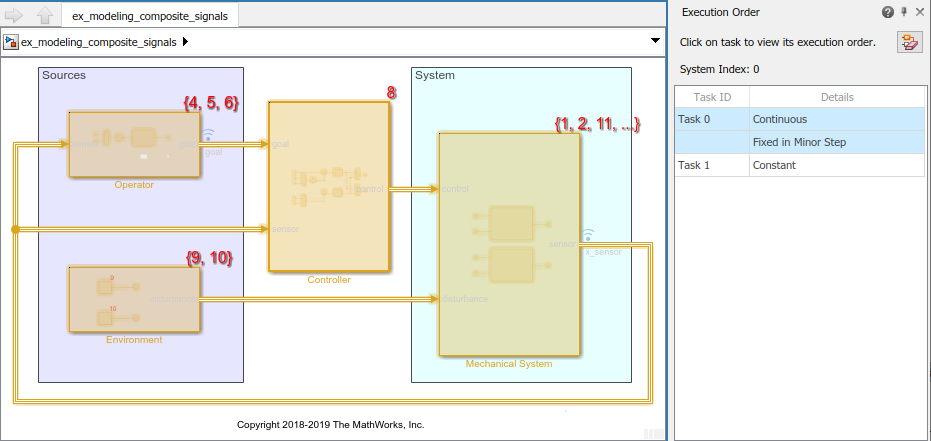
**注意**

ブロックメソッドと実行の詳細については、以下を参照してください。

* [ブロックメソッド](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/slref/simulink-concepts-simulation.html&usg=ALkJrhhowiICKhD9AHMo5iJUSSGHetFGjQ#mw_e75508d7-2339-4834-9a96-5a7ee9cbc8eb)
* [条件付きで実行されるサブシステムの概要](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/ug/about-conditional-subsystems.html&usg=ALkJrhjCLMB0XY6Rv36Bp-0yrlXOmS5IcQ)

**実行順序ビューア**

上の**デバッグ** ]タブ、選択**情報オーバーレイ** > **実行順序を**。実行順序ビューアーがSimulinkエディターの右側のペインに開きます。



実行順序ビューアには、現在のシステムのタスクのリストが表示されます。

各**システムインデックス**値は、非バーチャルサブシステム内のブロックなど、他のブロックから独立して実行されるブロックのグループに対応します。モデル内では、**システムインデックスの**値は一意です。モデル参照階層では、同じ**システムインデックス**値を複数回使用できますが、その階層内の各モデル内で一意のままです。サブシステムがモデル内の同じシステムの一部であるかどうかを判断するには、それらの**システムインデックス**値を比較します。

**[タスクID** ]列にリストされている各タスクは、サンプルレートを共有するブロックのグループに対応しています。固定ステップサイズの場合、[ [**各離散レートを個別のタスク**](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/gui/treat-each-discrete-rate-as-a-separate-task.html&usg=ALkJrhgvcy4ua4jRXIO5RWcWuFAHnL7_PA)構成パラメーター[**として扱う]**](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/gui/treat-each-discrete-rate-as-a-separate-task.html&usg=ALkJrhgvcy4ua4jRXIO5RWcWuFAHnL7_PA)は、Simulinkが離散レートのブロックを1つまたは複数のタスクで実行するかどうかを決定します。

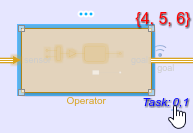
実行順序ビューアでタスクを選択すると、タスクに属するブロックが強調表示され、その実行順序が表示されます。既定では、Simulinkはアクティブなブロック線図に対応する最初のタスクを選択します。

アクティブなブロック線図に、特定のタスクの一部として実行されるブロックが含まれていない場合、実行順序ビューアーでタスクを選択できません。アクティブなブロック線図には、このタスクの一部であるInportブロックなどの仮想ブロックが含まれている場合があります。

ハイライトと実行の順序を非表示にするには、[ **ハイライト**を**クリア** ]ボタンをクリックします。https://de.mathworks.com/help/simulink/ug/tbs_clear_highlighting_icon.png。

**ブロックからタスクへのナビゲーション**

ブロックが実行されるタスクを表示するには、ブロックをクリックします。



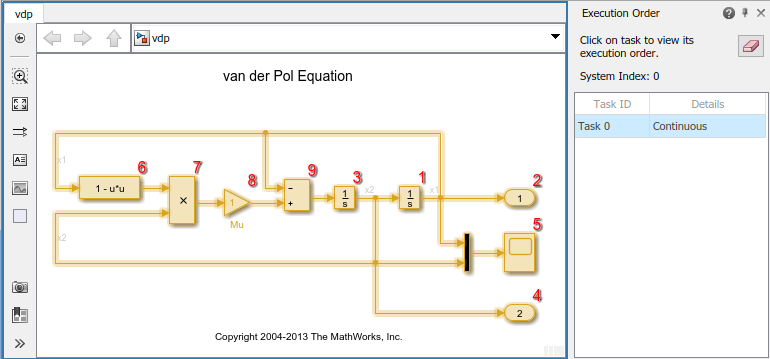
ラベル内のタスク番号は、実行順序ビューアで対応するタスクを選択するためにクリックできるリンクです。モデルに多くのタスクがある場合、これらのリンクは、実行順序ビューアでタスクのリストをスクロールする代わりに使用できます。

ブロック間でタスクを比較するには、複数のブロックを選択します。

**実行順序表記**

Simulinkは、各非バーチャルブロックの右上隅に番号を表示します。これらの番号は、ブロックが実行される順序を示します。実行する最初のブロックは、実行順序が最も低く、通常は1です。表示される実行順序は番号をスキップする場合がありますが、ブロックは常に表示される番号の順序で実行されます。タスクが実行順序1、2、4を表示するとします。1のラベルが付いたブロックは、2のラベルが付いたブロックの前に実行され、4のラベルが付いたブロックの前に実行されます。

たとえば、vdpモデルでは、ブロックの実行順序の範囲は 1〜9であり、各非仮想ブロックは実行順序を受け取ります。

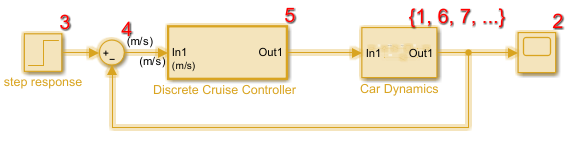


**仮想および非仮想サブシステム**

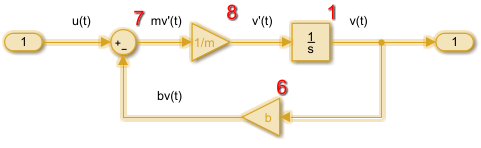
仮想サブシステムブロックはグラフィカルにのみ存在し、実行されません。したがって、それらは実行順序の一部ではありません。仮想サブシステム内のブロックには、ルートレベルモデルのコンテキストで実行順序があります。仮想サブシステムの場合、サブシステム内のブロック実行順序は、中括弧{}内にリストされています。

非バーチャルサブシステムブロックは機能的に存在し、1つのユニットとして実行されます。それらは単一の実行順序と親モデルとは異なるシステムインデックスを持っています。非バーチャルサブシステム内のブロックには、親モデルから独立した独自の実行順序があります。

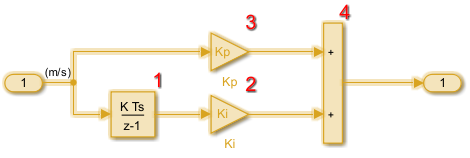
たとえば、次のモデルには、Car Dynamicsという名前の仮想サブシステムと、Discrete Cruise Controllerという名前のアトミックな非仮想サブシステムが含まれています。



仮想カーダイナミクスサブシステムは、含まれるブロックの波括弧内の実行順序のリストを表示します。含まれているブロックは、ルートレベルで実行されます。 インテグレータブロックは、最初に実行し、その出力を送信 スコープ第実行ルートレベルモデルでブロック。



非仮想Discrete Cruise Controllerサブシステムには単一の実行順序（5）があります。これは、サブシステムとその中のブロックが、ルートレベルのブロックに対して5番目に実行されることを示しています。

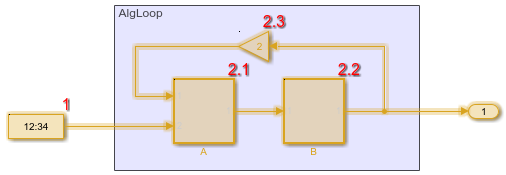


**注意**

モデル構成に応じて、Simulinkは非表示の非仮想サブシステムをモデルに挿入できます。その結果、非表示のSubsystemブロック内の可視ブロックは、現在のシステムインデックスとは異なるシステムインデックスを持つことができます。たとえば、**[条件付き入力ブランチ実行](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/gui/conditional-input-branch-execution.html&usg=ALkJrhg4z59a5uBF7Fwxm0lw_V5KCLETgg)**構成パラメーターを選択すると、Simulinkは非表示の非仮想サブシステムを作成します。これは、並べ替えられた実行順序に影響を与える可能性があります。

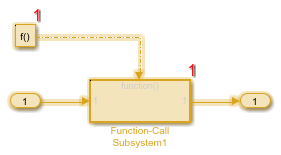
**代数ループ**

代数ループ内のブロックは、非仮想の非表示サブシステムに移動されます。隠しサブシステムの実行順序は、他のブロックのコンテキスト内で決定されます。次に、隠しサブシステム内のブロックの実行順序が決定されます。

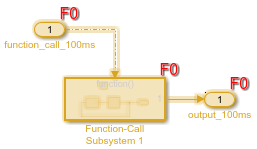


**関数呼び出しとアクションサブシステム**

Function-callおよびactionサブシステムの場合、サブシステムの実行はイニシエーターの実行に関連付けられています。したがって、サブシステムとそのイニシエーターは実行順序を共有します。



エクスポート関数モデルのルートレベルでは、関数呼び出しの実行順序にFプレフィックスがあり ます。

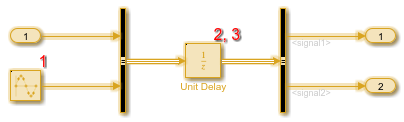


詳細については、「[エクスポート関数モデルの概要」を](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/ug/export-function-models.html&usg=ALkJrhjqKhsMrpDLJyuy4Htd4hnyeht_Gg)参照してください。

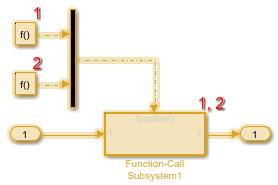
**バスと複数のイニシエーター**

ブロックは、そのブロックへの異なる実行パスに基づいて複数回実行される場合、複数の実行順序を持ちます。例えば：

* バスに接続されたブロックには、バスに含まれる各信号に対応する実行順序があります。



* 複数のイニシエーターを持つFunction-CallまたはActionサブシステムには、各イニシエーターに対応する実行順序があります。



**Simulinkが実行順序を決定する方法**

Simulinkは、タスクベースの並べ替えを使用して、派生したサンプル時間情報に基づいてブロックとポートの実行順序を設定します。タスクベースのソートは、ブロックの実行順序を決定するための効率的で単純なプロセスを提供します。

タスクベースの並べ替え：

* タスクは、サンプル時間に基づいて個別に並べ替えられます。
* すべてのタスクにわたるブロックの1つのフラット化されたソート済みリストの代わりに、複数のソート済みリストが生成されます。
* レート遷移処理が簡素化されています。
* さまざまなタスクのブロックに関連する誤ったデータ依存違反は回避されます。
* コード生成の結果は、効率的なレートグループに分類されます。
* 1つのサブシステムは、複数のタスクで異なるソートリストに属することができます。

**直接フィードスルーポートの実行順序への影響**

実行順序がブロック間のデータ依存関係を確実に反映するように、Simulinkはブロック入力ポートのブロック出力の依存関係に従ってブロック入力ポートを分類します。現在の値がブロック出力の1つの現在の値を決定する入力端子は、*直接フィードスルー*端子です。直接フィードスルー端子を持つブロックの例には、次のものがあります。

* [利得](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/slref/gain.html&usg=ALkJrhhWVK3-XckMtFd1R8tOSTeHpEweSw)
* [製品](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/slref/product.html&usg=ALkJrhhW_11hMd2t4lmw9OzGNGP-863UJw)
* [和](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/slref/add.html&usg=ALkJrhjM9JOL_8nT4ECXD9gBgzCsm3KMrA)

非直接フィードスルー入力を持つブロックの例には、次のものがあります。

* [インテグレーター](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/slref/integrator.html&usg=ALkJrhj4yrVzoQpn1JDtHt_Yk4PFnpUtpw) —出力は、その状態の関数です。
* [定数](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/slref/constant.html&usg=ALkJrhgKtIUghOKsBtA1ivNa4DuyQVWo0w) —入力はありません。
* [メモリ](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/slref/memory.html&usg=ALkJrhhJlgBX_f-4z-A7XsC9iKzmjwzIEQ) —出力は、前のタイムステップからの入力に依存します。

**ブロックの実行順序を決定するためのルール**

ブロックを並べ替えるために、Simulinkは次のルールを使用します。

* ブロックが別のブロックの直接フィードスルーポートを駆動する場合、そのブロックは、駆動するブロックの前に実行順序で表示される必要があります。

このルールは、Simulinkが現在の入力を必要とするブロックメソッドを呼び出すときに、ブロックへの直接フ​​ィードスルー入力が有効であることを保証します。

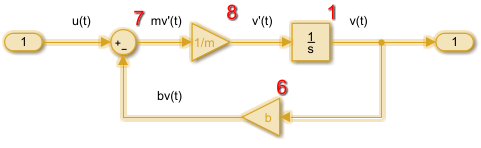
* 直接フィードスルー入力を持たないブロックは、駆動する直接フィードスルーブロックの前にある限り、実行順序のどこにでも出現できます。

直接フィードスルーポートを持たないすべてのブロックを実行順序の最初に配置すると、このルールが満たされます。この配置により、Simulinkは並べ替え処理中にこれらのブロックを無視できます。

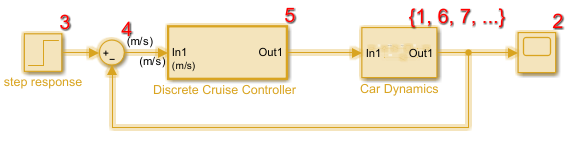
これらのルールを適用すると、実行順序が決まります。直接フィードスルー端子のないブロックは、リストの最初に特定の順序で表示されません。これらのブロックの後には、駆動するブロックに有効な入力を供給できるように配置された直接フィードスルーポートを備えたブロックが続きます。

次のモデルは、この結果を示しています。次のブロックには直接フィードスルーがないため、ルートレベルシステムの実行順序の最初に表示されます。

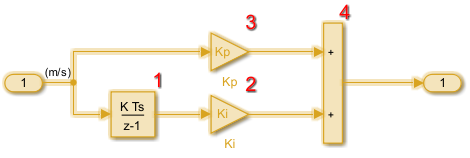
* Car Dynamics仮想サブシステムのIntegratorブロック



* ルートレベルモデルの速度ブロック



Discrete Cruise Controller Subsystem内では、直接フィードスルーポートを持つすべての[Gain](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/slref/gain.html&usg=ALkJrhhWVK3-XckMtFd1R8tOSTeHpEweSw)ブロックが、それらが駆動する[Sum](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/slref/add.html&usg=ALkJrhjM9JOL_8nT4ECXD9gBgzCsm3KMrA)ブロックの前に実行されます。



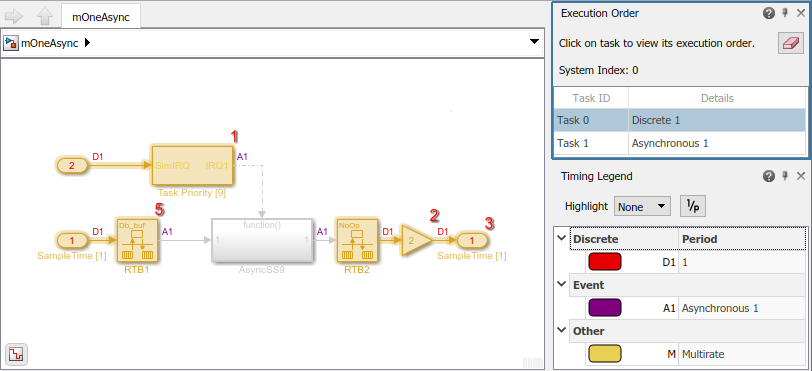
**非同期タスクを含むモデルのブロック実行順序を決定するためのルール**

シミュレーションでは、タスク内のブロックの並べ替え順序を決定するときに、非同期関数呼び出しイニシエーターが最も優先されます。詳細は、[非同期サンプル時間](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/simulink/ug/types-of-sample-time.html&usg=ALkJrhg6YKdKry0ZV_TcMUdxmdr6PFG4Fg#brrdmmw-13)と [レートの遷移と非同期ブロック](https://translate.googleusercontent.com/translate_c?depth=1&hl=ja&pto=aue&rurl=translate.google.co.jp&sl=en&sp=nmt4&tl=ja&u=https://de.mathworks.com/help/rtw/ug/rate-transitions-and-asynchronous-blocks.html&usg=ALkJrhg8t4NLgmbvw2qEBWE5FwTK3FCZlw)（Simulink Coder）を参照してください。

非同期サンプル時間を含むモデルのブロック実行順序を決定するために、Simulinkは次のルールを使用します。

* 非同期関数呼び出しイニシエーターが、対応するレート遷移ブロックによって共有される離散レートによってトリガーされる場合、共通の離散レートのタスクでは、非同期関数呼び出しイニシエーターが最初にソートされます。

たとえば、次のモデルでは、離散レートによってトリガーされる非同期関数呼び出しのイニシエーターが1つありますD1。個別のタスク内では、非同期関数呼び出しのイニシエーターが最初にソートされます。率遷移 ブロックは、非同期率、間の変換A1、および離散率 D1。

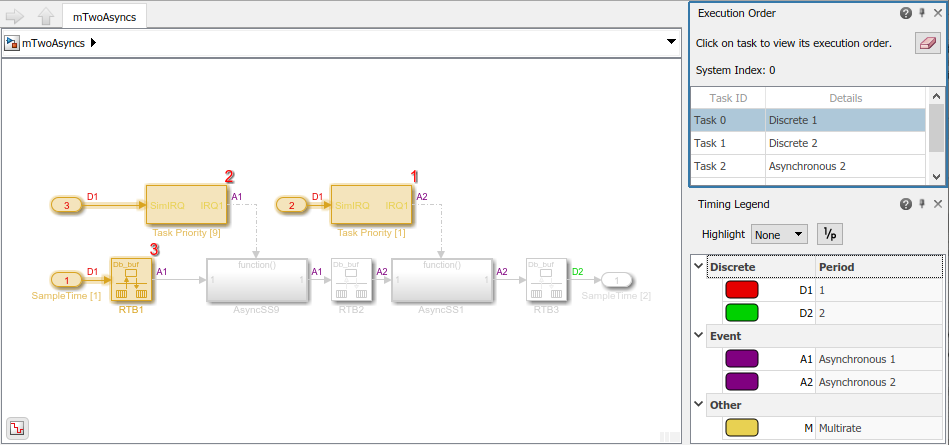


* 場合率遷移ブロックが還元されNoOp、 送信率遷移ブロックが実行されない、離散タスク内でソートされた順序でのその位置は、それが接続されている上流または下流の別個のブロックに転送されます。

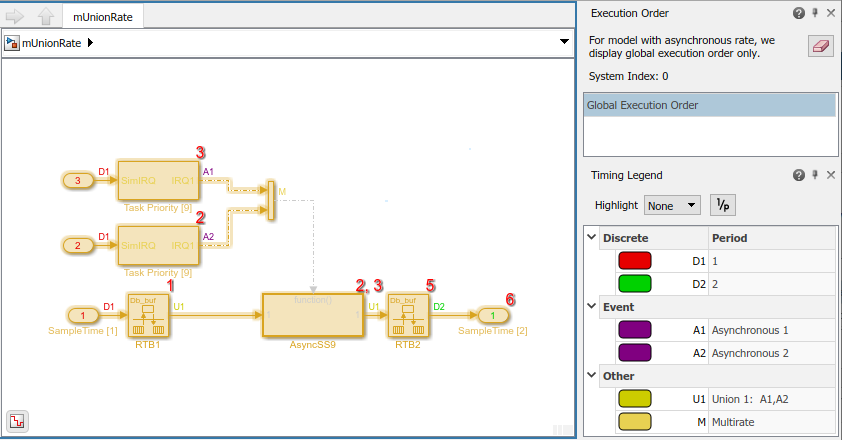
たとえば、上記のモデルでRTB2は、がに減少している NoOpため、ソートされた順序での位置が下流のGainブロックに転送されます。

* 2つの非同期関数呼び出しイニシエーターが共通の離散レートタスクを共有する場合、優先度の高い非同期関数呼び出しイニシエーターは、離散レートタスク内で最初に実行されます。

たとえば、以下のモデルでは、2つの非同期関数呼び出しイニシエーターが同じ離散レートによってトリガーされD1ます。タスクの優先度が高い方が最初にソートされます。



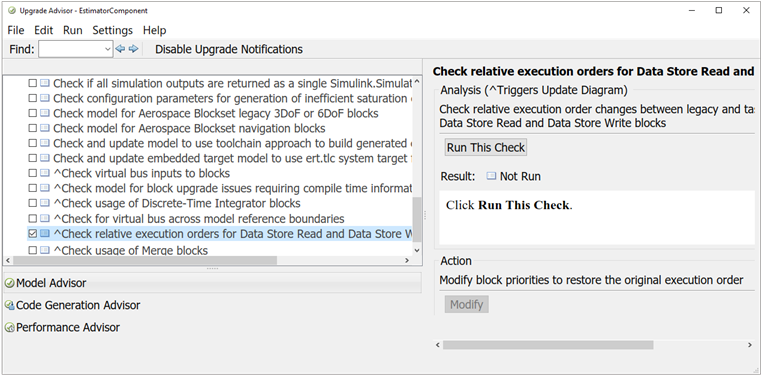
* 非同期サンプル時間の結合は、タスクベースの並べ替えではサポートされていません。複数の非同期サンプル時間の和集合を含むモデルは、デフォルトでグローバル実行順序になり、すべてのブロックが単一のタスク内でソートされます。



**データストアメモリブロックを含む実行順序の変更を確認する**

ブロックの並べ替えの代わりにタスクベースの並べ替えを使用すると、Data Store Memoryブロックに関連する相対的な実行順序を変更できます。アップグレードアドバイザーチェックは変更を検出し、モデルのアップグレード時に元の実行順序を維持するオプションを提供します。

1. アップグレードアドバイザーを開きます。上の**モデル**タブ、選択**モデルアドバイザーを** > **Upgrade Advisorを**。
2. のチェックボックスを選択しますCheck relative execution orders for Data Store Read and Data Store Write blocks。



1. [ **このチェック**を**実行** ]ボタンをクリックします。
2. **結果**テーブルの変更を確認します。元の実行順序を保持する場合は、[ **変更** ]ボタンをクリックします。