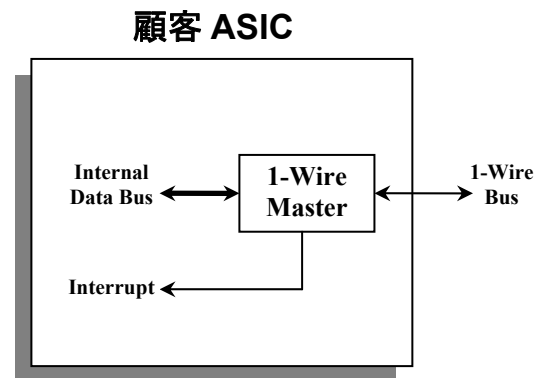


特長

- 任意の標準的バイト幅データバスにメモリマッピングが可能
- 1-Wire[®]のタイミングおよび制御信号をすべて内部で生成することによって、CPUの「ビットバンギング」が不要
- より効率的なプログラミングを可能にする割込みの生成
- 1-Wire バス上のすべての単一ビット操作からCPUを解放するROM 検索アクセラレータ
- オーバドライブモードおよびスレーブ割込みをサポート
- 3.2MHz~128MHz の任意のシステムクロックで動作可能
- 小型: 完全デジタル設計、わずか 1500 ゲート
- Verilog および VHDL の両方で利用可能
- アプリケーション: 1-Wire 通信バスを含むすべての回路



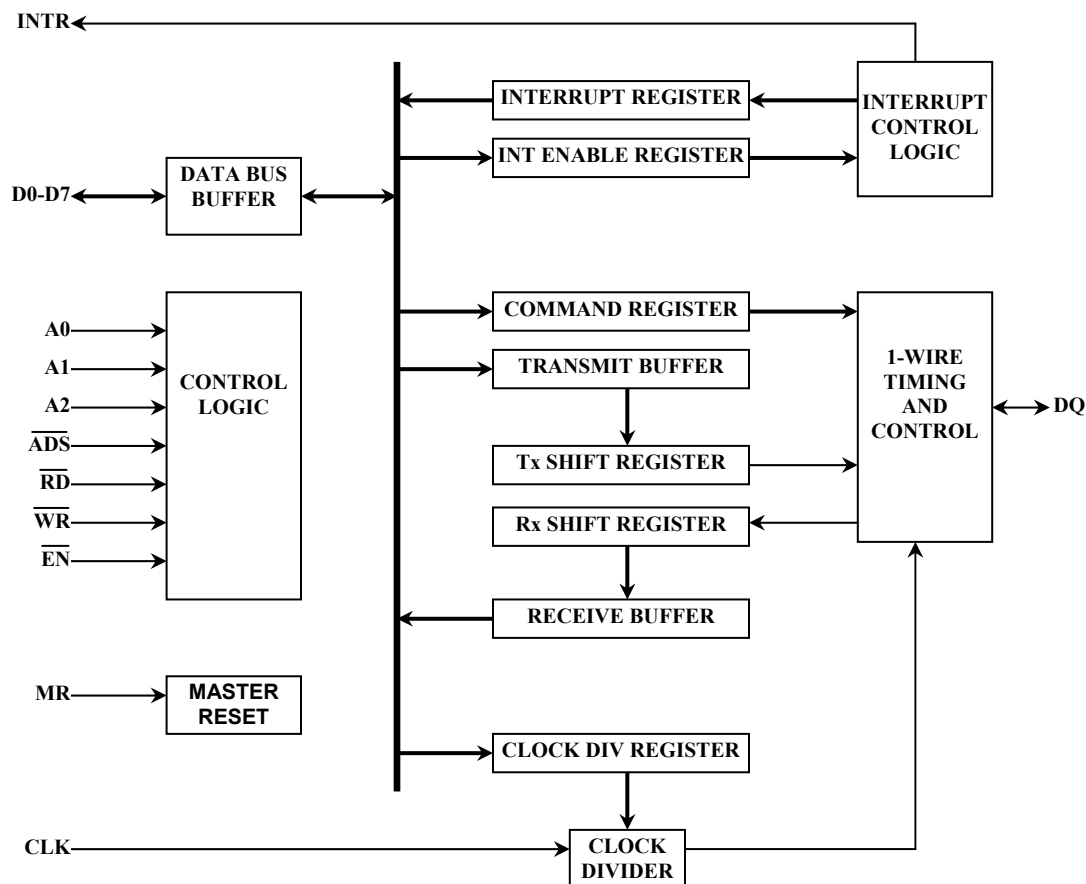
説明

利用可能な 1-Wire デバイスが増大するにつれて、それらのデバイスと通信を行うために必要な 1-Wire 信号の生成に対応することを求められるユーザの数も増え続けています。そのためには通常、マイクロプロセッサのポートピンを「ビットバンギング」して、1-Wire プロトコルで要求されるタイミング機能をマイクロプロセッサに実行させることが必要になります。1-Wire の伝送はバイト途中での割込みが可能ですが、ビットタイムスロットの「ロー」期間中に割込みを行うことはできません。したがって、1 ビット送信ごとに最大 60 マイクロ秒、そして 1-Wire のリセット生成時には少なくとも 480 マイクロ秒間は CPU がアイドル状態に置かれることとなります。1-Wire マスタは、こうした貴重な CPU サイクルの占有を伴わずに、ユーザシステムにおける 1-Wire デバイスとの通信を実現する上で役立ちます。ユーザの ASIC 中に 1-Wire ポートとして集積化されるこのコアは、VHDL と Verilog の両方のコードで利用可能であり、非常に小さなチップ面積しか使用しません (Verilog 版の場合で 1492 ゲート + 1 つのボンダパッド)。

この回路は、ユーザのシステムにメモリマップされるように設計されており、8 ビットのコマンドを通して 1-Wire バスの完全な制御を提供します。ホスト CPU は、5 つの個別レジスタを通して、コマンドのロード、データの読み書き、および割込み制御の設定を行います。1-Wire バスのタイミングと制御は、すべて内部で生成されます。ホストは単にコマンドまたはデータをロードするだけでよく、あとは他の作業に戻ることができます。CPU が受信する必要がある応答がバスの動作によって生成されると、1-Wire マスタはステータス・ビットをセットし、イネーブルされていれば CPU への割込みを生成します。読み書きの単純化に加えて、1-Wire マスタは ROM 検索アクセラレータ機能も備えており、CPU が 1-Wire バス上で単一ビット操作を行う必要がなくなっています。

1-Wireバスの動作については、*Book of iButton Standards* [1]に詳しく記述されています。したがって、この文書ではその詳細については解説しません。一般に、各スレーブデバイスはそれぞれ固有のコマンド群を備えており、そのデバイスのデータシートで詳述されています。特定のスレーブの実装に関する詳細は、それらの文書を参照してください。

ブロック図



端子説明

以下では、ブロックのすべての I/O 端子の機能を説明します。以下の説明の中で、0 は論理ロー、1 は論理ハイを示します。

A0、A1、A2、レジスタ選択:この 3 つの入力に接続されたアドレス信号によって、データ転送中に CPU が読取りまたは書込みを行うレジスタを選択します。レジスタとそのアドレスの表を次に示します。

レジスタアドレス

A2	A1	A0	レジスタ
0	0	0	コマンドレジスタ(読取り/書込み)
0	0	1	送信バッファ(書込み)、受信バッファ(読取り)
0	1	0	割込みレジスタ(読取り)
0	1	1	割込みイネーブルレジスタ(読取り/書込み)
1	0	0	クロック分周レジスタ(読取り/書込み)

\overline{ADS} 、アドレスストローブ:アクティブなアドレスストローブ(\overline{ADS})信号の正のエッジで、レジスタ選択(A0、A1、A2)が内部ラッチにセットされます。セットアップおよびホールドのタイミングが遵守されている限りにおいて、 \overline{ADS} をローに固定してラッチを透過にすることができます。

CLK、クロック入力:3.2MHz~128MHz の範囲で(可能なかぎり) 50%デューティサイクルのクロックを印加します。このクロックが 1-Wire バスのタイミングを提供します。

D7~D0、データバス:8 本の入出力ラインで構成されるバスです。このバスが 1-Wire マスタと CPU の間の双方向通信を提供します。データ、コントロールワード、およびステータス情報が、この D7~D0 のデータバスを通して転送されます。

DQ、1-Wire データライン:このオープンドレインのラインが、1-Wire の双方向データバスです。1-Wire のスレーブデバイスをこの端子に接続します。この端子は公称 5K Ω の外付け抵抗でハイにプルアップする必要があります。

\overline{EN} 、イネーブル: \overline{EN} がローのとき、1-Wire マスタがイネーブルされます。この信号は、デバイスのチップイネーブルの機能を果たします。これによって、1-Wire マスタと CPU の間の通信が可能になります。

INTR、割込み:割込みタイプのいずれか 1 つがアクティブハイ条件で、かつ割込みイネーブルレジスタを通してイネーブルされているとき、このラインがアクティブ状態になります。割込みレジスタの読取りが行われると、INTR 信号はインアクティブ状態にリセットされます。

MR、マスタリセット:この入力が高レベルのとき、1-Wire マスタのすべてのレジスタと制御ロジックがクリアされ、INTR はデフォルトのインアクティブ状態であるハイに設定されます。

\overline{RD} 、読取り:読取りサイクル中は、この端子がバスを駆動します。回路がイネーブルされているとき、 \overline{RD} をローに駆動することによって、CPU は選択されたレジスタからステータス情報またはデータを読み取ることができます。 \overline{RD} と \overline{WR} は、同時にローにしないでください。同時にローにした場合は、 \overline{WR} が優先されます。

\overline{WR} 、書込み:書込みサイクル中は、この端子がバスを駆動します。回路がイネーブルされているときに \overline{WR} がローになると、CPU は選択されたレジスタにコントロールワードまたはデータを書き込むことができます。 \overline{RD} と \overline{WR} は、同時にローにしないでください。同時にローにした場合は、 \overline{WR} が優先されます。

動作 — クロック分周

1-Wire のすべてのタイミングパターンは、1.0MHz のベースクロックを使って生成されます。1-Wire マスタは、CLK 端子に印加される外部リファレンスを元に、内部でこのクロック周波数を生成します。外部クロックは周波数が 3.2MHz~128MHz でなければならず、可能なら 50%のデューティサイクルが望まれます。クロック分周レジスタが内蔵のクロック分周器を制御して、必要なリファレンス周波数を提供します。これは 2 つの段階で行われます。最初にプリスケアラが 1、3、5、または 7 分周を行い、次に残りの回路が 2、4、8、16、32、64、または 128 分周を行います。

クロック分周レジスタ

Addr. 04h	X	X	X	DIV2	DIV1	DIV0	PRE1	PRE0
	MSB							LSB

1-Wire バス上での通信を行う前に、クロック分周の設定を行う必要があります。マスタリセットが発生すると、このレジスタには 0x00h がセットされます。下の表を使用して、CLK のリファレンス周波数を元に正しいレジスタ値を見つけてください。たとえば、15MHz の入力クロックを供給する場合は、この位置に 0x10h を書き込むことになります。

入力クロック速度に対するクロック分周レジスタの設定

最小 CLK 周波数 (MHz)	最大 CLK 周波数 (MHz)	分周比	DIV3	DIV2	DIV1	PRE1	PRE0
> 3.2	4.0	4	0	1	0	0	0
> 4.0	5.0	5	0	0	0	1	0
> 5.0	6.0	6	0	0	1	0	1
> 6.0	7.0	7	0	0	0	1	1
> 7.0	8.0	8	0	1	1	0	0
> 8.0	10.0	10	0	0	1	1	0
> 10.0	12.0	12	0	1	0	0	1
> 12.0	14.0	14	0	0	1	1	1
> 14.0	16.0	16	1	0	0	0	0
> 16.0	20.0	20	0	1	0	1	0
> 20.0	24.0	24	0	1	1	0	1
> 24.0	28.0	28	0	1	0	1	1
> 28.0	32.0	32	1	0	1	0	0
> 32.0	40.0	40	0	1	1	1	0
> 40.0	48.0	48	1	0	0	0	1
> 48.0	56.0	56	0	1	1	1	1
> 56.0	64.0	64	1	1	0	0	0
> 64.0	80.0	80	1	0	0	1	0
> 80.0	96.0	96	1	0	1	0	1
> 96.0	112.0	112	1	0	0	1	1
> 112.0	128.0	128	1	1	1	0	0

動作 — データの送受信

1-Wire マスタが送受信するデータは、送受信バッファ位置を通過します。実際の 1-Wire マスタでは、独立した送信バッファと受信バッファによるダブルバッファを使用しています。この位置に書き込みを行うと送信バッファがデータバスに接続され、読取りを行うと受信バッファがデータバスに接続されます。

送信バッファ(書き込み) / 受信バッファ(読取り)

Addr. 01h	Data7	Data6	Data5	Data4	Data3	Data2	Data1	Data0
	MSB				LSB			

バイト書き込み

1-Wire バス上に 1 バイトを送信するには、所望のデータを送信バッファに書き込みます。するとそのデータが送信シフトレジスタに移動され、順次シフトされて LSB から先にバスに送出されます。これで新しいデータバイトを送信バッファに書き込むことができます。送信シフトレジスタが空になると、すぐにデータが送信バッファから転送され、このプロセスが繰り返されます。これらのレジスタは、それぞれ割込みソースとして使用することができるフラグを備えています。送信バッファエンプティ(TBE)フラグは、送信バッファが空で、新しいバイトの受入れが可能なときにセットされます。バイトが送信バッファに書き込まれると、直ちに TBE はクリアされます。送信シフトレジスタエンプティ(TEMT)フラグは、シフトレジスタ内にデータが存在せず、新しいバイトの受入れが可能なときにセットされます。送信バッファからデータバイトが転送されると、TEMT はクリアされ TBE がセットされます。適正な 1-Wire プロトコルでは、何らかのバス通信を行う前に必ずリセットが要求されることに注意してください。

バイト読取り

スレーブデバイスからデータを読み取るためには、そのデバイスがあらかじめ CPU から受け取ったコマンドに応じてデータを送信する準備ができていなければならない必要があります。データは、書き込み動作の場合と同じような形でバスから取り出されます。ホストは送信バッファに書き込むことによって読取りを開始します。それによって受信シフトレジスタ内にシフトされるデータは、書き込んだデータとスレーブデバイスからのデータがワイヤード AND されたものになります。したがって、スレーブデバイスからのバイトを読み取るためには、ホストは 0xFFh を書き込む必要があります。受信シフトレジスタが一杯になると、データが受信バッファに転送され、ホストからアクセス可能になります。ここで再び 0xFFh を送信することによって、さらにバイトを読み取ることができます。スレーブデバイスの送信準備ができていない場合、受信データは送信したものと同一になります。受信バッファレジスタも、割込みを生成することができます。受信バッファフラグ(RBF)は、データが受信シフトレジスタから転送されるときにセットされ、ホストがレジスタを読み取るときにクリアされます。RBF がセットされた場合、1-Wire バス上でそれ以上の送信が行われてはならず、その場合は受信バッファ内のバイトが次の受信データで上書きされるため、データが失われる恐れがあります。バイト受取り動作の詳細については、タイミング図を参照してください。バス上での 1-Wire のリセット生成については、コマンド動作の項で取り上げます。割込みフラグについては、割込み動作の項でさらに詳しく説明します。タイミング図の中で、書き込みおよび読取り動作について詳細に示してあります。

動作 – コマンド

コマンドレジスタは、DS1WM の動作モードを決定するとともに、バス上での特殊な転送処理も扱います。このレジスタで DS1WM の動作モードを選択して、オーバドライブモード、ROM 検索アクセラレータモード、または標準モードのいずれかで動作させることができます。またこのレジスタは、バスのリセット、1-Wire マスタのリセット、およびバスの直接読取り / 書込みの制御も行います。

コマンドレジスタ(読取り/書込み)

Addr. 00h	OD	X	RST	X	DQI	DQO	SRA	1WR
	MSB				LSB			

OD: オーバドライブビット。このビットで、1-Wire バス上の通信速度を選択します。このビットに 1 をセットすると、バス上のすべてのスレーブデバイスがオーバドライブモードをサポートしている場合、高速通信が可能になります。標準の 1-Wire 通信を行う場合は、このビットに 0 をセットする必要があります。

RST: ソフトウェアリセット。このビットをセットすると、あらゆる通信が即座に中止され、通信を制御する内部のステートマシンがすべてリセットされます。このコマンドによって、レジスタ内のデータはリセットされません。すべてのレジスタデータは維持されますが、割込みレジスタは 1-Wire マスタの新しい状態を反映して変化する可能性があります。通常動作時にはこのビットを 0 にする必要があります。

DQO: DQ 出力。このビットを使用することで、必要に応じて 1-Wire マスタの動作をバイパスしてバスを直接駆動することができます。このビットをハイにセットすると、クリアするかまたは 1-Wire マスタをリセットするまでバスがローに駆動されます。1-Wire バスがローに保たれている間、他の 1-Wire マスタの動作が機能することはありません。このビットをセットしている時間と、ラインをサンプリングするポイントを制御することによって(次の DQI を参照)、ホストコントローラによる任意の 1-Wire 通信の生成が可能になります。誤ってバスに書込みが行われるのを防ぐため、あらかじめ割込みイネーブルレジスタの DQOE ビットに 1 をセットしておかないと、DQO ビットが機能しないようになっています。このビットは、通電時またはマスタリセット時に 0 でクリアされます。

DQI: DQ 入力。このビットは読取り専用であり、1-Wire バスの現在の状態が反映されます。バスを直接制御する場合は、DQO ビットと組合せて使用します。このビットの状態は、1-Wire マスタの他のどの機能にも影響しません。このビットの動作は、DQOE の状態による影響を受けません。

1WR: 1-Wire リセット。このビットがセットされると、1-Wire バス上でリセットが生成されます。このビットをセットすると、SRA ビットが自動的にクリアされます。1-Wire リセットが完了するとすぐ、1WR ビットは自動的にクリアされます。リセットが完了し、プレゼンス検出の発生に十分な時間が経過すると、1-Wire マスタはプレゼンス検出(PD)割込みフラグをセットします。プレゼンス検出の結果は、割込みレジスタの PDR ビットに設定されます。プレゼンス検出パルスを受け取った場合は PDR がクリアされ、受け取らなかった場合は PDR がセットされます。

SRA: ROM 検索アクセラレータ。このビットをセットすると、1-Wire マスタは ROM 検索アクセラレータモードに切り替わります。このモードは、あらかじめ 1-Wire バス上でリセットおよびそれに続く ROM 検索コマンド(0xF0h)が発行されていることを前提としています。1-Wire システムで ROM 検索が実際にどのように行われるかの詳細については、[1] を参照してください。簡単に言うと、バスマスタが 2 つのビット(1 つのビットとその補数)を読み取り、次にどのデバイスがバス上に残って処理を進めるか指定するビットを書き込むというアルゴリズムになっています。

1-Wire マスタが ROM 検索アクセラレータモードになった後、CPU は 16 個のバイトを送信して、1-Wire バス上における ROM 検索パスを完了する必要があります。それらのバイトは、次のような構成になっています。

1 番目のバイト

7	6	5	4	3	2	1	0
r_3	x_3	r_2	x_2	r_1	x_1	r_0	x_0

中略

16 番目のバイト

7	6	5	4	3	2	1	0
r_{63}	x_{63}	r_{62}	x_{62}	r_{61}	x_{61}	r_{60}	x_{60}

図では、1-Wire デバイスの ROM ID 中におけるビット位置を、添え字(0~63 の値、「 n 」)で表しています。「 x 」は、フィルタの役目を持ち特定の値を必要としないビット(don't care bits)を示します。「 r 」は、ROM 検索実行中に競合が発生した場合その特定ビットに書き込む、選択されたビット値を示します。

各ビット位置 n (0~63 の値)に対して、1-Wire マスタは 1-Wire バス上に 3 つのタイムスロットを生成します。それぞれを次のように表現します。

b0 第 1 のタイムスロット(データ読取り)
 b1 第 2 のタイムスロット(データ読取り)
 b2 第 3 のタイムスロット(データ書込み)

1-Wire マスタは、次のようにしてタイムスロット b2 のタイプ(1 を書き込むか、または 0 を書き込むか)を決定します。

b2 = r_n : 競合発生時(ホストが選択した値)
 = b_0 : 競合なし(選択肢なし)
 = 1 : エラー(応答なし)

検索アクセラレータを使用した ROM 検索機能を通して CPU が完了時に読み取るデータレジスタ内の応答バイト群は、次のような 16 バイトで構成されます。

1 番目のバイト

7	6	5	4	3	2	1	0
r'_3	d_3	r'_2	d_2	r'_1	d_1	r'_0	d_0

中略

16 番目のバイト

7	6	5	4	3	2	1	0
r'_{63}	d_{63}	r'_{62}	d_{62}	r'_{61}	d_{61}	r'_{60}	d_{60}

先ほどと同様、添え字は 1-Wire デバイスの ROM ID 中におけるビット位置を表します。「 d 」は、その特定のビット位置における不一致フラグを示します。不一致フラグは、その特定ビット位置に競合が存在するか、または応答がなかった場合は 1 に、それ以外は 0 になります。「 r 」は、その特定ビット位置における、実際に選択された経路を示します。選択された経路は、ROM ID の特定のビット位置に対する b2 と同一です。

ROM検索シーケンスを実行する場合、最初はすべての r_n ビットを 0 にした状態で開始します。バスエラーが発生すると、コマンドレジスタのビット 1 に 0 を書き込むことによって検索アクセラレータを終了させるまで、以後の応答ビット r'_n はすべて 1 になります。したがって、もし r'_{63} と d_{63} が両方 1 なら検索手順の中でエラーが発生したということであり、最後のシーケンスを再度実行する必要があります。エラーが発生しなければ、 r'_n ($n=0\dots63$)は発見されアクセスされた

デバイスのROMコードになります。ROM検索プロセスが完了したら、SRAビットをクリアして 1-Wire マスタをROM検索アクセラレータモードから解放します。

次にROM検索シーケンスを行うときは、前回の r_n ($n=0\dots63$)を再利用しますが、値が 1 である最も上位の不一致フラグの添え字を「m」として、 r_m に 1 をセットし、 $i>m$ であるすべての r_i に 0 をセットします。このプロセスが、連続する 2 回のパスで最上位の不一致が同じビット位置で発生するようになるまで繰り返されます。

例 — 高速 ROM 検索

次の例は、ホストが 1-Wire マスタを使用して、1-Wire バス上にある 4 つの異なるデバイスを識別するというものです。デバイスの ROM データは、次の通りです(LSB が先)。

```
ROM1 = 00110101...
ROM2 = 10101010...
ROM3 = 11110101...
ROM4 = 00010001...
```

- 1) ホストがコマンドレジスタに 0x01h を書き込むことによって、リセットパルスを発行します。すべてのスレーブデバイスがプレゼンス検出によって同時に応答します。
- 2) ホストが送信バッファに 0xF0h を書き込むことによって、ROM 検索コマンドを発行します。
- 3) ホストがコマンドレジスタに 0x02 を書き込むことによって、1-Wire マスタをアクセラレータモードにします。
- 4) ホストは送信バッファに 0x00h を書き込み、戻りデータを受信バッファから読み取ります。このプロセスが、計 16 バイトについて繰り返されます。読み取ったデータは、 r の位置に ROM4 が含まれ、 d_0 と d_2 の位置で不一致ビットがセットされます(r の位置を下線で、最上位の不一致を太字で示してあります)。

```
RECEIVED DATA 1 = 10001001100000001...
```

- 5) 次にホストはデータのインタリーブを解除し、位置 d_2 に最後の不一致を含む、00010001...という ROM コードに到達します。
- 6) ホストはコマンドレジスタに 0x00h を書き込んで、検索アクセラレータモードを終了します。これでホストは自由にコマンドの送信を行い、このデバイスから直接データを読み取ることができます。
- 7) 次のデバイスを見つけるため、ここでステップ 1~6 を繰り返します。今回送信する 16 バイトのデータは、最後の不一致ビット(この例では d_2)までは ROM4 と同一です。下に示すように、最後の不一致ビットは反転し、その後のデータにはすべて 0 をセットします。受信データは、 r の位置に ROM1 が格納され、ビット d_0 および d_2 が再びセットされます。

```
TRANSMITTED DATA 2 = 0000010000000000...
RECEIVED DATA 2 = 1000110100010001...
```

- 8) 最上位の不一致(d_2)が変化しなかったため、次に大きな不一致(d_0)を使用してプロセスを繰り返します。さらに繰り返すことで、次のようなデータが得られます。

```
TRANSMITTED DATA 3 = 0100000000000000...
RECEIVED DATA 3 = 1110010001000100...
TRANSMITTED DATA 4 = 0101000000000000...
RECEIVED DATA 4 = 1111010100010001...
```

- 9) この時点で、最上位の不一致(d1)が変化しなかったため、次に大きなもの(d0)を使用しなければなりません。しかし、d0 に到達したのはこれで 2 度目であり、それより小さな不一致は存在しないため、計 4 個のデバイスを発見して検索完了になります。

動作 — フラグと割込み

送信、受信、および 1-Wire リセットの各動作によるフラグが、割込みレジスタ内に設けられています。プレゼンス検出フラグ(PD)だけは割込みレジスタの読取り時にクリアされ、他のフラグは送信および受信バッファの読み書き時に自動的にクリアされます。割込みイネーブルレジスタで該当するイネーブルビットがセットされている場合、これらのフラグは INTR 端子上に割込みを発生させることができます。割込みレジスタの読取りを行うと、たとえすべてのフラグがクリアされていなくても、常に INTR 端子がインアクティブになります。

割込みレジスタ(読取り専用)

Addr. 02h	DQI	NBSY	SINT	RBF	TEMT	TBE	PDR	PD
	MSB						LSB	

PD: プレゼンス検出。1-Wire リセットの発行後、プレゼンス検出パルスの発生に十分な時間が経過した後で、このフラグがセットされます。このビットのデフォルト状態は 0 です。このビットは、割込みレジスタの読取り時にクリアされます。

PDR: プレゼンス検出結果。プレゼンス検出割込みの発生時、このビットには読み取ったプレゼンス検出の結果が反映されます。スレーブデバイスが発見された場合は 0、発見されなかった場合は 1 になります。このビットのデフォルト状態は 1 です。

TBE: 送信バッファエンプティ。このフラグは、送信バッファにデータが書き込まれた時にクリアされ、そのデータが送信シフトレジスタに転送されたときセットされます。このビットのデフォルト状態は 1 です。

TEMT: 送信シフトレジスタエンプティ。このフラグは、送信バッファから送信シフトレジスタにデータがシフトされたときクリアされ、最後のビットが 1-Wire バス上に送信された後でセットされます。このビットのデフォルト状態は 1 です。

RBF: 受信バッファフル。このフラグは、受信バッファ内に読取り待ちのバイトが存在するときセットされます。このフラグは、受信バッファレジスタからそのバイトが読み取られたときクリアされます。このビットのデフォルト状態は 0 です。

SINT: スレーブ割込み。このフラグは、1-Wire バスがスレーブデバイスによって 960 μ s より長い間ローに保持されたときセットされます。このビットは、レジスタが読み取られたときクリアされます。このビットのデフォルト状態は 0 です。

NBSY: ノットビジュー。このビットのデフォルト状態は 1 であり、現在 1-Wire マスタの内部で何の動作も行われていないことを示します。読取り、書込み、リセット、またはスレーブ割込みが発生するとクリアされます。動作が終了すると、このフラグは 1 状態に戻ります。

DQI: DQ 入力。このビットはコマンドレジスタのビット 3 のミラーです。これによって、割込みレジスタだけを読み取ることでマスタとバスの全状態を判断することができます。このビットは、対応するイネーブルビットが割込みイネーブルレジスタ内に存在しません。

割込みイネーブルレジスタを使うと、INTR 端子をアクティブにする割込みソースをシステムプログラマが指定することができます。INTR 端子のアクティブ状態を定義することができます。マスタリセットを受信すると、このレジスタのすべてのビットが 0 にクリアされ、すべての割込みソースがディセーブルされ、INTR 端子のアクティブ状態がローに設定されます。すなわち、すべての割込みがディセーブルされているため、INTR 端子がハイにプルアップされることになります。INTR 端子は、割込みレジスタの読取りによってインアクティブ状態にリセットされます。

割込みイネーブルレジスタ(読取り/書込み)

Addr. 03h	DQOE	ENBSY	ESINT	ERBF	ETMT	ETBE	IAS	EPD
	MSB						LSB	

EPD: プレゼンス検出割込みイネーブル。このビットが 1 で、かつプレゼンス検出フラグがセットされている場合、1-Wire リセットが送信されてプレゼンス検出パルスが発生した十分な時間後に、常に INTR がアクティブになります。

IAS: INTR のアクティブ状態。このビットで、INTR 端子のアクティブ状態が決まります。このビットが 1 の場合、INTR 端子はアクティブハイです。このビットが 0 の場合、INTR 端子はアクティブローです。

EBTE: 送信バッファEMPTY割込みイネーブル。このビットが 1 で、かつ送信バッファEMPTYフラグがセットされている場合、INTR 端子がアクティブになります。

ETMT: 送信シフトレジスタEMPTY割込みイネーブル。このビットが 1 で、かつ送信シフトレジスタEMPTYフラグがセットされている場合、INTR 端子がアクティブになります。

ERBF: 受信バッファフル割込みイネーブル。このビットが 1 で、かつ受信バッファフルフラグがセットされている場合、INTR 端子がアクティブになります。

ESINT: スレーブ割込みイネーブル。このビットが 1 で、かつスレーブ割込みフラグがセットされている場合、INTR 端子がアクティブになります。

ENBSY: ノットビジー割込みイネーブル。このビットが 1 で、かつノットビジーフラグがセットされている場合、INTR 端子がアクティブになります。

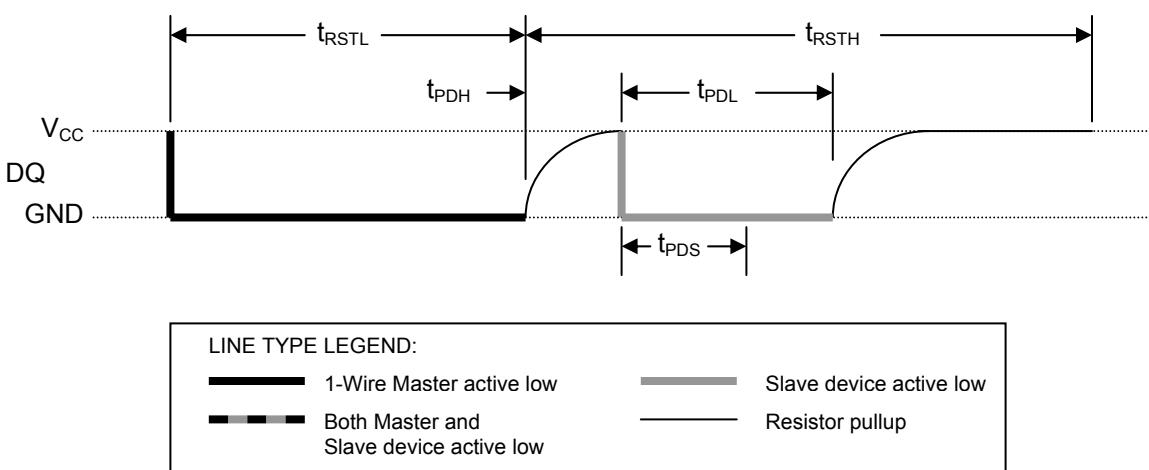
DQOE: DQ 出力イネーブル。このビットは、DQ バスの制御を選択する役割を果たします。0 がセットされていると(デフォルト)、バスは通常通り 1-Wire マスタによって制御されます。1 がセットされていると、コマンドレジスタ内の DQ0 ビットによってバスの状態が直接制御されます。通電時またはリセット時におけるこのビットのデフォルト状態は 0 であり、DQ0 ビットを通して手作業でバスを制御したい場合以外は、0 のままにしておく必要があります。

I/O シグナリング

1-Wire バスでは、データの完全性を保証するため厳密なシグナリングプロトコルが要求されます。1-Wire マスタが使用する 4 つのプロトコルは、初期化シーケンス(リセットパルスとそれに続くプレゼンスパルス)、0 書込み、1 書込み、およびデータ読取りです。プレゼンスパルスを除いて、マスタがこれらすべての種類のシグナリングを開始します。

バススレーブと何らかの通信を開始するために必要となる初期化シーケンスを、図 1 に示します。リセットパルスに続くプレゼンスパルスは、スレーブがROMコマンドを受け取る用意ができたことを示します。1-Wire マスタは t_{RSTL} の間リセットパルスを送信します。次に 1-Wire バスラインがプルアップ抵抗によってハイに強制されます。DQ 端子での立上りエッジ検出後、スレーブは t_{PDH} だけ待機し、次に t_{PDL} の間プレゼンスパルスを送信します。マスタはスレーブの応答から t_{PDS} 後にバスをサンプリングして、有効なプレゼンスパルスがないか調べます。このサンプリングの結果は、割込みレジスタのPDRビットに格納されます。リセットのタイムスロットは、マスタがバスを解放してから t_{RSTH} 後に終了します。

1-Wire 初期化シーケンス(リセットパルスとプレゼンスパルス) 図 1



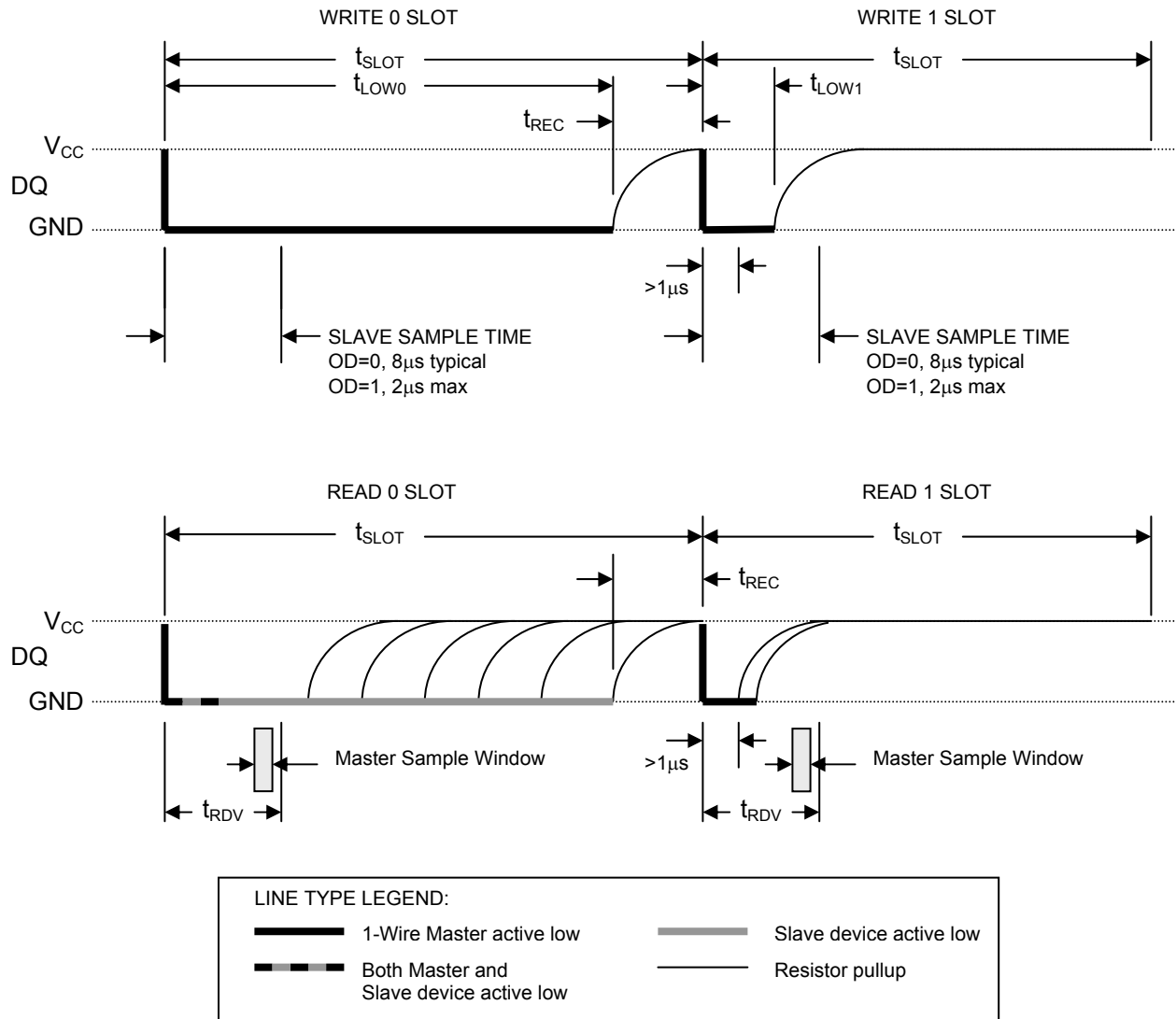
書込みタイムスロット

書込みタイムスロットは、1-Wire マスタが 1-Wire バスラインを論理ハイ(インアクティブ)レベルから論理ローレベルに引き下げたとき開始されます。マスタは、 t_{LOW1} 後にラインを解放して論理ハイレベルに戻すことによって、1 書込みのタイムスロットを生成します。0 書込みタイムスロットを生成するには、 t_{LOW0} の間ラインをローに維持します。スレーブデバイスは、ラインがローになって 15 μ s~60 μ sの間(オーバドライブモードなら 2 μ s~6 μ sの間)に、1-Wire バスラインをサンプリングします。サンプリング時にラインがハイなら、1 書込みが発生します。サンプリング時にラインがローなら、0 書込みが発生します(図 2 参照)。

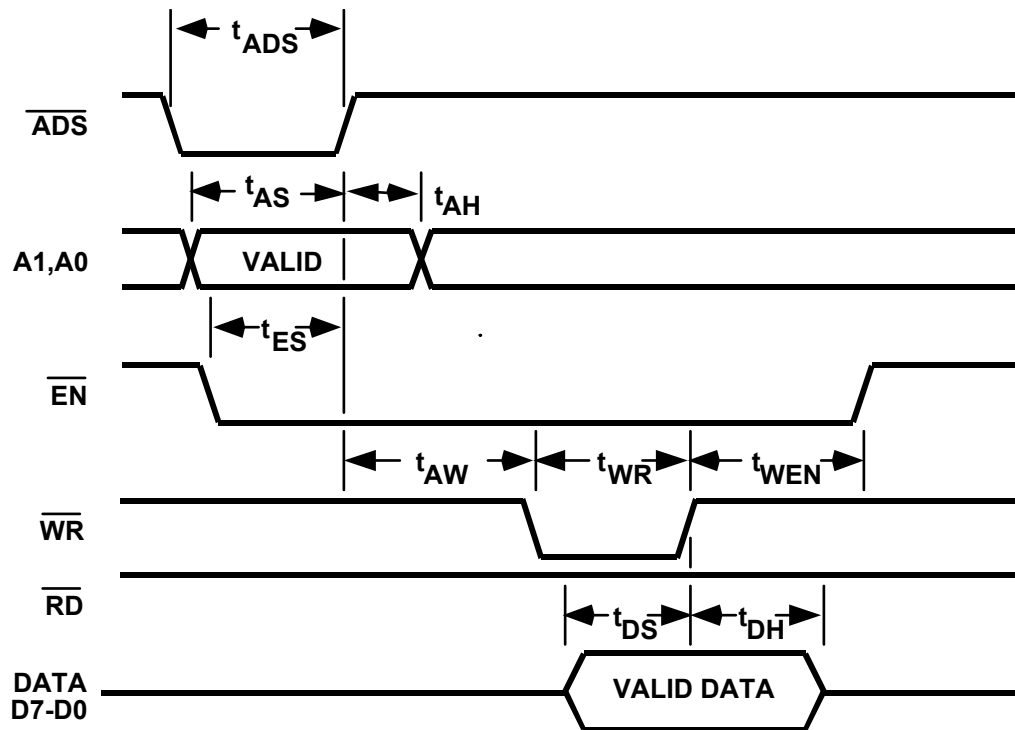
読取りタイムスロット

読取りタイムスロットは、1-Wire マスタが少なくとも 1 μ sの間バスをローに引き下げ、それから解放したときに開始されます。スレーブデバイスの応答が 0 の場合、最大 60 μ s (オーバドライブモードでは 6 μ s)の間ラインをローに維持し続け、それ以外の場合は直ちにラインを解放します。マスタは、読取りタイムスロットの開始から t_{RDV} 後にデータをサンプリングします。マスタは、 t_{SLOT} 後に読取りタイムスロットを終了します。詳細については、図 2 を参照してください。

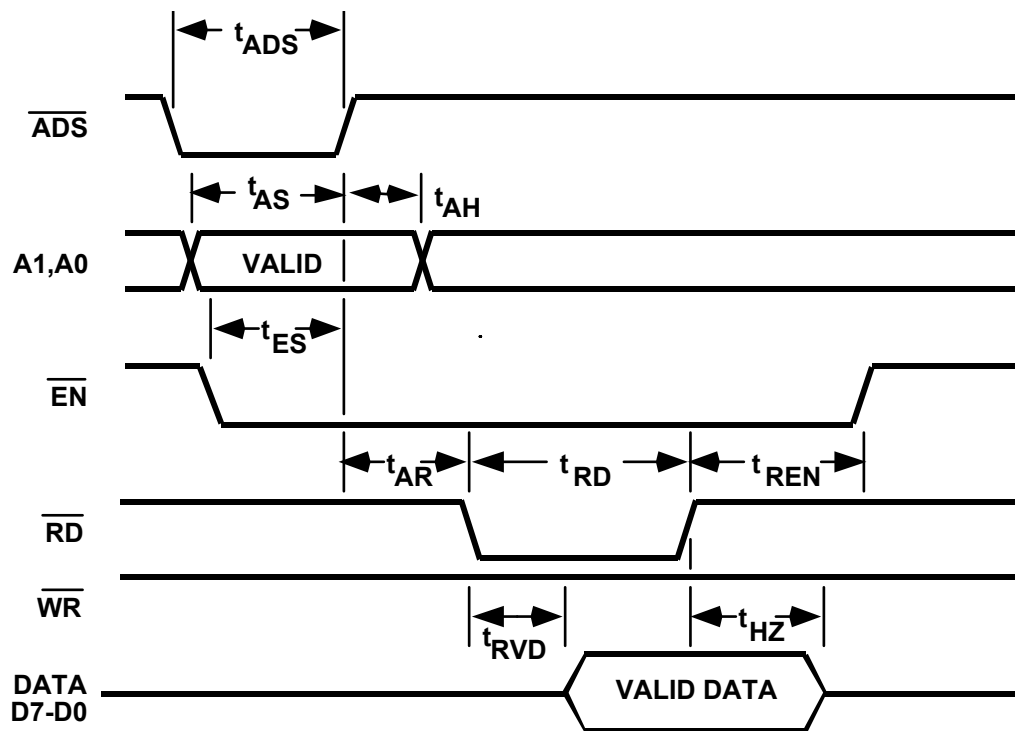
1-Wire 書込みおよび読取りタイムスロット 図 2



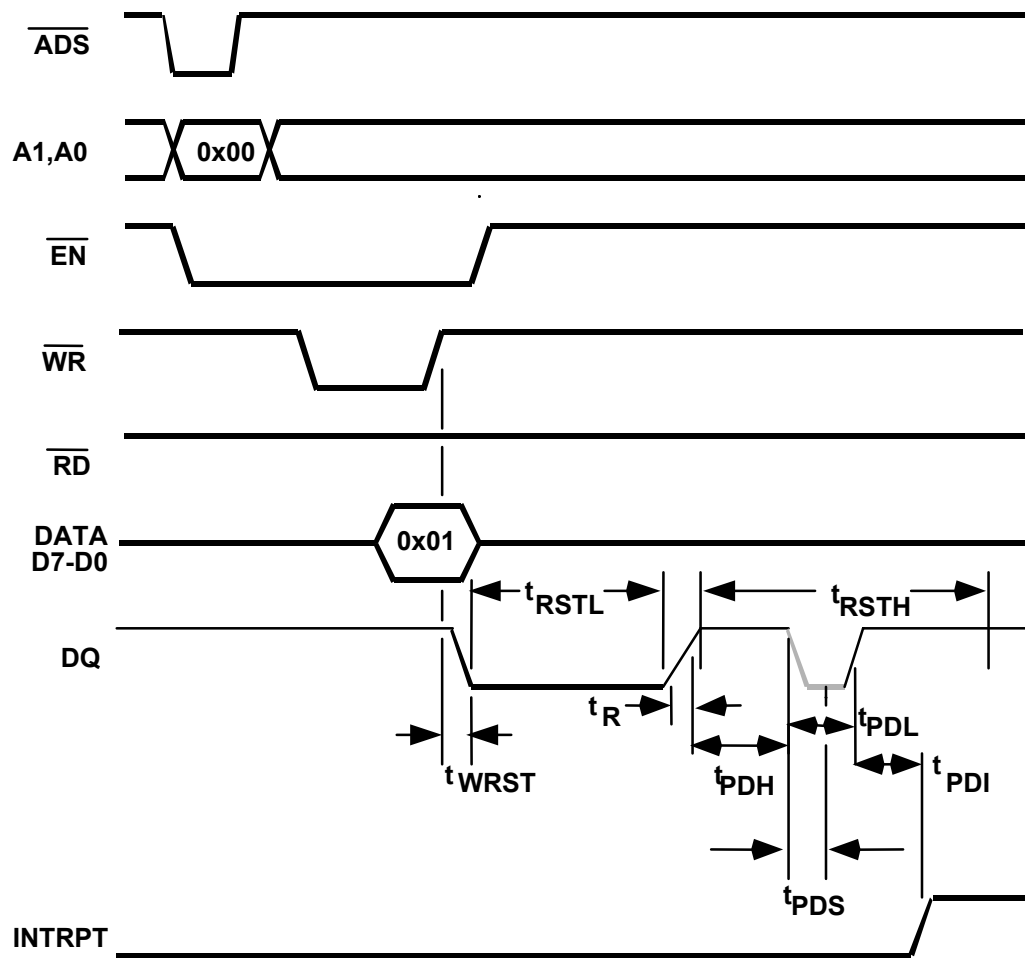
書き込みサイクル



読取りサイクル



1-Wire リセットの生成



タイミング仕様

Symbol	Parameter	Conditions	Min	Max	Units
t_{ADS}	Address Strobe Width	Note 1,3	60		ns
t_{AH}	Address Hold Time	Note 1,3	0		ns
t_{AR}	Address Latch to Read	Note 1,3	60		ns
t_{AS}	Address Setup Time	Note 1,3	60		ns
t_{AW}	Address Latch to Write	Note 1,3	60		ns
t_{DH}	Data Hold Time	Note 1	30		ns
t_{DS}	Data Setup Time	Note 1	30		ns
t_{ES}	Enable Setup Time	Note 1	60		ns
t_{HZ}	RD to Floating Data Delay	Note 1	0	100	ns
t_{LOW0}	Write 0 Low Time, OD=0 Write 0 Low Time, OD=1	63τ	63	78.75	μs
		8τ	8	10	
t_{LOW1}	Write 1 Low Time, OD=0 Write 1 Low Time, OD=1	6τ	6	7.5	μs
		1τ	1	1.25	
t_{PDH}	Presence Detect High Normal Presence Detect High Overdrive	Note 4	26	60	μs
			2.6	6	
t_{PDI}	Presence Detect to INTR	Note 1	0	100	ns
t_{PDL}	Presence Detect Low Normal Presence Detect Low Overdrive	Note 4	104	240	μs
			10.4	24	
t_{PDS}	Presence Detect Sample, OD=0 Presence Detect Sample, OD=1	30τ (Note 2) 3τ	30	37.5	μs
			3	3.75	
t_{RD}	RD Strobe Width	Note 1	125		ns
t_{RDV}	Read Data Valid Normal Read Data Valid Overdrive		15		μs
			2		
t_{REC}	Recovery Time		1		μs
t_{REN}	Enable Hold Time from RD	Note 1	20		ns
t_{RSTH}	Reset Time High, OD=0 Reset Time High, OD=1		500	625	μs
			50	62.5	
t_{RSTL}	Reset Time Low, OD=0 Reset Time Low, OD=1		488	610	μs
			61	76.25	
t_{RVD}	Delay from RD to Data	Note 1		60	ns
t_{SLOT}	Time Slot, OD=0 Time Slot, OD=1		73	91.25	μs
			11	13.75	
τ	Timebase Period		1	1.25	μs
t_{WEN}	Enable Hold Time from WR	Note 1	20		ns
t_{WR}	WR Strobe Width	Note 1	100		ns
t_{WRST}	WR High to Reset	Note 1	0	100	ns

注:

- これらの値は、回路の実現に使用するプロセスによって変わります。ここに示した値は、あくまでも例に過ぎません。
- 1-Wire マスタは、リセットから最大 60 μs (オーバドライブ時は 6 μs)の間、DQ ラインの立下りエッジが検出されるのを待ちます。
- ADSがロー固定の場合、 t_{AR} と t_{AW} は t_{ES} が基準になります。したがって \overline{RD} または \overline{WR} は \overline{EN} がローになって少なくとも $t_{ES} + t_{AR}$ または $t_{ES} + t_{AW}$ 後に発生する必要があります。
- 1-Wire のスレーブが決定するタイミング

参考資料:

[1] *Book of iButton Standards*, Dallas Semiconductor発行、
<http://www.ibutton.com/ibuttons/standard.pdf>で入手可能(英語のみ)