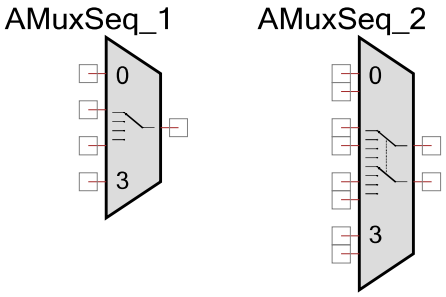
Analog Multiplexer Sequencer (AMuxSeq)

1.50

特長



- シングル接続および差動接続が可能
- 2～32 の接続に対応可能
- ソフトウェアで制御
- 接続はピンソースまたは内部ソース
- 同時接続なし
- 双方向(受動型)

概要説明

Analog multiplexer sequencer (AMuxSeq)のコンポーネントは、「フックアップ・オーダ」のシーケンスで接続、切断を行って、単一のアナログ信号を一度に複数の共通アナログ信号に接続するために使用します。AMuxSeq は、主に時分割多重化で使用します。

AMuxSeq の用途

AMuxSeq コンポーネントは、複数のアナログ信号を同時に単一のソースや送信先に多重化したい場合に使用できます。AMuxSeq は受動型であるため、入力信号でも出力信号でも多重化できます。

AMuxSeq の API は、AMux の API より簡単で高速です。複数の同時接続を必要とせず、信号が必ず同じ順番でアクセスされる場合は、AMux の代わりに AMuxSeq を使用してください。

入出力接続

このセクションでは、AMuxSeq の様々な入出力の接続について説明します。I/O リストにアスタリスク(*)が付いている場合、その I/O の説明で一覧されている条件で、I/O がシンボルに隠れている可能性があることを示します。

0-31 – アナログ

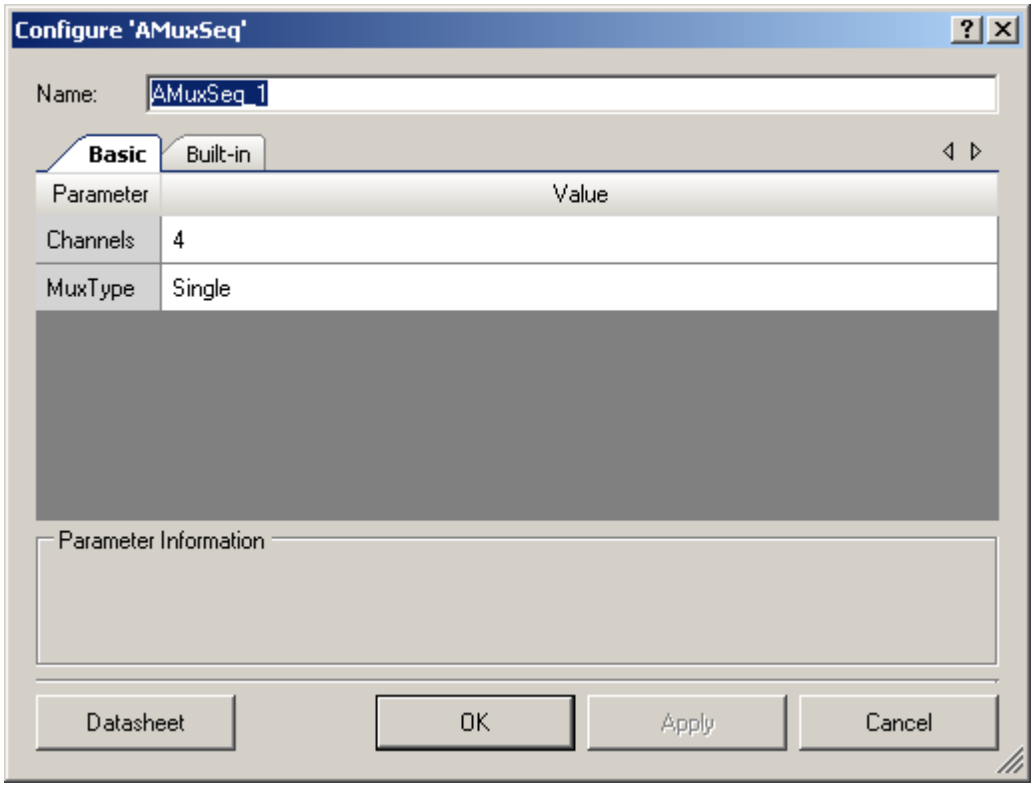
AMuxSeq は、2～32 のアナログ・スイッチ可能な接続に対応できます。**MuxType** パラメータが **Differential** に設定されている場合、ペアになった接続が存在します。

共通 – アナログ

「共通」信号は、共通な接続です。ラベル付けはされていません。AMuxSeq_Next()関数で選択されたスイッチ可能な接続信号は、この端子に接続されます。**MuxType** パラメータが **Differential** に設定されている場合、ペアになった信号が存在します。

コンポーネント・パラメータ

AMuxSeq コンポーネントをデザイン上にドラッグし、ダブルクリックして **Configure** ダイアログを開きます。



AMuxSeq は、次のパラメータを提供します。

チャネル

このパラメータは、**MuxType** に基づいて複数の入力またはペアになった入力を選択します。有効な値は 2～32 です。

MuxType

このパラメータは、接続毎に単一の入力(**Single**)か、デュアル入力 **Differential** 入力マルチプレクサを選択します。**Single** は、入力信号がすべて V_{SSA} などの同じ信号を参照している場合に使用します。2 つ以上信号が

別の信号を参照している場合は、**Differential** オプションを選択します。差動モードは、差動入力を提供する ADC で最も頻繁に使用されます。

リソース

AMuxSeq は個々のスイッチを使用して、ブロックやピンをアナログ・バスに接続します。

アプリケーション・プログラミング・インタフェース

アプリケーション・プログラミング・インターフェース(API)ルーチンにより、ソフトウェアを使用してコンポーネントを設定できます。次の表は、各関数へのインターフェースとその説明を示しています。続くセクションでは、各関数について詳しく説明します。

初期設定では、PSoC Creator はインスタンス名「AMuxSeq_1」を所定の設計上のコンポーネントの最初のインスタンスに割り当てます。インスタンス名は、識別子の構文ルールに従った固有の値に変更できます。インスタンス名は、すべてのグローバル関数名、変数名、定数名のプリフィックスになります。読みやすいように、下表では「AMuxSeq」というインスタンス名を使用しています。

関数	説明
AMuxSeq_Init()	すべてのチャネルの接続を解除
AMuxSeq_Start()	すべてのチャネルの接続を解除
AMuxSeq_Stop()	すべてのチャネルの接続を解除
AMuxSeq_Next()	前のチャネルを切断し、シーケンスで次のチャネルを接続します。
AMux_DisconnectAll()	すべてのチャネルの接続を解除
AMuxSeq_GetChannel()	現在接続しているチャネルを返します。チャネルが接続されていなければ、「–1」を返します。

void AMuxSeq_Init(void)

説明: すべてのチャネルを切断します。次回、AMuxSeq_Next()を呼び出すと、最初のチャネルが選択されます。

パラメータ: なし

返り値: なし

副作用: すべてのレジスタが初期値にリセットされます。

void AMuxSeq_Start(void)

説明: すべてのチャネルを切断します。次回、AMuxSeq_Next()を呼び出すと、最初のチャネルが選択されます。

パラメータ: なし

返り値: なし

副作用: なし

void AMuxSeq_Stop(void)

説明: すべてのチャネルを切断します。次回、AMuxSeq_Next()を呼び出すと、最初のチャネルが選択されます。

パラメータ: なし

返り値: なし

副作用: なし

void AMuxSeq_Next(void)

説明: 前のチャネルを切断し、シーケンスで次のチャネルを接続します。AMuxSeq_Next()を初めて呼び出すか、AMuxSeq_Init()、AMuxSeq_Start()、AMuxSeq_Enable()、AMuxSeq_Stop()、またはAMuxSeq_DisconnectAll()の後に呼び出すと、チャネル0に接続します。

パラメータ: なし

返り値: なし

副作用: なし

void AMux_DisconnectAll(void)

説明: この関数は、すべてのチャネルを切断します。次回、AMuxSeq_Next()を呼び出すと、最初のチャネルが選択されます。

パラメータ: なし

返り値: なし

副作用: なし

int8 AMuxSeq_GetChannel(void)

説明: 現在接続しているチャネルを返します。チャネルが接続されていなければ、「–1」を返します。

パラメータ: なし

返り値: 現在のチャネル、または「–1」。

副作用: なし

ファームウェア・ソースコードのサンプル

PSoC Creator は、[Find Example Project] ダイアログに多数のサンプルプロジェクトを提供しており、そこには回路図およびサンプル コードが含まれています。コンポーネント固有の例を見るには、[Component Catalog] または回路図に置いたコンポーネント インスタンスからダイアログを開きます。一般例については、[Start Page] または **[File]** メニューからダイアログを開きます。必要に応じてダイアログにある **Filter Options** を使用し、選択できるプロジェクトのリストを絞り込みます。

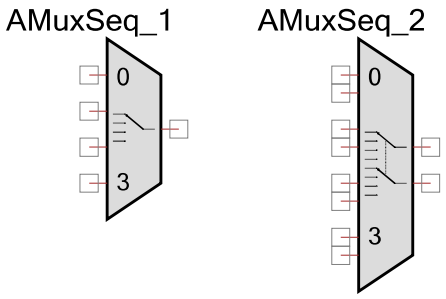
詳しくは、PSoC Creator ヘルプの「Find Example Project（サンプルプロジェクトを検索）」トピックを参照してください。

機能の説明

AMuxSeq は、ハードウェアではなく、ファームウェアによって制御されます。共通信号に接続できるのは、同時に 1 つの信号だけです。

以下は、単一および差動として構成された AMuxSeq のフローを示しています。

シングル **差動**



性能

Sequential Analog Mux は、ソフトウェアによって制御されるので、スイッチングの性能は提供される API の実行時間によります。性能は、設計におけるマルチプレクサの正確な構成に応じて変動しますが、単一入力は切断されて、他の呼び出しで接続されるので、入力の数には影響されません。表 1 は、スイッチングの性能に関するガイダンスです。

すべての性能測定は、CPU 周波数 48 MHz で行っています。性能は、CPU 周波数にほぼ比例します。PSoC Creator にバンドルされているコンパイラは最高に最適化されています。PSoC 3 では、Keil コンパイラの設定はスピード最適化レベル 5 またはサイズに対しての最適化が行われています。PSoC 5 では、GNU コンパイラの設定はサイズまたはスピードに対して最適化が行われています。

表 1. 性能

関数	最適化	PSoC 3 (µs)	PSoC 5 (µs)
Next	Size	8.8	1.4
	Speed	2.4	1.4

DC 電気的特性と AC 電気的特性

AMuxSeq は有効な電源電圧の全範囲で動作します。

コンポーネントの変更

ここでは、過去のバージョンからコンポーネントに加えられた主な変更を示します。

バージョン	変更の説明	変更の理由 / 影響
1.50.c	データシートに性能セクションを追加	
1.50.b	データシートのマイナーな編集と更新	
1.50.a	データシートのマイナーな編集と更新	
1.50	AMuxSeq_Init()関数を追加	企業の標準に準拠し、コンポーネントを起動せずに初期化、復元できるAPIを提供するため。
1.20.a	シリコン リビジョンとの互換性について知らせる情報をコンポーネントに追加しました。	コンポーネントが互換性のないシリコンで使用されると、ツールはエラーか警告をレポートします。エラーが表示されたら、対象デバイスをサポートするリビジョンにアップデートしてください。
1.20	シンボルの画像を更新しました。	企業標準に準拠して、シーケンスを示すように更新しました。
	AMuxSeq_GetChanne()l API関数を追加しました。	現在接続されているチャネルを取得するため。
	引数がない関数に欠けていた'void'を追加しました。 -1はチャネルが選択されていないことを示すために使用されるので、AMuxチャネル変数のタイプを符号なしから符号付き整数へ変更しました。	こうした変更で、MDKおよびRVDSコンパイラによるコンパイル中に表示される、deprecated宣言に関する警告を解決しました。

Copyright © 2005-2012 Cypress Semiconductor Corporation 本文書に記載される情報は、予告なく変更される場合があります。Cypress Semiconductor Corporationは、サイプレス製品に組み込まれた回路以外のいかなる回路を使用することに対しても一切の責任を負いません。特許又はその他の権限下で、ライセンスを譲渡又は暗示することもありません。サイプレス製品は、サイプレスとの書面による合意に基づくものでない限り、医療、生命維持、救命、重要な管理、又は安全の用途のために仕様することを保証するものではなく、また使用することを意図したものでもありません。さらにサイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことを合理的に予想される、生命維持システムの重要なコンポーネンツとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

PSoC Designer™及びProgrammable System-on-Chip™は、Cypress Semiconductor Corp.の商標、PSoC®は同社の登録商標です。本文書で言及するその他全ての商標又は登録商標は各社の所有物です。

全てのソースコード(ソフトウェア及び/又はファームウェア)はCypress Semiconductor Corporation (以下「サイプレス」)が所有し、全世界(米国及びその他の国)の特許権保護、米国の著作権法並びに国際協定の条項により保護され、かつそれらに従います。サイプレスが本書面によるライセンシーに付与するライセンスは、個人的、非独占的かつ譲渡不能のライセンスであって、適用される契約で指定されたサイプレスの集積回路と併用されるライセンシーの製品のみをサポートするカスタムソフトウェア及び/又はカスタムファームウェアを作成する目的に限って、サイプレスのソースコードの派生著作物を複製、使用、変更、そして作成するためのライセンス、並びにサイプレスのソースコード及び派生著作物をコンパイルするためのライセンスです。上記で指定された場合を除き、サイプレスの書面による明示的な許可なくして本ソースコードを複製、変更、変換、コンパイル、又は表示することは全て禁止されます。

免責条項:サイプレスは、明示的又は黙示的を問わず、本資料に関するいかなる種類の保証も行いません。これには、商品性又は特定目的への適合性の黙示的な保証が含まれますが、これに限定されません。サイプレスは、本文書に記載される資料に対して今後予告なく変更を加える権利を留保します。サイプレスは、本文書に記載されるいかなる製品又は回路を適用又は使用したことによって生ずるいかなる責任も負いません。サイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことが合理的に予想される生命維持システムの重要なコンポーネンツとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

ソフトウェアの使用は、適用されるサイプレスソフトウェアライセンス契約によって制限され、かつ制約される場合があります。