

**TOSHIBA**

東芝 オリジナル CMOS 16ビット マイクロコントローラ

**TLCS-900/L1 シリーズ**

TMP91FW27UG

TMP91FW27FG

Not Recommended  
for New Design

株式会社 **東芝** セミコンダクター社

## はじめに

この度は弊社 16 ビットマイクロコントローラ TLCS-900/L1 シリーズ、TMP91FW27 をご利用いただき、誠にありがとうございます。

本 LSI をご利用になる前に、「使用上の注意、制限事項」の章を参照されませうことをお願いいたします。

特に下記に示す注意事項に関しましては、十分にご注意願います。

### ホルト状態からの解除に関する注意事項

通常は、割り込みによってホルト状態を解除することができませんが、HALT モードが IDLE1、STOP モードに設定されている状態 (IDLE2 は対象外) で、CPU が HALT モードに移行しようとしている期間 ( $f_{FPH}$  約 5 クロックの間) に、HALT モードを解除可能な割り込み ( $\overline{NMI}$ , INT0, INTRTC) が入力されても、ホルトが解除できない場合があります (割り込み要求は内部に保留されます)。

HALT モードへ完全に移行された後に、再度割り込みが発生すれば、問題なく HALT モードを解除できますが、割り込み処理は内部に保留された割り込みと現在の割り込みを比較し、その優先順位に従って順次処理されます。

低電圧/低消費電力

## CMOS 16 ビット マイクロコントローラ TMP91FW27UG / TMP91FW27FG

### 1. 概要と特長

TMP91FW27 は、低電圧/低消費電力動作が可能な高速・高機能 16 ビットマイクロコントローラです。

TMP91FW27UG、TMP91FW27FG は、64 ピンフラットパッケージ製品です。

特長は次のとおりです。

- (1) オリジナル 16 ビット CPU (900/L1\_CPU 使用)
  - TLCS-90 と命令ニモニックで上位互換
  - 16 M バイトのリニアアドレス空間
  - 汎用レジスタ&レジスタバンク方式
  - 16 ビット乗除算命令、ビット転送/演算命令
  - マイクロ DMA: 4 チャンネル (593 ns/2 バイト@27MHz)
- (2) 最小命令実行時間: 148 ns (@27 MHz)
- (3) 内蔵 RAM: 12 K バイト  
内蔵 ROM: 128 K バイトフラッシュメモリ  
: 4 K バイトマスク ROM (ブート機能用)

### 当社半導体製品取り扱い上のごお願い

20070701-JA

- 当社は品質、信頼性の向上に努めておりますが、一般に半導体製品は誤作動したり故障することがあります。当社半導体製品をご使用いただく場合は、半導体製品の誤作動や故障により、生命・身体・財産が侵害されることのないように、購入者側の責任において、機器の安全設計を行うことをお願いします。  
なお、設計に際しては、最新の製品仕様をご確認の上、製品保証範囲内でご使用いただくと共に、考慮されるべき注意事項や条件について「東芝半導体製品の取り扱い上のご注意とお願い」、「半導体信頼性ハンドブック」などをご確認ください。
- 本資料に掲載されている製品は、一般的電子機器（コンピュータ、パーソナル機器、事務機器、計測機器、産業用ロボット、家電機器など）に使用されることを意図しています。特別に高い品質・信頼性が要求され、その故障や誤作動が直接人命を脅かしたり人体に危害を及ぼす恐れのある機器（原子力制御機器、航空宇宙機器、輸送機器、交通信号機器、燃焼制御、医療機器、各種安全装置など）にこれらの製品を使用すること（以下“特定用途”という）は意図もされていませんし、また保証もされていません。本資料に掲載されている製品を当該特定用途に使用することは、お客様の責任でなされることとなります。
- 本資料に掲載されている製品を、国内外の法令、規則及び命令により製造、使用、販売を禁止されている応用製品に使用することはできません。
- 本資料に掲載してある技術情報は、製品の代表的動作・応用を説明するためのもので、その使用に際して当社及び第三者の知的財産権その他の権利に対する保証または実施権の許諾を行うものではありません。
- 本資料に掲載されている製品の RoHS 適合性など、詳細につきましては製品個別に必ず弊社営業窓口までお問合せください。本資料に掲載されている製品のご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令などの法令を十分調査の上、かかる法令に適合するようご使用ください。お客様が適用される法令を遵守しないことにより生じた損害に関して、当社は一切の責任を負いかねます。
- 本資料の掲載内容は、技術の進歩などにより予告なしに変更されることがあります。

本製品は、米国 SST 社 (Silicon Storage Technology, Inc.) からライセンスを受けた Super Flash® 技術を使用しています。Super Flash® は SST 社の登録商標です。

- (4) 外部メモリ拡張
- 16 M バイト (プログラム/データ共通) まで拡張可能
  - 外部データバス 8/16 ビット幅共存可能  
…ダイナミックデータバスサイジング
- (5) 8 ビットタイマ: 6 チャンネル
- (6) 16 ビットタイマ: 1 チャンネル
- (7) 汎用シリアルインタフェース: 2 チャンネル
- UART/同期両モード対応: 2 チャンネル
  - IrDA ver1.0 (115.2 kbps) 対応モード選択可能: 1 チャンネル
- (8) シリアルバスインタフェース: 1 チャンネル  
I<sup>2</sup>C バスモード/クロック同期式モード選択可能
- (9) 10 ビット AD コンバータ (サンプルホールド回路内蔵): 4 チャンネル
- (10) ウォッチドッグタイマ
- (11) 時計用タイマ
- (12) チップセレクト/ウェイトコントローラ: 4 ブロック
- (13) 割り込み機能: 34 本
- CPU 9 本 …… ソフトウェア割り込み命令、未定義命令実行違反
  - 内部 21 本 …… 7 レベルの優先順位の設定が可能
  - 外部 4 本 …… 7 レベルの優先順位の設定が可能  
(3 本はエッジの極性選択可能)
- (14) 入出力ポート: 53 端子
- (15) スタンバイ機能  
3 種類のホールドモード (プログラマブル IDLE2, IDLE1, STOP)
- (16) クロック制御機能
- クロックギア機能: 高周波クロック  $f_c \sim f_c/16$  まで切り替え可能
  - 時計用クロック ( $f_s = 32.768 \text{ kHz}$ )
- (17) 動作電圧
- $V_{CC} = 2.7 \sim 3.6 \text{ V}$  ( $f_c \text{ max} = 27 \text{ MHz}$  Flash 読み出し動作時)
  - $V_{CC} = 2.2 \sim 3.6 \text{ V}$  ( $f_c \text{ max} = 16 \text{ MHz}$  Flash 読み出し動作時)
  - $V_{CC} = 2.7 \sim 3.6 \text{ V}$  ( $f_c \text{ max} = 27 \text{ MHz}$  Flash 消去/書き込み動作時)
- (18) パッケージ: LQFP64-P-1010-0.50D (TMP91FW27UG)  
QFP64-P-1414-0.80A (TMP91FW27FG)

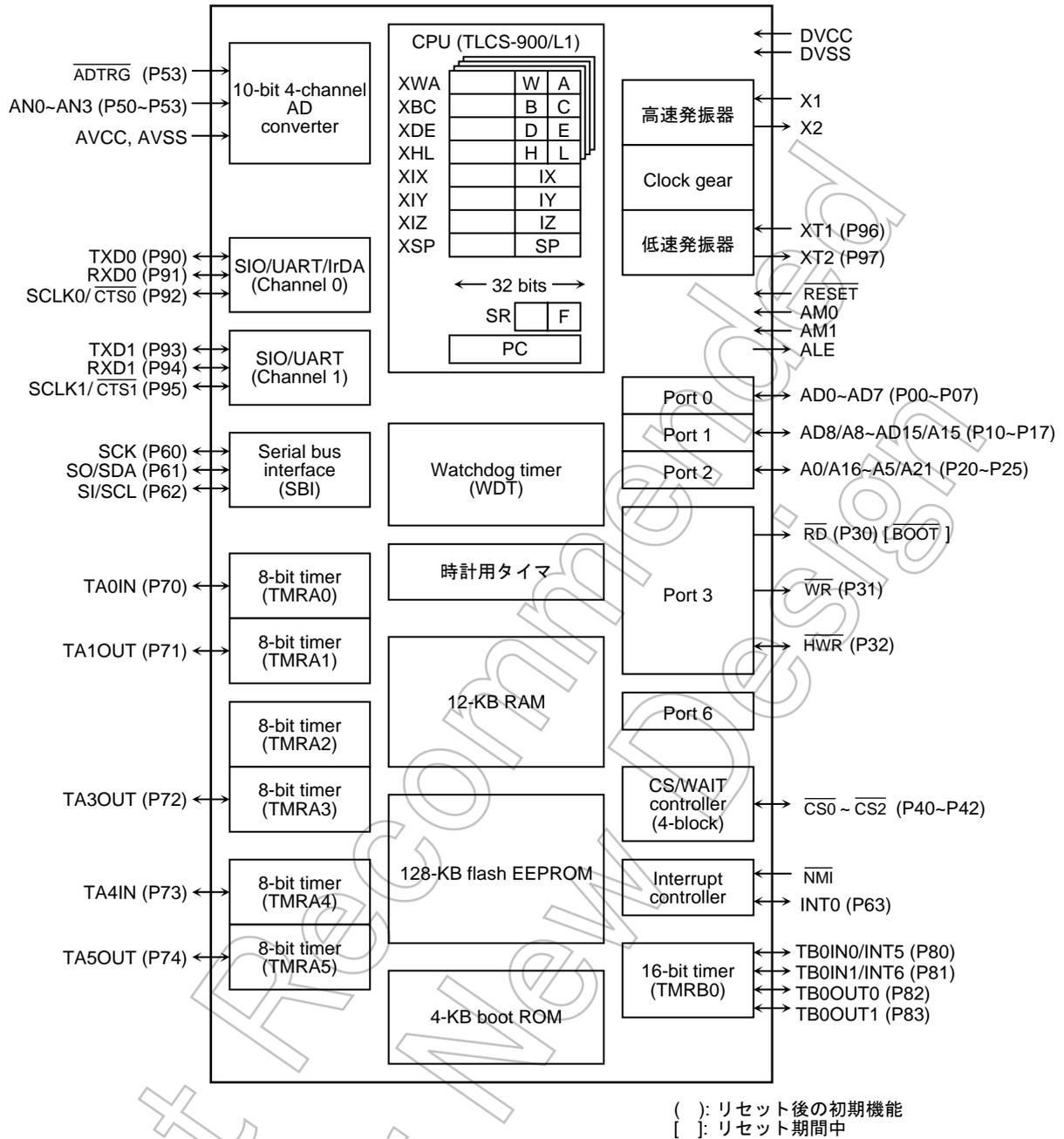


図1.1 TMP91FW27 ブロック図

## 2. ピン配置とピン機能

TMP91FW27 のピン配置図および入出力ピンの名称と概略機能を示します。

### 2.1 ピン配置図

TMP91FW27UG、TMP91FW27FG ピン配置図は、図2.1.1のとおりです。

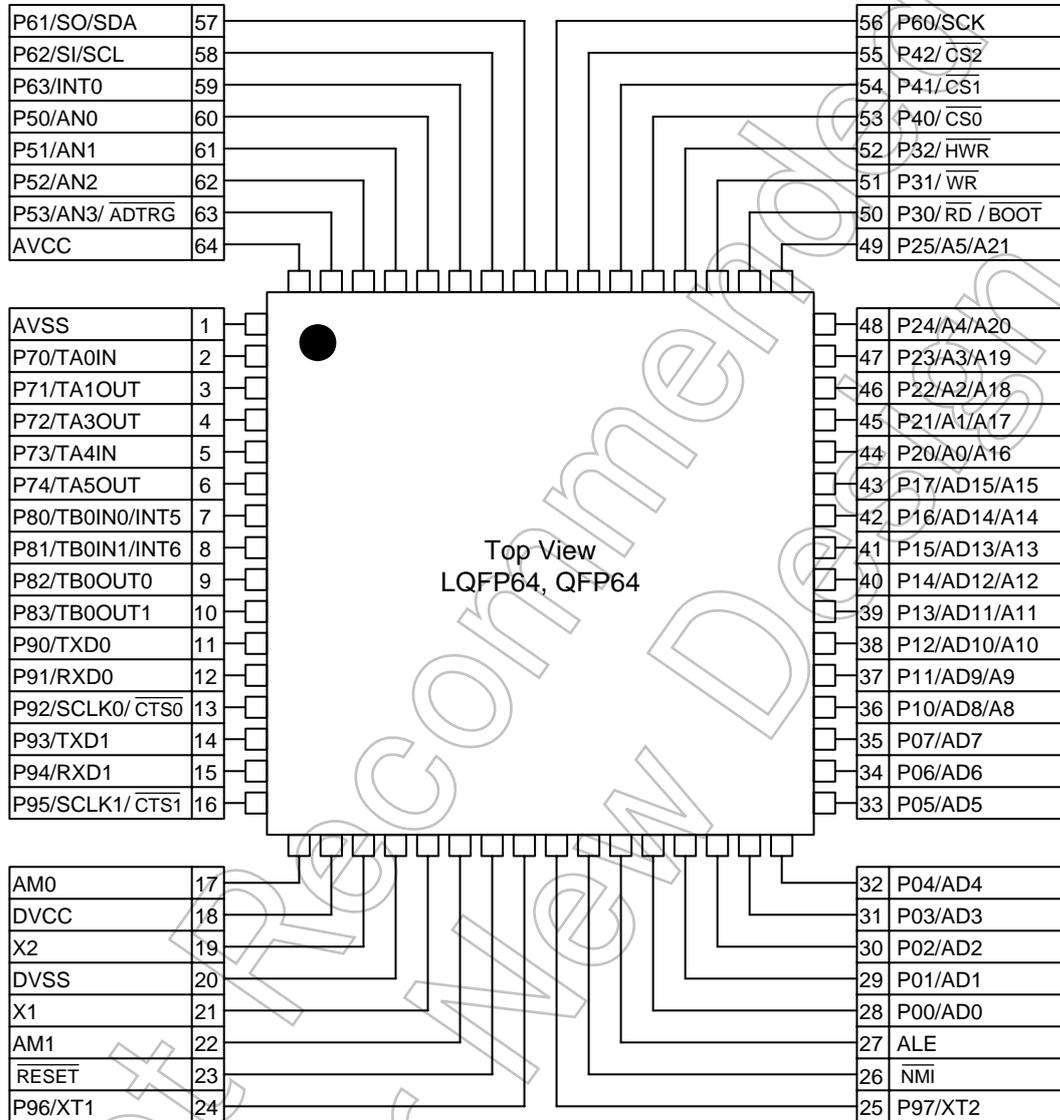


図2.1.1 ピン配置図 (64 ピン LQFP, QFP)

## 2.2 ピン名称と機能

入出力ピンの名称と機能は、表 2.2.1~ 表 2.2.2のとおりです。

表2.2.1 ピン名称と機能 (1/2)

ピン名称	ピン数	入出力	機能
P00~P07 AD0~AD7	8	入出力 入出力	ポート 0: ビット単位で入出力の設定ができる入出力ポート アドレスデータ (下位): アドレス/データバス 0~7
P10~P17 AD8~AD15 A8~A15	8	入出力 入出力 出力	ポート 1: ビット単位で入出力の設定ができる入出力ポート アドレスデータ (上位): アドレス/データバス 8~15 アドレス: アドレスバス 8~15
P20~P25 A0~A5 A16~A21	6	入出力 出力 出力	ポート 2: ビット単位で入出力の設定ができる入出力ポート アドレス: アドレスバス 0~5 アドレス: アドレスバス 16~21
P30 $\overline{RD}$  $\overline{BOOT}$	1	出力 出力  入力	ポート 30: 出力専用ポート リード: 外部メモリをリードするためのストローブ信号 ( $P3 < P30 > = 0$ , $P3FC < P30F > = 1$ ) にすることによって、内部エリアをリードしたときも $\overline{RD}$ が出ます。 シングルブートモード設定端子です (リセット期間中のみ)。 (詳細は “3.2.3 動作モード” を参照してください。)
P31 $\overline{WR}$	1	出力 出力	ポート 31: 出力専用ポート ライト: AD0~AD7 端子のデータをライトするためのストローブ信号
P32 $\overline{HWR}$	1	入出力 出力	ポート 32: 入出力ポート (プルアップ付) 上位ライト: AD8~AD15 端子のデータをライトするためのストローブ信号
P40 $\overline{CS0}$	1	入出力 出力	ポート 40: 入出力ポート (プルアップ付) チップセレクト 0: アドレスが指定したアドレス領域内なら “0” を出力
P41 $\overline{CS1}$	1	入出力 出力	ポート 41: 入出力ポート (プルアップ付) チップセレクト 1: アドレスが指定したアドレス領域内なら “0” を出力
P42 $\overline{CS2}$	1	入出力 出力	ポート 42: 入出力ポート (プルアップ付) チップセレクト 2: アドレスが指定したアドレス領域内なら “0” を出力
P50~P53 AN0~AN3 $\overline{ADTRG}$	4	入力 入力 入力	ポート 5: 入力専用ポート アナログ入力: AD コンバータ用のアナログ入力端子 AD トリガ: AD コンバータの外部スタート要求端子 (P53 と兼用)
P60 SCK	1	入出力 入出力	ポート 60: 入出力ポート シリアルバスインタフェースの SIO モード時のクロック入出力端子
P61 SO SDA	1	入出力 出力 入出力	ポート 61: 入出力ポート シリアルバスインタフェースの SIO モード時のデータ送信端子 シリアルバスインタフェースの I <sup>2</sup> C モード時のデータ送受信端子 プログラムによりオープンドレイン出力端子となります。
P62 SI SCL	1	入出力 入力 入出力	ポート 62: 入出力ポート シリアルバスインタフェースの SIO モード時のデータ受信端子 シリアルバスインタフェースの I <sup>2</sup> C モード時のクロック入出力端子 プログラムによりオープンドレイン出力端子となります。
P63 INT0	1	入出力 入力	ポート 63: 入出力ポート (シュミット入力) 割り込み要求端子 0: レベル/立ち上がり/立ち下がりエッジを選択可能な割り込み要求端子
P70 TA0IN	1	入出力 入力	ポート 70: 入出力ポート 8 ビットタイマ 0 入力: 8 ビットタイマ TMRA0 の入力端子
P71 TA1OUT	1	入出力 出力	ポート 71: 入出力ポート 8 ビットタイマ 1 出力: 8 ビットタイマ TMRA0 または TMRA1 の出力端子

表2.2.2 ピン名称と機能 (2/2)

ピン名称	ピン数	入出力	機能
P72 TA3OUT	1	入出力 出力	ポート 72: 入出力ポート 8ビットタイマ3出力: 8ビットタイマ TMRA2 または TMRA3 の出力端子
P73 TA4IN	1	入出力 入力	ポート 73: 入出力ポート 8ビットタイマ4入力: 8ビットタイマ TMRA4 の入力端子
P74 TA5OUT	1	入出力 出力	ポート 74: 入出力ポート 8ビットタイマ5出力: 8ビットタイマ TMRA4 または TMRA5 の出力端子
P80 TB0IN0  INT5	1	入出力 入力  入力	ポート 80: 入出力ポート 16ビットタイマ0入力0: 16ビットタイマ TMRB0 のカウント/キャプチャトリガ入力になります。 割り込み要求端子 5: 立ち上がり/立ち下がりエッジを選択可能な割り込み要求端子
P81 TB0IN1  INT6	1	入出力 入力  入力	ポート 81: 入出力ポート 16ビットタイマ0入力1: 16ビットタイマ TMRB0 のカウント/キャプチャトリガ入力になります。 割り込み要求端子 6: 立ち上がりエッジの割り込み要求端子
P82 TB0OUT0	1	入出力 出力	ポート 82: 入出力ポート 16ビットタイマ0出力0: 16ビットタイマ TMRB0 の出力端子
P83 TB0OUT1	1	入出力 出力	ポート 83: 入出力ポート 16ビットタイマ0出力1: 16ビットタイマ TMRB0 の出力端子
P90 TXD0	1	入出力 出力	ポート 90: 入出力ポート シリアル0送信データ プログラムによりオープンドレイン出力端子となります。
P91 RXD0	1	入出力 入力	ポート 91: 入出力ポート シリアル0受信データ
P92 SCLK0 CTS0	1	入出力 入出力 入力	ポート 92: 入出力ポート シリアル0クロック入出力 シリアル0データ送信可能 (Clear to send)
P93 TXD1	1	入出力 出力	ポート 93: 入出力ポート シリアル1送信データ プログラムによりオープンドレイン出力端子となります。
P94 RXD1	1	入出力 入力	ポート 94: 入出力ポート シリアル1受信データ
P95 SCLK1 CTS1	1	入出力 入出力 入力	ポート 95: 入出力ポート シリアル1クロック入出力 シリアル1データ送信可能 (Clear to send)
P96 XT1	1	入出力 入力	ポート 96: 入出力ポート (オープンドレイン出力端子) 低周波発振子接続端子
P97 XT2	1	入出力 出力	ポート 97: 入出力ポート (オープンドレイン出力端子) 低周波発振子接続端子
ALE	1	出力	アドレスラッチイネーブル (ノイズ削減のため出力禁止に設定できます。)
$\overline{\text{NMI}}$	1	入力	ノンマスクブル割り込み要求端子: 立ち下がりエッジの割り込み要求端子 (シュミット入力) プログラムにより、立ち上がりエッジでも割り込み要求可能となります。
AM0, AM1	2	入力	動作モード: AM1 = "1", AM0 = "1" に固定してください。
RESET	1	入力	リセット: LSI を初期化します (シュミット入力、ブルアップ抵抗付き)。
AVCC	1		AD コンバータ電源端子と AD コンバータ用基準電源(H)共用端子
AVSS	1		AD コンバータ GND 端子 (0 V) と AD コンバータ用基準電源 (L) 共用端子
X1	1	入出力	高周波発振子接続端子
X2	1	入出力	高周波発振子接続端子
DVCC	1		電源端子 (全 DVCC 端子を電源に接続してください。)
DVSS	1		GND 端子 (全 DVSS 端子を GND (0 V) に接続してください。)

### 3. 動作説明

TMP91FW27 について、ハードウェアの構成、および、その動作を説明します。

本デバイスは、TMP91CP27 の内蔵マスク ROM を 128 K バイト内蔵フラッシュメモリに置き替え拡張したものです。その他の構成、および機能は、TMP91CP27 と同一です。

ここに記載されていない機能については、TMP91CP27 のデータシートを参照してください。

#### 3.1 メモリマップ

図 3.1.1 に、TMP91FW27 のシングルチップモード時のメモリマップと CPU の各アドレッシングモードのアクセス範囲を示します。

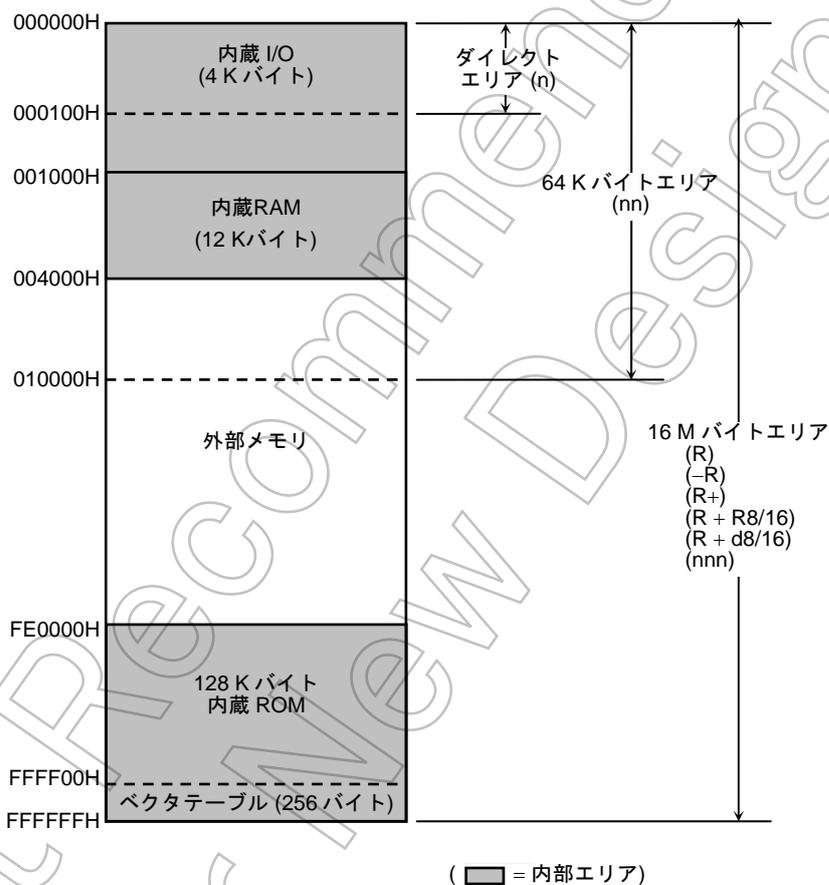


図3.1.1 TMP91FW27 メモリマップ (シングルチップモード)

## 3.2 フラッシュメモリ

TMP91FW27は3V単一電源による電氣的消去および書き込み可能なフラッシュメモリを内蔵しています。

フラッシュメモリの書き込みおよび消去は、JEDEC標準コマンドで行います。コマンド入力後、書き込みおよび消去が内部で自動的に行われます。また、消去動作は一括消去、セクタ単位での消去ができます。

内蔵フラッシュメモリの構成およびその動作を説明します。

### 3.2.1 特長

- 書き込み/消去時の電源電圧  
Vcc = 2.7~3.6 V (-10°C~40°C)
- 構成  
64 K × 16 ビット (128 K バイト)
- 機能  
1ワード書き込み  
チップイレース  
セクタイレース  
データポーリング/トグルビット
- セクタサイズ  
4 K バイト × 32
- モードコントロール  
JEDEC標準コマンド準拠
- プログラミング方法  
オンボード書き込み  
パラレルライター書き込み
- セキュリティ  
ライトプロテクト  
リードプロテクト

### 3.2.2 ブロック図

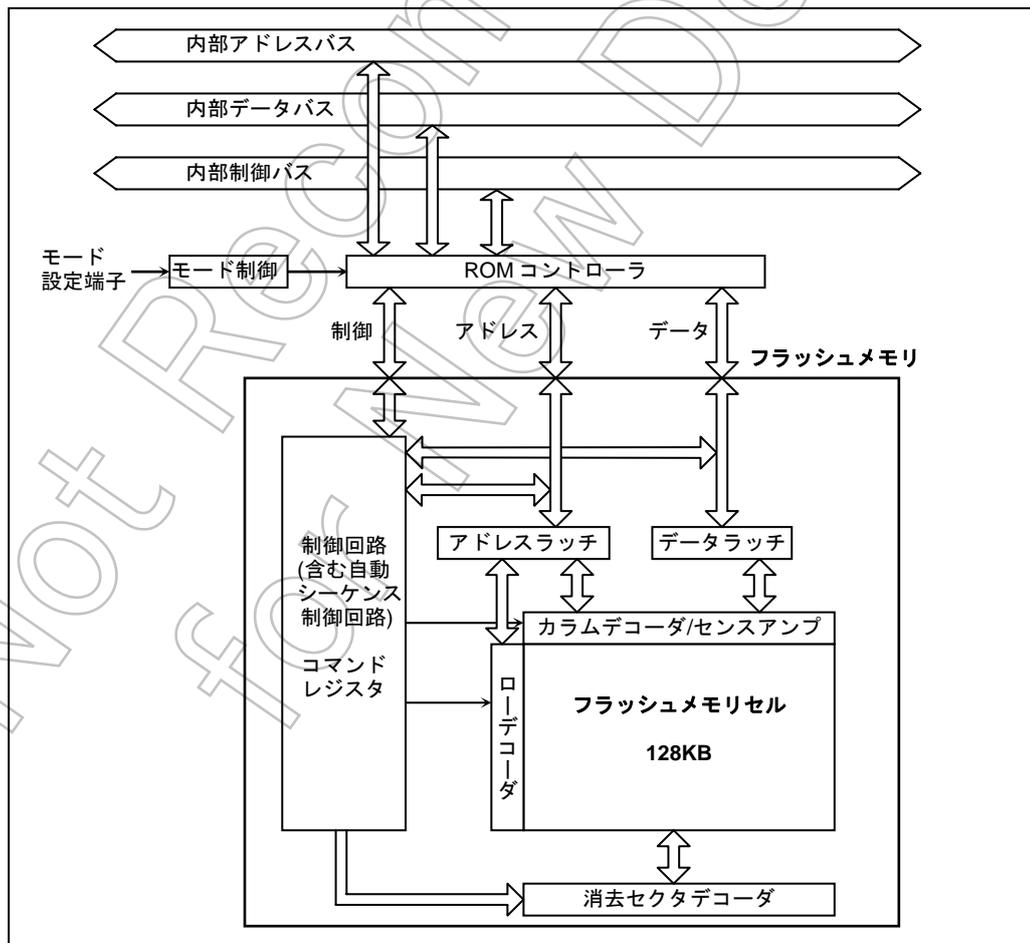


図 3.2.1 フラッシュ部ブロック図

### 3.2.3 動作モード

#### 3.2.3.1 概要

内蔵フラッシュメモリの書き込み/消去などの制御は、以下の3通りの動作状態（モード）が存在します。

表 3.2.1 動作モード説明

動作モード名	動作の内容
シングルチップモード	リセット解除後、内蔵のフラッシュメモリから起動します。 本動作モードの中で、ユーザのアプリケーションプログラムを実行するモードと、ユーザのセット上でフラッシュメモリの書き替えを実行するモードとに分けて定義します。前者を「ノーマルモード」、後者を「ユーザブートモード」と呼びます。 この両者の切り替えはユーザが独自に設定できます。 例えばポート 00 が “1” のときノーマルモード、“0” のときにユーザブートモードというように自由に設計することが可能です。 ユーザはアプリケーションプログラムの一部に切り替えを判定するためのルーチンを準備してください。
ノーマルモード	ユーザのアプリケーションプログラムより起動します。
ユーザブートモード	ユーザ指定の方法により内蔵フラッシュの書き替えを実行します。
シングルブートモード	リセット解除後、内蔵するブート ROM (Mask ROM) から起動します。ブート ROM には、シリアルポートを経由してユーザのセット上で書き込み/消去を行う為のプログラムを、デバイスの RAM 上に転送することができるアルゴリズムがプログラムされています。書き込みプログラムを RAM 上に搭載し、外部ホストから書き込みデータを受信しつつ、フラッシュへの書き込みコマンドを発行することで、フラッシュの書き込み/消去が実行できます。
ライターモード	汎用のプログラムライターで内蔵フラッシュメモリを書き込み/消去をするためのモードです。 プログラムライターのサポート状況については、当社営業窓口までお問い合わせください。

表 3.2.1のうち、ユーザブートモード、シングルブートモード、ライターモードの3つが内蔵フラッシュメモリの書き替えが可能な動作モードです。ユーザのセット上で書き込み/消去が可能なモードは、オンボードプログラミングモードと定義します。

オンボードプログラミングモードは、「シリアル I/O を利用した当社独自の書き込み/消去方式をサポートするシングルブートモード」、「シングルチップモード内においてユーザが独自に書き込み/消去方式を構築できるユーザブートモード」があります。

また、本デバイスはライターモード中に、ROM データの読み出しを禁止する「リードプロテクト機能」を持っています。プログラミング完了時にリードプロテクト機能をオンにしておくことで、第三者への ROM データ流出を阻止することができます。

シングルチップ、ブートおよびライタの各動作モードは、リセット状態で入力端子 AM0、AM1、 $\overline{\text{BOOT}}$  (P30)のレベルを外部で設定することにより決定されます。

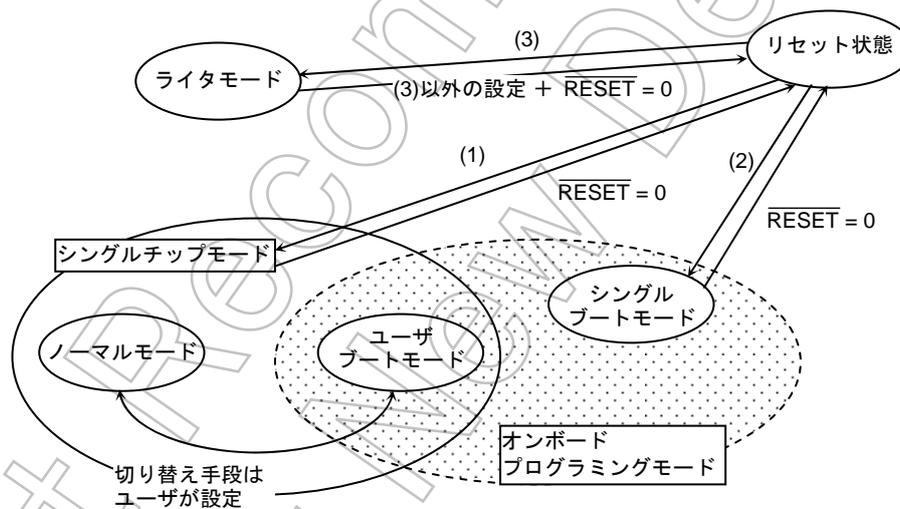
CPU はライタモードを除き、状態設定後リセットを解除することにより各動作モードで動作を開始します。ライタモードは $\overline{\text{RESET}} = "0"$ のまま使用します。それぞれモード設定後は動作中にレベルの変更がないようにしてください。以下に各動作モードの設定方法とモード遷移図を示します。

表 3.2.2 動作モード設定表

	動作モード	入力端子			
		RESET	BOOT (P30)	AM1	AM0
(1)	シングルチップモード (ノーマルおよびユーザブート)		1	1	1
(2)	シングルブートモード		0	1	1
(3)	ライタモード	0	—	1	0

注) P30 は出力専用ポートですが、リセット期間中のみはプルアップ抵抗付きの入力モードになります。リセット解除後は、動作モードにより P30 の状態は以下になります。

- シングルチップモード：出力専用ポート (プルアップ抵抗なし)
- シングルブートモード：プルアップ (入力無効, 出力ゲートはハイインピーダンス)



( ) 数字は上表の対応するモードの入力端子状態に設定することを意味します。

図 3.2.2 モード遷移図

### 3.2.3.2 リセット動作

本デバイスにリセットをかけるには、電源電圧が動作電圧範囲内で、かつ内部高周波発振器の発振が安定した状態で、少なくとも 10 システムクロック間  $\overline{\text{RESET}}$  入力を "0" にしてください。

### 3.2.3.3 モード別メモリマップ

本製品では動作モードごとにメモリマップが変わります。以下に、各動作モードのメモリマップとモード別セクタアドレス範囲表を示します。

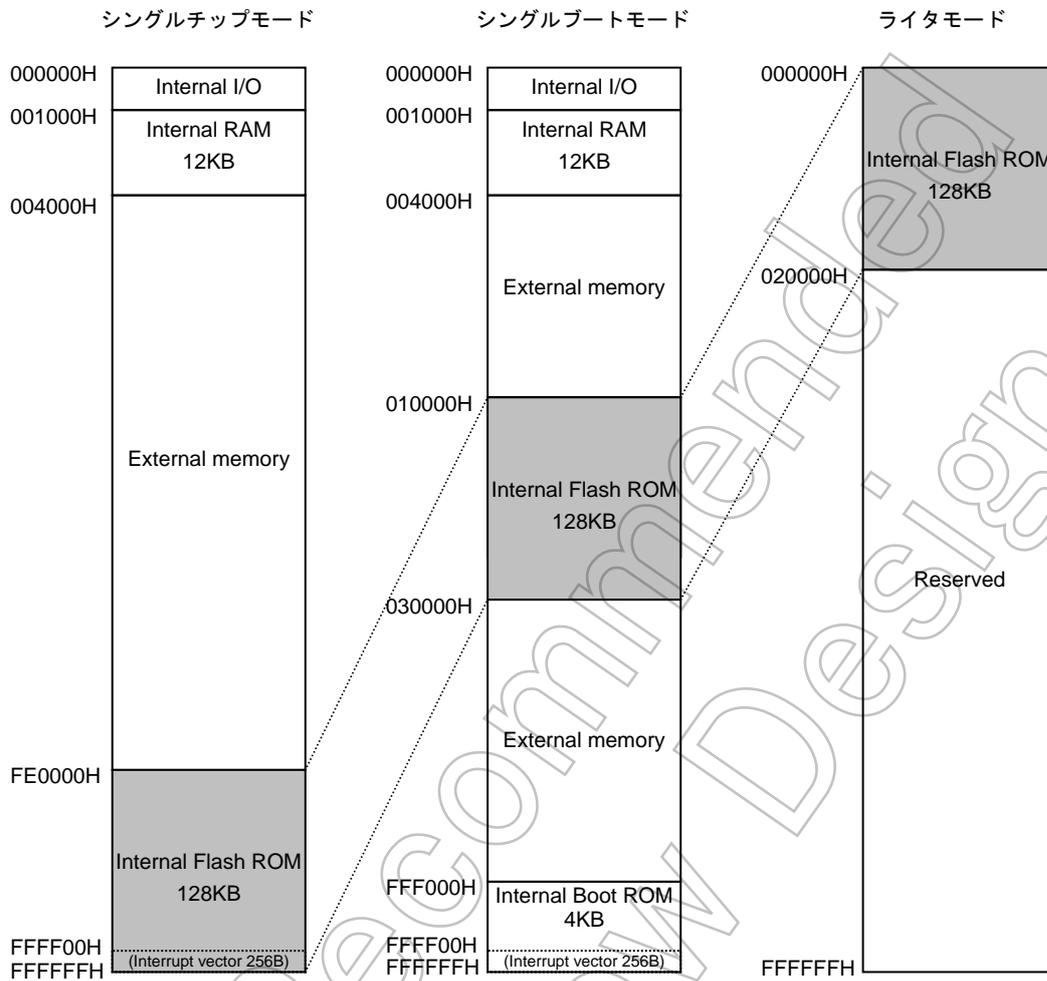


図 3.2.3 TMP91FW27 モード別メモリマップ

表 3.2.3 モード別セクタアドレス範囲表

	シングルチップモード	シングルブートモード
Sector-0	FE000H から FE0FFFH まで	10000H から 10FFFH まで
Sector-1	FE1000H から FE1FFFH まで	11000H から 11FFFH まで
Sector-2	FE2000H から FE2FFFH まで	12000H から 12FFFH まで
Sector-3	FE3000H から FE3FFFH まで	13000H から 13FFFH まで
Sector-4	FE4000H から FE4FFFH まで	14000H から 14FFFH まで
Sector-5	FE5000H から FE5FFFH まで	15000H から 15FFFH まで
Sector-6	FE6000H から FE6FFFH まで	16000H から 16FFFH まで
Sector-7	FE7000H から FE7FFFH まで	17000H から 17FFFH まで
Sector-8	FE8000H から FE8FFFH まで	18000H から 18FFFH まで
Sector-9	FE9000H から FE9FFFH まで	19000H から 19FFFH まで
Sector-10	FEA000H から FEAFFFH まで	1A000H から 1AFFFH まで
Sector-11	FEB000H から FEBFFFH まで	1B000H から 1BFFFH まで
Sector-12	FEC000H から FECFFFH まで	1C000H から 1CFFFH まで
Sector-13	FED000H から FEDFFFH まで	1D000H から 1DFFFH まで
Sector-14	FEE000H から FEEFFFH まで	1E000H から 1EFFFH まで
Sector-15	FEF000H から FEFFFFH まで	1F000H から 1FFFFH まで
Sector-16	FF0000H から FF0FFFH まで	20000H から 20FFFH まで
Sector-17	FF1000H から FF1FFFH まで	21000H から 21FFFH まで
Sector-18	FF2000H から FF2FFFH まで	22000H から 22FFFH まで
Sector-19	FF3000H から FF3FFFH まで	23000H から 23FFFH まで
Sector-20	FF4000H から FF4FFFH まで	24000H から 24FFFH まで
Sector-21	FF5000H から FF5FFFH まで	25000H から 25FFFH まで
Sector-22	FF6000H から FF6FFFH まで	26000H から 26FFFH まで
Sector-23	FF7000H から FF7FFFH まで	27000H から 27FFFH まで
Sector-24	FF8000H から FF8FFFH まで	28000H から 28FFFH まで
Sector-25	FF9000H から FF9FFFH まで	29000H から 29FFFH まで
Sector-26	FFA000H から FFAFFFH まで	2A000H から 2AFFFH まで
Sector-27	FFB000H から FFBFFFH まで	2B000H から 2BFFFH まで
Sector-28	FFC000H から FFCFFFH まで	2C000H から 2CFFFH まで
Sector-29	FFD000H から FFDFFFH まで	2D000H から 2DFFFH まで
Sector-30	FFE000H から FFEFFFH まで	2E000H から 2EFFFH まで
Sector-31	FFF000H から FFFFFFH まで	2F000H から 2FFFFH まで

### 3.2.4 シングルブートモード

内蔵ブートROM (マスク ROM) を起動して、外部から書き込みルーチン(ユーザ作成のブートプログラム)をRAMへ転送し、そのルーチンプログラムを利用してフラッシュメモリを書き込み/消去する方法です。このモードでは、内蔵ブートROMが割り込みベクタテーブルを含む領域にマッピングされ、内蔵ブートROMプログラムが起動されます。また、フラッシュメモリはブートROM領域と別のアドレス空間にマッピングされます(図 3.2.3参照)。

本デバイスの SIO (SIO1) とコントローラを接続し、コントローラ側からデバイスの内蔵RAMに書き込みプログラムを転送し、RAM上の書き込みルーチンを実行してフラッシュメモリの書き込みを行います。

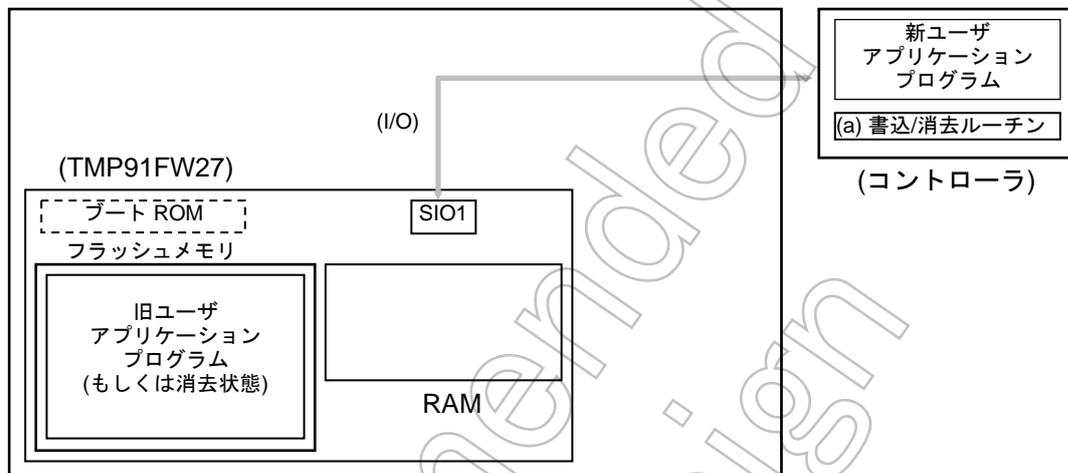
書き込みルーチンは、コントローラ側からコマンドおよび書き込みデータを送出することにより実行します。コントローラ側との通信の詳細は後述のプロトコルに従ってください。RAMへのプログラム転送は、ユーザのROMデータに対するセキュリティ確保のため、実行に先立ちユーザパスワードの照合を行います。パスワードが一致しない場合は、RAM転送そのものが実行されません。シングルブートモードでは、割り込み動作を禁止し、割り込み要求は割り込み要求フラグにて確認してください。

注) シングルブートモード時、ブートROMのプログラムはNORMALモードで動作します。  
書き換えプログラムの中で他のモードへ遷移させないでください。

## 3.2.4.1 内蔵ブートROMの書き込み/消去アルゴリズムを利用する場合

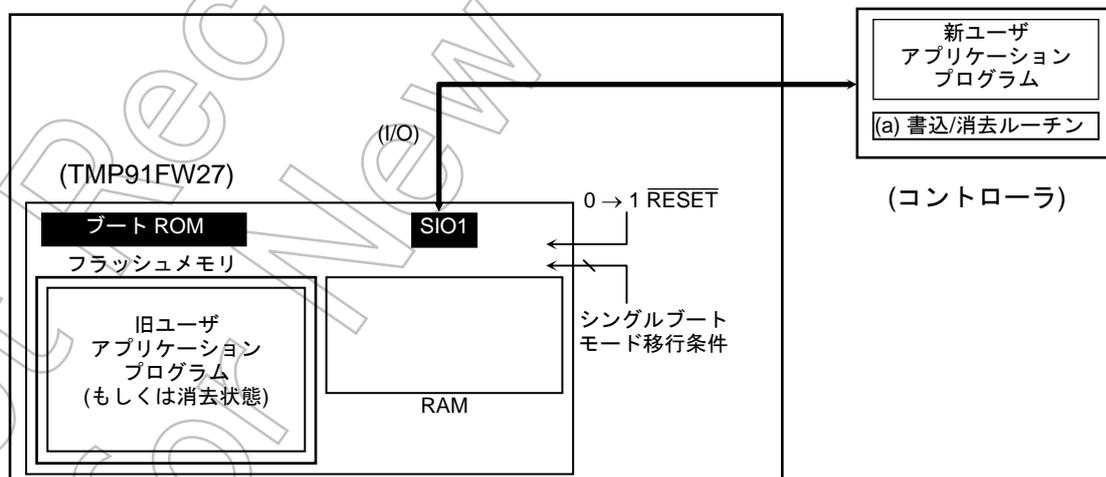
## (Step-1)環境準備

書き込み/消去ルーチン、書き替えデータなどの転送は SIO (SIO1) を経由して行いますので、ボード上で本デバイスの SIO (SIO1) と外部コントローラとをつなげます。書き込み/消去を行うための (a) 書き込み/消去ルーチンをコントローラ上に用意します。



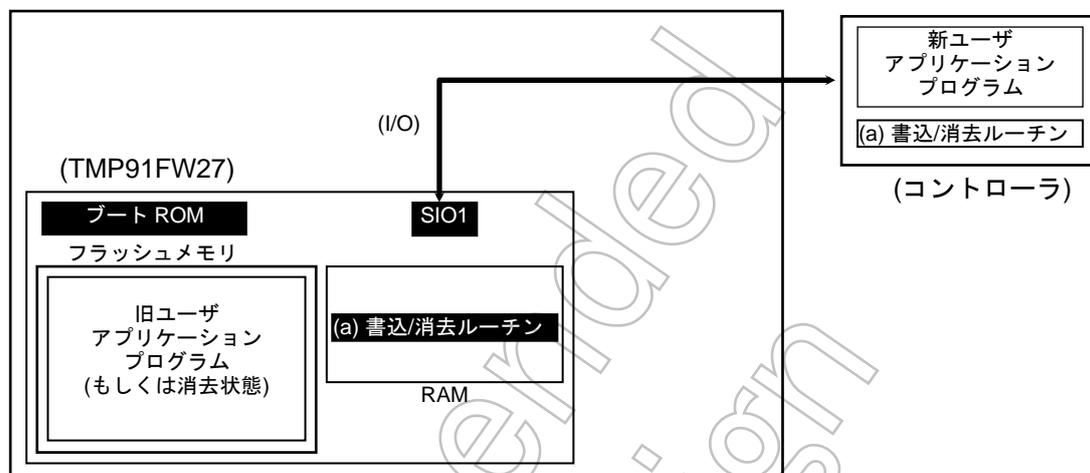
## (Step-2) ブートの起動(内蔵ブートROM起動)

ブートモードの端子条件設定でリセットを解除し、内蔵ブートROMを起動します。シングルブート動作の通信手順に従い、SIOを経由して転送元(コントローラ)より(a)書き込み/消去ルーチンの転送を行います。ここでは、ユーザアプリケーションプログラム上に記録されているパスワードとの照合を行います。(フラッシュメモリが消去されている状態でも、消去データ("0xFF"12バイト長)をパスワードとして照合を行います。)



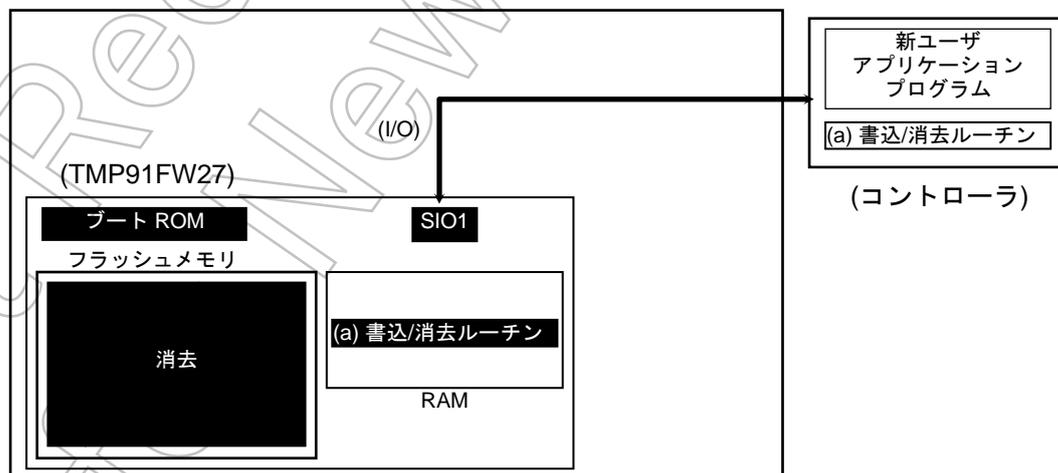
**(Step-3) RAM への書き替えルーチンのコピー**

パスワードの照合が終了すると、ブート ROM はシリアル通信を使用し、コントローラから(a)書き込み/消去ルーチンを内蔵 RAM へコピーします。ただし、RAM 上のアドレス 001000H~003DFH の範囲に格納してください。

**(Step-4) RAM からの書き替えルーチンの実行**

RAM 上の (a) 書き込み/消去ルーチンへ制御を移し、消去が必要な場合は、旧ユーザアプリケーションプログラムエリアの消去を行ってください(セクタ単位もしくはチップ消去)。

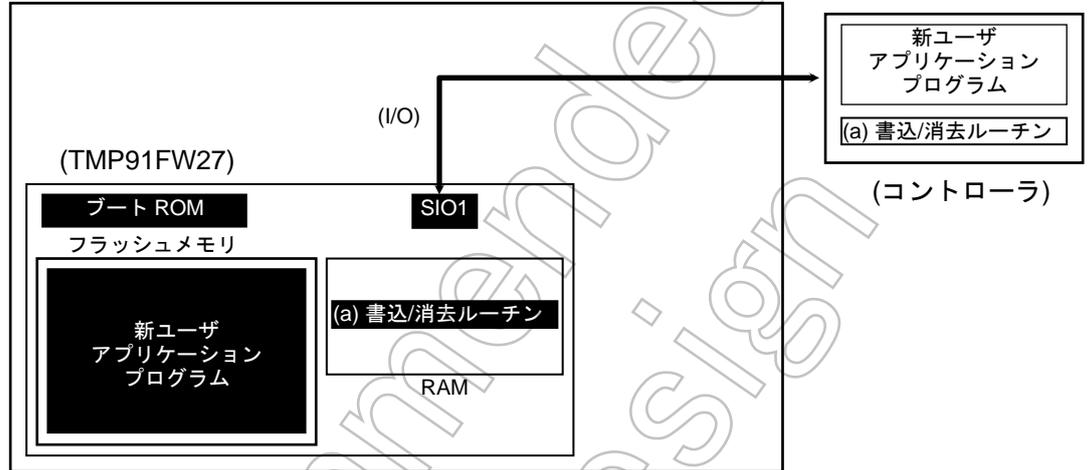
注) 内蔵ブートは、消去コマンドを持っているため、書き込み/消去ルーチンを用いずに、コントローラからチップ消去が可能です。セクタ消去を行う場合は、書き込み/消去ルーチン上に必要なプログラムを組み込んでください。



(Step-5) 新ユーザアプリケーションのコピー

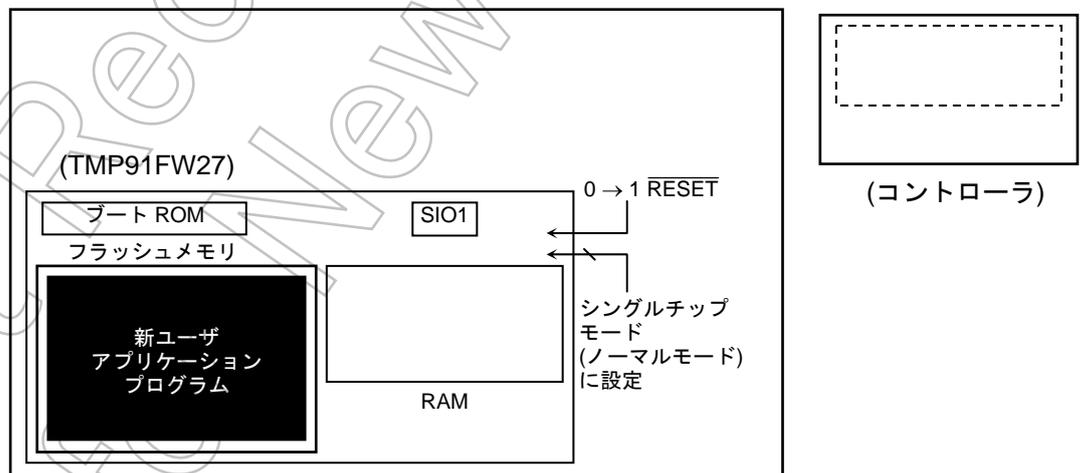
さらに、RAM 上の (a) 書き込み/消去ルーチンを実行して、転送元 (コントローラ) より新ユーザアプリケーションプログラムのデータをロードし、フラッシュメモリの消去したエリアに書き込みを行います。

下の例では、書き替えデータ転送時には、書き替え/消去ルーチンを転送したときと同様の通信設定を使用していますが、書き替え/消去ルーチンを転送後は、転送したルーチンを用いて、設定 (データバスおよび転送元) を変更できます。必要に応じて、ボードのハードおよび書き込み/消去ルーチンを組み立ててください。



(Step-6) 新ユーザアプリケーションプログラムの起動

書き込みが完了したら、一度ボードの電源を落とし、転送元 (コントローラ) と接続していたケーブルをはずします。その後、再度電源を入れ直し、シングルチップモード (ノーマルモード) に設定し、新しいユーザアプリケーションプログラムを実行します。



### 3.2.4.2 シングルブートモードでの接続例

シングルブートモードでは、シリアル転送によるフラッシュメモリの書き替えを行います。したがって、オンボードプログラミングは本デバイスのSIO (チャンネル 1)とコントローラ (書き込みツール) を接続し、コントローラ側からコマンドを送出することにより実行します。図 3.2.4に書き込みコントローラとターゲットボードの接続例を、図 3.2.5にRS232Cボードとターゲットボードの接続例を示します。

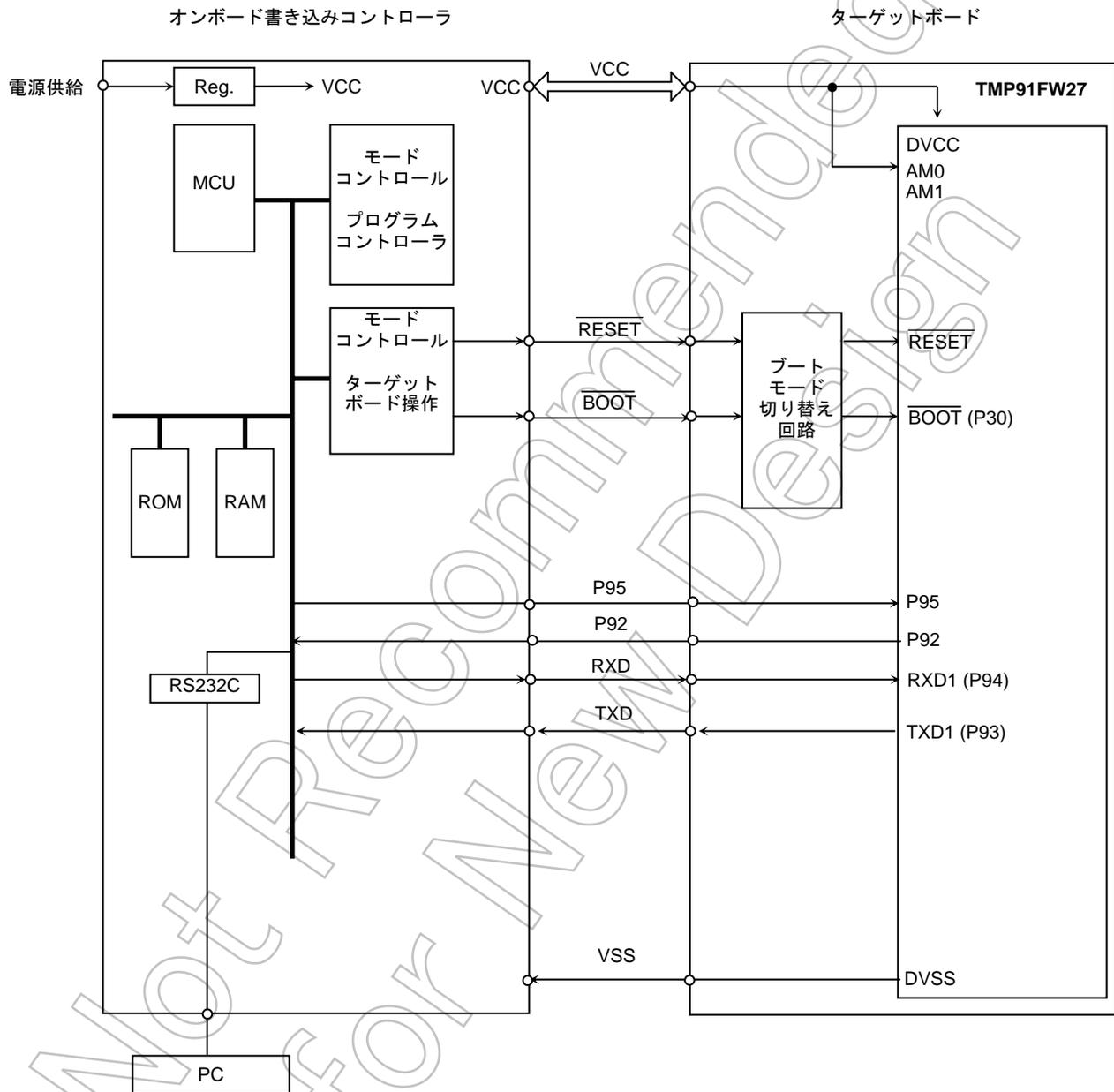


図 3.2.4 シングルブートモードでの外部コントローラとの接続例

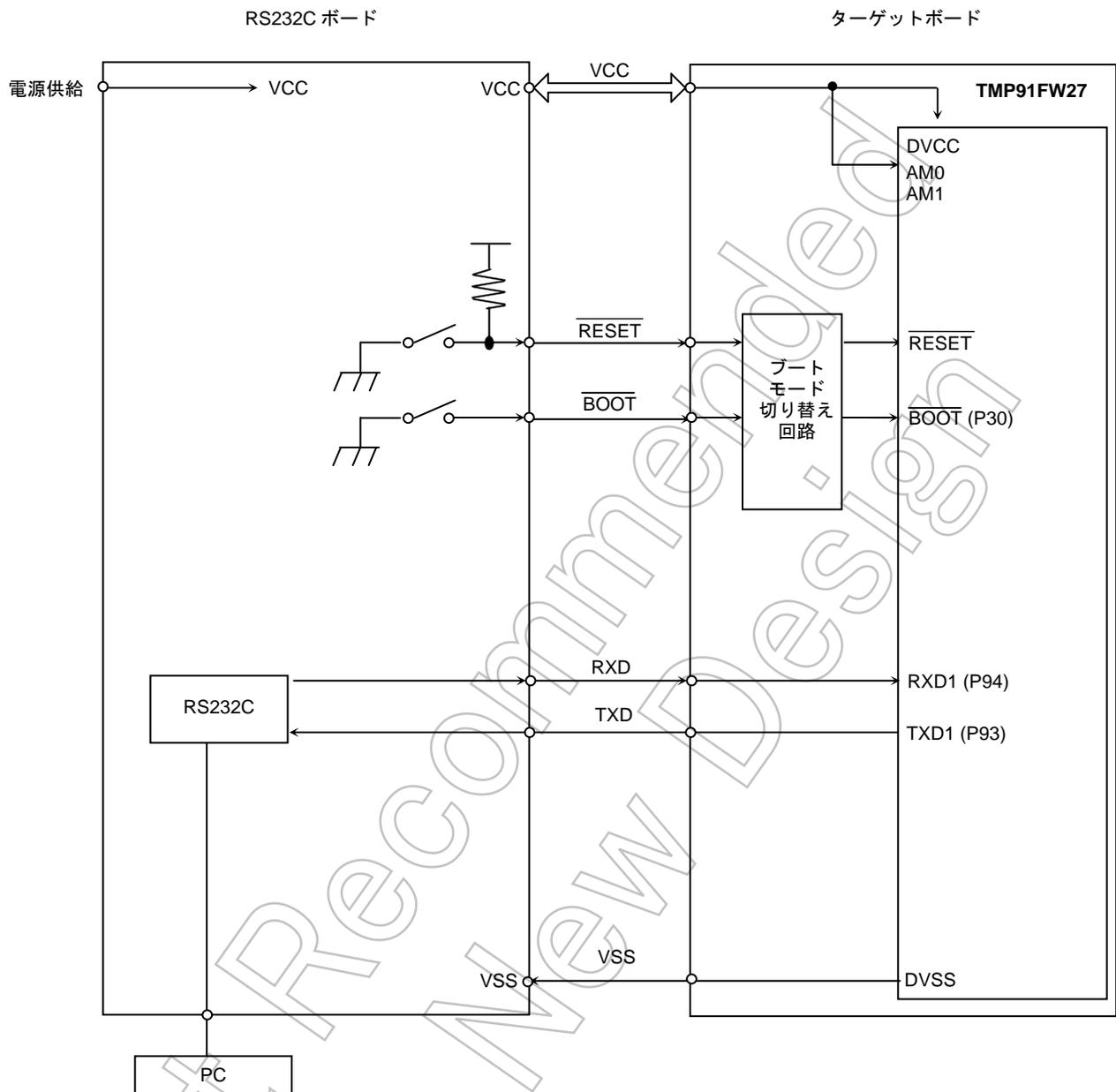


図 3.2.5 シングルブートモードでの RS232C ボードとの接続例

### 3.2.4.3 モード設定

オンボードプログラミングを実行するためには、本デバイスをシングルブートモードで立ち上げます。シングルブートモードで立ち上げるための設定を以下に示します。

- $\overline{AM0}, \overline{AM1} = 1$
- $\overline{BOOT} = 0$
- $\overline{RESET} = 0 \rightarrow 1$

$\overline{RESET}$  入力端子を“0”の状態にして、 $\overline{AM0}$ 、 $\overline{AM1}$ 、 $\overline{BOOT}$ の各端子をあらかじめ上記条件に設定します。その後、 $\overline{RESET}$  解除を行うとシングルブートモードで起動します。

### 3.2.4.4 メモリマップ

図 3.2.6 にノーマルモードとシングルブートモードのメモリマップの比較を示します。図 3.2.6 のように、シングルブートモードでは、内蔵フラッシュメモリは 10000H~2FFFFH 番地 (物理アドレス) にマッピングされます。また、FFF000H 番地から FFFFFFFH 番地にはブートROM (マスクROM) がマッピングされます。

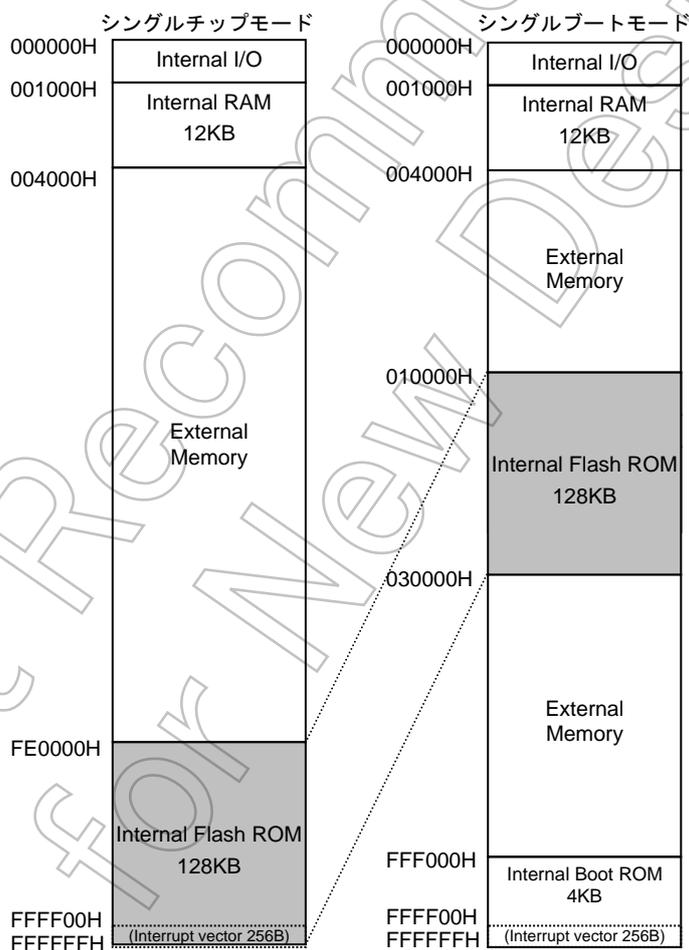


図 3.2.6 メモリマップの比較

### 3.2.4.5 インタフェース仕様

シングルブートモードでの SIO の通信フォーマットを以下に示します。シリアル動作のモードは、UART (非同期通信)に対応しています。

オンボードプログラミングを実行するためには、書き込みコントローラ側の通信フォーマットも同様に設定する必要があります。

#### ● UART(非同期)通信

- 通信チャンネル : SIOチャンネル1 (使用端子は表 3.2.4 参照)
- シリアル転送モード : UART (非同期通信) モード
- データ長 : 8 ビット
- パリティビット : なし
- STOP ビット : 1 ビット
- ボーレート : 表 3.2.5、表 3.2.6 参照

表 3.2.4 端子の接続

端 子		UART
電源系端子	DVCC	○
	DVSS	○
モード設定端子	AM1,AM0, BOOT	○
リセット端子	RESET	○
通信端子	TXD1	○
	RXD1	○

注) 未使用端子は、リセット解除後の初期状態になっています。

表 3.2.5 ボーレート表

SIO	転送レート(bps)				
UART	115200	57600	38400	19200	9600

表 3.2.6 シングルブートモード時の動作周波数とボーレート対応表

基準周波数 (MHz)	基準ボーレート (bps)		9600		19200		38400		57600		115200	
	対応範囲 (MHz)	ボーレート (bps)	誤差 (%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	
8	7.83~8.14	9615	+0.16	—	—	—	—	—	—	—	—	—
10	9.64~10.02	9766	+1.73	19531	+1.73	39063	+1.73	—	—	—	—	—
11.0592	10.84~11.28	9600	0	19200	0	—	—	—	—	—	—	—
12.2880	12.05~12.53	9600	0	19200	0	38400	0	—	—	—	—	—
14.7456	14.46~15.04	9600	0	19200	0	38400	0	57600	0	115200	0	0
16	15.66~16.29	9615	+0.16	19231	+0.16	—	—	—	—	—	—	—
18.4320	18.07~18.80	9600	0	19200	0	—	—	57600	0	—	—	—
20	19.27~20.05	9766	+1.73	19531	+1.73	39063	+1.73	—	—	—	—	—
22.1184	21.68~22.56	9600	0	19200	0	38400	0	57600	0	—	—	—
24.5760	24.09~25.06	9600	0	19200	0	38400	0	—	—	—	—	—
25		9766	+1.73	19531	+1.73	39063	+1.73	—	—	—	—	—
25.8048	25.29~26.32	9600	0	—	—	—	—	57600	0	—	—	—
27	26.50~27.57	9588	-0.13	19176	-0.13	38352	-0.13	—	—	—	—	—

基準周波数: シングルブートで対応可能な高速発振回路の周波数。

シングルブートモードを使ってフラッシュメモリの書き替えを行う場合は、高速クロックとして基準周波数のいずれかを選択してください。

対応範囲: 各基準周波数として検出されるクロック周波数の目安です。この範囲に含まれないクロック周波数では、シングルブート動作が行えない場合があります。

注) 基準周波数 (マイコンクロック周波数) の自動検出を行うために、フラッシュメモリ書き替えコントローラの送信ボーレートの誤差と、発振周波数誤差との総合誤差を、 $\pm 2\%$ 以内にしてください。

### 3.2.4.6 データ転送フォーマット

動作コマンド、および各動作モード時のデータ転送フォーマットをそれぞれ表 3.2.7～表 3.2.13に示します。

表 3.2.7 動作コマンドデータ

動作コマンドデータ	動作モード
10H	RAM 転送
20H	フラッシュメモリ SUM
30H	製品情報読み出し
40H	フラッシュメモリチップ消去
60H	フラッシュメモリプロテクト設定

Not Recommended for New Design

表 3.2.8 シングルブートプログラムの転送フォーマット [RAM 転送の場合]

	転送バイト数	コントローラ→本デバイスへの 転送データ	ボーレート	本デバイス→コントローラへの 転送データ	
BOOT ROM	1 バイト目	ボーレート設定 UART	86H	—	
	2 バイト目	—	所望の ボーレート (注 1)	ボーレート設定に対する ACK 応答 正常 (設定可能) の場合 ・UART 86H (ボーレートの設定が不可能と判断した 場合は動作停止)	
	3 バイト目	動作コマンドデータ (10H)		—	
	4 バイト目	—		動作コマンドに対する ACK 応答 (注 2) 正常の場合 10H 異常の場合 x1H プロテクト設定済みの場合(注 4) x6H 通信異常の場合 x8H	
	5 バイト目 ~ 16 バイト目	PASS WORD データ (12 バイト) (02FEF4H~02FEFFH)		—	
	17 バイト目	5 ~ 16 バイト目の CHECK SUM 値		—	
	18 バイト目	—		CHECK SUM 値に対する ACK 応答 (注 2) 正常の場合 10H 異常の場合 11H 通信異常の場合 18H	
	19 バイト目	RAM 格納開始アドレス 31 ~ 24 (注 3)		—	
	20 バイト目	RAM 格納開始アドレス 23 ~ 16 (注 3)		—	
	21 バイト目	RAM 格納開始アドレス 15 ~ 8 (注 3)		—	
	22 バイト目	RAM 格納開始アドレス 7 ~ 0 (注 3)		—	
	23 バイト目	RAM 格納バイト数 15 ~ 8 (注 3)		—	
	24 バイト目	RAM 格納バイト数 7 ~ 0 (注 3)		—	
	25 バイト目	19~24 バイト目の CHECK SUM 値(注 3)		—	
	26 バイト目	—		CHECK SUM 値に対する ACK 応答 (注 2) 正常の場合 10H 異常の場合 11H 通信異常の場合 18H	
	27 バイト目 ~ m バイト目	RAM 格納データ		—	
	m+1 バイト目	27 ~ m バイト目の CHECK SUM 値		—	
	m+2 バイト目	—		CHECK SUM 値に対する ACK 応答 (注 2) 正常の場合 10H 異常の場合 11H 通信異常の場合 18H	
	RAM	m+3 バイト目		—	JUMP RAM 格納開始アドレス

注 1) 所望のボーレートは表 3.2.6 を参照してください。

注 2) 異常応答後は、動作コマンド (3 バイト目) 待ちになります。

注 3) 19 バイト目~25 バイト目のデータは、RAM 上のアドレス 001000H~003DFFH (11.5KB)の領域内に納まるようにプログラムしてください。

注 4) リードプロテクトまたはライトプロテクトが設定されている場合は、受信したコマンドの動作を中止し、次の動作コマンド (3 バイト目) 待ちになります。

表 3.2.9 シングルブートプログラムの転送フォーマット [フラッシュメモリ SUM の場合]

	転送バイト数	コントローラ→本デバイスへの 転送データ	ボーレート	本デバイス→コントローラへの 転送データ
BOOT ROM	1 バイト目	ボーレート設定 UART	86H	—
	2 バイト目	—		ボーレート設定に対する ACK 応答 正常 (設定可能) の場合 ・UART 86H (ボーレートの設定が不可能と判断した 場合は動作停止)
	3 バイト目	動作コマンドデータ (20H)	—	—
	4 バイト目	—		動作コマンドに対する ACK 応答 (注 2) 正常の場合 20H 異常の場合 x1H 通信異常の場合 x8H
	5 バイト目	—		SUM (上位)
	6 バイト目	—		SUM (下位)
	7 バイト目	—		5 ~ 6 バイト目の CHECK-SUM 値
	8 バイト目	(次の動作コマンドデータ待ち)		—

- 注 1) 所望のボーレートは表 3.2.6 を参照してください。  
 注 2) 異常応答後は、動作コマンド (3 バイト目) 待ちになります。

表 3.2.10 シングルブートプログラムの転送フォーマット [製品情報読み出しの場合] (1/2)

	転送バイト数	コントローラ→本デバイスへの 転送データ	ボーレート	本デバイス→コントローラへの 転送データ
BOOT ROM	1 バイト目	ボーレート設定 UART	86H	—
	2 バイト目	—	所望の ボーレート (注 1)	ボーレート設定に対する ACK 応答 正常 (設定可能) の場合 ・UART 86H (ボーレートの設定が不可能と判断した 場合は、動作停止)
	3 バイト目	動作コマンドデータ (30H)		—
	4 バイト目	—		動作コマンドに対する ACK 応答 (注 2) 正常の場合 30H 異常の場合 x1H 通信異常の場合 x8H
	5 バイト目	—		フラッシュメモリデータ (02FEF0H 番地)
	6 バイト目	—		フラッシュメモリデータ (02FEF1H 番地)
	7 バイト目	—		フラッシュメモリデータ (02FEF2H 番地)
	8 バイト目	—		フラッシュメモリデータ (02FEF3H 番地)
	9 バイト目 ~ 20 バイト目	—		製品名 (アスキーコード、12 バイト) 9 バイト目から 'TMP91FW27_ _ _ '
	21 バイト目 ~ 24 バイト目	—		Password 比較開始アドレス (4 バイト) 21 バイト目から F4H, FEH, 02H, 00H
	25 バイト目 ~ 28 バイト目	—		RAM 開始アドレス (4 バイト) 25 バイト目から 00H, 10H, 00H, 00H
	29 バイト目 ~ 32 バイト目	—		RAM(ユーザ領域)終了アドレス (4 バイト) 29 バイト目から FFH, 3DH, 00H, 00H
	33 バイト目 ~ 36 バイト目	—		RAM 終了アドレス (4 バイト) 33 バイト目から FFH, 3FH, 00H, 00H
	37 バイト目 ~ 40 バイト目	—		ダミーデータ (4 バイト) 37 バイト目から 00H, 00H, 00H, 00H
	41 バイト目 ~ 44 バイト目	—		ダミーデータ (4 バイト) 41 バイト目から 00H, 00H, 00H, 00H
	45 バイト目 ~ 46 バイト目	—		FUSE 情報 (2 バイト) 45 バイト目から リード/ライトプロテクト設定 1) あり / あり : 00H, 00H 2) なし / あり : 01H, 00H 3) あり / なし : 02H, 00H 4) なし / なし : 03H, 00H
	47 バイト目 ~ 50 バイト目	—		フラッシュメモリ開始アドレス (4 バイト) 47 バイト目から 00H, 00H, 01H, 00H
	51 バイト目 ~ 54 バイト目	—		フラッシュメモリ終了アドレス (4 バイト) 51 バイト目から FFH, FFH, 02H, 00H
	55 バイト目 ~ 56 バイト目	—		フラッシュメモリセクタ分割数情報 (2 バイト) 55 バイト目から 20H, 00H
	57 バイト目 ~ 60 バイト目	—		フラッシュメモリ同一セクタサイズの 開始アドレス (4 バイト) 57 バイト目から 00H, 00H, 01H, 00H

表 3.2.11 シングルブートプログラムの転送フォーマット [製品情報読み出しの場合] (2/2)

	転送バイト数	コントローラ→本デバイスへの 転送データ	ポーレート	本デバイス→コントローラへの 転送データ
BOOT ROM	61 バイト目 ~ 64 バイト目	—		フラッシュメモリ同一セクタサイズの サイズ (ハーフワード) (4 バイト) 61 バイト目から 00H, 08H, 00H, 00H
	65 バイト目	—		フラッシュメモリ同一セクタサイズの 個数 (1 バイト) 20H
	66 バイト目	—		5 ~ 65 バイト目の CHECK SUM 値
	67 バイト目	(次の動作コマンドデータ待ち)		—

- 注 1) 所望のポーレートは表 3.2.6 を参照してください。  
 注 2) 異常応答後は、動作コマンド (3 バイト目) 待ちになります。

Not Recommended  
for New Design

表 3.2.12 シングルブートプログラムの転送フォーマット [フラッシュメモリチップ消去の場合]

	転送バイト数	コントローラ→本デバイスへの 転送データ	ボーレート	本デバイス→コントローラへの 転送データ
BOOT ROM	1 バイト目	ボーレート設定 UART	86H 所望の ボーレート (注 1)	—
	2 バイト目	—		ボーレート設定に対する ACK 応答 正常 (設定可能) の場合 ・UART 86H (ボーレートの設定が不可能と判断した 場合は動作停止)
	3 バイト目	動作コマンドデータ (40H)		—
	4 バイト目	—		動作コマンドに対する ACK 応答 (注 2) 正常の場合 40H 異常の場合 x1H 通信異常の場合 x8H
	5 バイト目	消去イネーブルコマンドデータ (54H)		—
	6 バイト目	—		動作コマンドに対する ACK 応答 (注 2) 正常の場合 54H 異常の場合 x1H 通信異常の場合 x8H
	7 バイト目	—		消去コマンドに対する ACK 応答 正常の場合 4FH 異常の場合 4CH
	8 バイト目	—		ACK 応答 正常の場合 5DH 異常の場合 60H
	9 バイト目	(次の動作コマンドデータ待ち)		—

注 1) 所望のボーレートは表 3.2.6 を参照してください。  
 注 2) 異常応答後は、動作コマンド (3 バイト目) 待ちになります。

表 3.2.13 シングルブートプログラムの転送フォーマット [フラッシュメモリプロテクト設定の場合]

	転送バイト数	コントローラ→本デバイスへの 転送データ	ボーレート	本デバイス→コントローラへの 転送データ
BOOT ROM	1 バイト目	ボーレート設定 UART	86H 所望の ボーレート (注 1)	—
	2 バイト目	—		ボーレート設定に対する ACK 応答 正常 (設定可能) の場合 ・UART 86H (ボーレートの設定が不可能と判断した 場合は動作停止)
	3 バイト目	動作コマンドデータ (60H)		—
	4 バイト目	—		動作コマンドに対する ACK 応答 (注 2) 正常の場合 60H 異常の場合 x1H 通信異常の場合 x8H
	5 バイト目 ~ 16 バイト目	PASS WORD データ (12 バイト) (02FEF4H~02FEFFH)		—
	17 バイト目	5 ~ 16 バイト目の CHECK SUM 値		—
	18 バイト目	—		CHECK SUM 値に対する ACK 応答 (注 2) 正常の場合 60H 異常の場合 61H 通信異常の場合 68H
	19 バイト目	—		プロテクト設定コマンドに対する ACK 応答 正常の場合 6FH 異常の場合 6CH
	20 バイト目	—		ACK 応答 正常の場合 31H 異常の場合 34H
	21 バイト目	(次の動作コマンドデータ待ち)		—

注 1) 所望のボーレートは表 3.2.6 を参照してください。

注 2) 異常応答後は、動作コマンド (3 バイト目) 待ちになります。

### 3.2.4.7 ブートプログラム

シングルブートモード立ち上げ時にはブートプログラムが起動します。

ここではシングルブートモード起動時の内蔵ブートプログラムがコントローラと通信を行うのに必要なタイミングを述べます。ユーザがシングルブートを使用するコントローラを作成する場合や、ユーザがユーザブートの環境を構築する為の情報として活用ください。

#### 1. RAM 転送コマンド

RAM 転送は、コントローラから送られてくるデータを内蔵 RAM へ格納します。転送が正常に終了するとユーザプログラムの実行を開始します。ユーザプログラムのサイズは、最大 11.5K バイトまで転送可能です(スタックポインタエリア等の保護のため、ブートプログラム上で制限をかけています)。また、実行開始アドレスは、RAM 格納開始アドレスになります。

このRAM転送機能でユーザ作成の書き込み/消去プログラムを実行することにより、ユーザ独自のオンボードプログラミング制御を行うことができます。ユーザプログラムでオンボードプログラミングを実行するためには、3.2.6「フラッシュメモリコマンドシーケンス」を使う必要があります。RAM転送コマンドが終了した後は、内蔵RAMの全領域を使用することができます。

なお、デバイスにリードプロテクトまたはライトプロテクトが設定されている場合は、本コマンドは実行されません。また、パスワードエラーが発生した場合も、本コマンドは実行されません。

#### 2. フラッシュメモリ SUM コマンド

フラッシュメモリ 128K バイトの SUM を計算しその結果を返します。ブートプログラムではフラッシュメモリの全エリアのデータを読み出す動作コマンドはサポートしていません。その代わりに、このフラッシュメモリ SUM コマンドがあります。SUM を読み出すことで、アプリケーションプログラムのレビジョンを管理することができます。

#### 3. 製品情報読み出しコマンド

本デバイスの製品名やメモリ情報などを返します。製品情報読み出しコマンドでは、フラッシュメモリの一部エリア (02FEF0H~02FEF3H) のデータを返します。フラッシュメモリ SUM コマンド以外に、このデータを用いることでも、アプリケーションプログラムのレビジョンを管理することができます。

#### 4. フラッシュメモリチップ消去コマンド

すべてのセクタのフラッシュメモリを消去します。リード/ライトプロテクトがかかっている場合、メモリセルの全てのセクタを消去し、リード/ライトプロテクトを解除します。

本コマンドは、パスワードを忘れた場合のブートプログラムの操作を回復する機能も兼用しているため、パスワード比較は行っていません。

5. フラッシュメモリプロテクト設定コマンド

リードプロテクトとライトプロテクトを同時に設定します。但し、パスワードエラーが発生した場合は、本コマンドは実行されません。

リードプロテクトを設定するとライターモードの時にフラッシュメモリのリードができなくなります。また、ライトプロテクトを設定するとライターモードの時にフラッシュメモリのライトが出来なくなります。

Not Recommended  
for New Design

### 3.2.4.8 RAM転送コマンド (表 3.2.8参照)

#### 1. コントローラ→デバイス

1 バイト目のデータは、ボーレートを判定するデータになります。1 バイト目のデータは、受信を禁止した状態 ( $SC1MOD0<RXE> = 0$ ) にしています。(ここでは、内部タイマでボーレート判定をおこなっています。)

- UARTで通信

コントローラからターゲットボードへは、UART の設定で、所望のボーレートでデータを 86H にして送信してください。シリアル動作モードの判定で UART と判定した場合、ボーレートの設定が可能かどうかを判定します。ボーレートが適切では無く、設定が不可能と判断した場合は動作を停止するため、通信が行えなくなります。

#### 2. デバイス→コントローラ

2 バイト目の送信データは、1 バイト目のシリアル動作モード設定データに対する ACK 応答データになります。1 バイト目のデータが、UART と判定されボーレートの設定が可能な場合 86H を送信します。

- ボーレート判定

ボーレートの設定が可能かどうかを判定します。設定が可能と判定した場合、内蔵ブートプログラムが、BR1CR、BR1ADD の値を書き替え、86H を送信し、設定が不可能と判定した場合、動作を停止するため何も送信しません。コントローラは、1 バイト目のデータの送信が終了した後、タイムアウト時間 (5 秒) を設けます。タイムアウト時間内に、データ (86H) を正常受信できなければ、通信不能と判断してください。受信を許可 ( $SC1MOD0<RXE> = 1$ ) するタイミングは、送信バッファにデータ (86H) を書き込む前に行っています。

#### 3. コントローラ→デバイス

3 バイト目の受信データは、動作コマンドデータになります。この場合は、RAM 転送コマンドデータ (10H) になります。

#### 4. デバイス→コントローラ

4 バイト目の送信データは、3 バイト目の動作コマンドデータに対する ACK 応答データになります。最初に、3 バイト目の受信データに受信エラーがあるかをチェックします。受信エラーがある場合、通信異常の ACK 応答データ (bit 3) x8H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。送信データの上位 4 ビットは、不定値になります(直前の動作コマンドデータの上位 4 ビットになります)。

次に、3 バイト目の受信データが、表 3.2.7 の動作コマンドデータのいずれかに該当する場合は、受信データをエコーバック送信 (正常ACK応答データ) します。RAM転送コマンドを選択した場合には、プロテクト(リードまたはライト)が設定されていないことをチェックし、10Hをエコーバック送信してRAM転送処理ルーチンに分岐します。プロテクトが設定されている場合は、ACK応答データ (bit 2/1) x6H を送信して、再び動作コマンド (3 バイト目) データ待ちになります。送信データの上位 4 ビットは、不定値になります(直前の動作コマンドデータの上位 4 ビットになります)。

RAM転送処理ルーチンに分岐後、パスワードエリアのデータをチェックします。パスワードエリアのデータのチェック方法は、3.2.4.15 「パスワードについて」を参照してください。

3 バイト目の受信データがいずれのコマンドにも該当しない場合は、動作コマンドエラーの ACK 応答データ (bit 0) x1H を送信して、再び動作コマンド (3 バイト目) データ待ちになります。送信データの上位 4 ビットは、不定値になります(直前の動作コマンドデータの上位 4 ビットになります)。

## 5. コントローラー→デバイス

5 バイト目～16 バイト目の受信データは、パスワードデータ (12 バイト) になります。5 バイト目の受信データはフラッシュメモリの 02FEF4H のデータと照合し、6 バイト目の受信データはフラッシュメモリの 02FEF5H のデータと照合します。同様に 16 バイト目の受信データはフラッシュメモリの 02FEFFH のデータと照合します。

## 6. コントローラー→デバイス

17 バイト目の受信データは、CHECK SUMデータになります。5 バイト目から 16 バイト目の送信データを符号なしの 8 ビット加算 (オーバーフローを無視) して得られた下位 8 ビット値の 2 の補数をコントローラから送信してください。CHECK SUMデータの計算方法は、3.2.4.17 「CHECK SUMの計算方法」を参照してください。

## 7. デバイス→コントローラ

18 バイト目の送信データは、5 バイト目～17 バイト目のデータに対する ACK 応答データ (CHECK SUM 値に対する ACK 応答) になります。最初に、5 バイト目～17 バイト目の受信データに受信エラーがあるかをチェックします。受信エラーがある場合、通信異常の ACK 応答データ (bit 3) 18H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。送信データの上位 4 ビットは、直前の動作コマンドデータの上位 4 ビットになるので、“1” になります。

次に、17 バイト目の CHECK SUM データをチェックします。CHECK SUM データのチェック方法は、5 バイト目～17 バイト目までの受信データを符号なし 8 ビット加算 (オーバーフローを無視) して得られた値の下位 8 ビットが、00H かどうかをチェックしています。00H 以外の場合、CHECK SUM エラーの ACK 応答データ (bit 0) 11H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。

最後に、パスワードの照合結果をチェックします。5 バイト目～16 バイト目のパスワードデータの照合がすべて一致しない場合、パスワードエラーの ACK 応答データ (bit 0) 11H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。

上記のチェックを終えて、すべて正常なら、正常 ACK 応答データ 10H を送信します。

## 8. コントローラー→デバイス

19 バイト目～22 バイト目までの受信データは、ブロック転送における格納先の RAM の開始アドレスを表します。19 バイト目がアドレスの 31 ビット～24 ビットに対応、20 バイト目がアドレスの 23 ビット～16 ビットに対応、21 バイト目が 15 ビット～8 ビット、22 バイト目が 7 ビット～0 ビットに対応します。

## 9. コントローラー→デバイス

23 バイト目、24 バイト目の受信データは、ブロック転送するバイト数を表します。23 バイト目が転送バイト数の 15 ビット～8 ビット目に対応し、24 バイト目が 7 ビット～0 ビット目に対応します。

## 10. コントローラー→デバイス

25 バイト目の受信データは、CHECK SUMデータになります。19 バイト目から 24 バイト目の送信データを符号なし 8 ビット加算 (オーバーフローを無視) して得られた下位 8 ビット値の 2 の補数値をコントローラから送信してください。CHECK SUMデータ計算方法は、3.2.4.17 「CHECK SUMの計算方法」を参照してください。

注) 19 バイト目～25 バイト目のデータは RAM 上のアドレス 001000H～003DFFH (11.5KB) の領域に納まるようにプログラムしてください。

## 11. デバイス→コントローラ

26 バイト目の送信データは、19 バイト目～25 バイト目のデータに対する ACK 応答データ (CHECK SUM 値に対する ACK 応答) になります。

最初に、19 バイト目～25 バイト目の受信データに受信エラーがあるかどうかをチェックします。受信エラーがある場合、通信異常の ACK 応答データ (bit 3) 18H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。送信データの上位 4 ビットは、直前の動作コマンドデータの上位 4 ビットになるので“1”になります。

次に、25 バイト目の CHECK SUM データをチェックします。CHECK SUM データのチェック方法は、19 バイト目～25 バイト目までの受信データを符号なし 8 ビット加算 (オーバフローを無視) して得られた値の下位 8 ビットが、00H かどうかをチェックしています。00H 以外の場合、CHECK SUM エラーの ACK 応答データ (bit 0) 11H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。

上記のチェックを終えてすべて正常なら、正常 ACK 応答データ 10H を送信します。

## 12. コントローラ→デバイス

27 バイト目～ $m$  バイト目の受信データは、RAM へ格納するデータになります。RAM に格納するデータを、19 バイト目から 22 バイト目で指定されたアドレスから書き込み、23 バイト目から 24 バイト目に指定されたバイト数分だけ書き込みます。

## 13. コントローラ→デバイス

$m + 1$  バイト目の受信データは、CHECK SUM データになります。27 バイト目～ $m$  バイト目の送信データを符号なし 8 ビット加算 (オーバフローを無視) して得られた下位 8 ビット値の 2 の補数をコントローラから送信してください。CHECK SUM データの計算方法は、3.2.4.17 「CHECK SUM の計算方法」を参照してください。

## 14. デバイス→コントローラ

$m + 2$  バイト目の送信データは、27 バイト目～( $m + 1$ ) バイト目のデータに対する ACK 応答データ (CHECK SUM に対する ACK 応答) になります。

最初に 27 バイト目～ $m + 1$  バイト目の受信データに受信エラーがあるかどうかをチェックします。受信エラーがある場合、通信異常の ACK 応答データ (bit 3) 18H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。送信データの上位 4 ビットは、直前の動作コマンドデータの上位 4 ビットになるので“1”になります。

次に、 $m + 1$  バイト目の CHECK SUM データをチェックします。CHECK SUM データのチェック方法は、27 バイト目～ $m + 1$  バイト目までの受信データを符号なし 8 ビット加算 (オーバフローを無視) して得られた値の下位 8 ビットが、00H かどうかをチェックしています。00H 以外の場合、CHECK SUM エラーの ACK 応答データ (bit 0) 11H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。

上記のチェックを終えてすべて正常なら、正常 ACK 応答データ 10H を送信します。

## 15. デバイス→コントローラ

$m + 2$  バイト目の ACK 応答データが正常 ACK 応答データの場合、正常 ACK 応答データ 10H を送信後、19 バイト目～22 バイト目で指定された RAM のスタートアドレスに分岐します。

## 3.2.4.9 フラッシュメモリSUMコマンド (表 3.2.9参照)

1. 1バイト目~2バイト目までの送受信データはRAM転送コマンドの場合と同一になります。
2. コントローラ → デバイス  
3バイト目の受信データは動作コマンドデータになります。この場合は、フラッシュメモリ SUM コマンドデータ (20 H) になります。
3. デバイス → コントローラ  
4バイト目の送信データは、3バイト目の動作コマンドデータに対する ACK 応答データになります。  
最初に、3バイト目の受信データに受信エラーがあるかをチェックします。受信エラーがある場合、通信異常のACK応答データ (bit 3) x8Hを送信して、次の動作コマンド(3バイト目)データ待ちになります。送信データの上位4ビットは不定値になります(直前の動作コマンドデータの上位4ビットになります)。  
次に、3バイト目の受信データが、表 3.2.7の動作コマンドデータのいずれかに該当する場合は、受信データをエコーバック送信(正常ACK応答データ)します。この場合、20Hをエコーバック送信して、フラッシュメモリSUM処理ルーチンに分岐します。該当しない場合は、動作コマンドエラーのACK応答データ (bit 0) x1Hを送信して、次の動作コマンド(3バイト目)データ待ちになります。送信データ上位4ビットは不定値になります(直前の動作コマンドデータの上位4ビットになります)。
4. デバイス → コントローラ  
5バイト目の送信データはSUM値の上位データ、6バイト目の送信データはSUM値の下位データになります。SUMの計算方法は、3.2.4.16「SUMの計算方法」を参照してください。
5. デバイス → コントローラ  
7バイト目の送信データは、CHECK SUM データになります。5バイト目から6バイト目の送信データを符号なし8ビット加算(オーバフローを無視)を行い、得られた下位8ビット値の2の補数を送信します。
6. コントローラ → デバイス  
8バイト目の受信データは、次の動作コマンドデータになります。

## 3.2.4.10 製品情報読み出しコマンド (表 3.2.10、表 3.2.11参照)

1. 1バイト目~2バイト目までの送受信データはRAM転送コマンドの場合と同一になります。
2. コントローラ → デバイス  
3 バイト目の受信データは、動作コマンドデータになります。この場合は、製品情報読み出しコマンドデータ (30H) になります。
3. デバイス → コントローラ  
4 バイト目の送信データは、3 バイト目の動作コマンドデータに対する ACK 応答データになります。  
最初に、3 バイト目の受信データに受信エラーがあるかどうかをチェックします。受信エラーがある場合、通信異常の ACK 応答データ (bit 3) x8H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。送信データの上位 4 ビットは不定値になります(直前の動作コマンドデータの上位 4 ビットになります)。  
次に、3 バイト目の受信データが、表 3.2.7の動作コマンドデータのいずれかに該当する場合は、受信データをエコーバック送信 (正常ACK応答データ) します。この場合、30Hをエコーバック送信して、製品情報読み出し処理ルーチンに分岐します。該当しない場合は、動作コマンドエラーのACK応答データ (bit 0) x1Hを送信して、次の動作コマンド (3 バイト目) データ待ちになります。送信データの上位 4 ビットは不定値になります(直前の動作コマンドデータの上位 4 ビットになります)。
4. デバイス → コントローラ  
5 バイト目 ~ 8 バイト目の送信データは、フラッシュメモリのデータ (02FEF0H~02FEF3H 番地のデータ) になります。この番地にソフトなどの ID 情報を書き込んでおくことにより、書き込んだソフトのバージョン管理をすることができます。  
(例えば 0002H なら、書き込んだソフトがバージョン 2 と管理する。)
5. デバイス → コントローラ  
9 バイト目~20 バイト目の送信データは製品名になります。9 バイト目から、アスキーコードで、'TMP91FW27\_\_\_'を送信します。  
注) ' ': スペース
6. デバイス → コントローラ  
21 バイト目 ~ 24 バイト目の送信データはパスワード比較開始アドレスになります。  
21 バイト目から、F4H、FEH、02H、00H を送信します。
7. デバイス → コントローラ  
25 バイト目~28 バイト目の送信データは RAM 開始アドレスになります。25 バイト目から、00H、10H、00H、00H を送信します。
8. デバイス → コントローラ  
29 バイト目 ~ 32 バイト目の送信データは RAM(ユーザ領域)終了アドレスになります。  
29 バイト目から、FFH、3DH、00H、00H を送信します。
9. デバイス → コントローラ  
33 バイト目 ~ 36 バイト目の送信データは RAM 終了アドレスになります。33 バイト目から、FFH、3FH、00H、00H を送信します。

## 10. デバイス → コントローラ

37 バイト目 ~ 44 バイト目の送信データは、ダミーデータになります。

## 11. デバイス → コントローラ

45 バイト目 ~ 46 バイト目の送信データは、プロテクトの設定状態や、フラッシュメモリがセクタ分割されているかを各ビットに割り付けて送信します。

## ●0 ビット目は、リードプロテクトの設定状態を示します。

- ・“0”はリードプロテクトが設定されている
- ・“1”はリードプロテクトが設定されていない

## ●1 ビット目は、ライトプロテクトの設定状態を示します。

- ・“0”はライトプロテクトが設定されている
- ・“1”はライトプロテクトが設定されていない

## ●2 ビット目は、フラッシュメモリが分割されているかを示します。

- ・“0”は分割されている
- ・“1”は分割されていない

## ●3 ビット目から 15 ビット目までは“0”を送信します。

## 12. デバイス → コントローラ

47 バイト目 ~ 50 バイト目の送信データは、フラッシュメモリ開始アドレスになります。47 バイト目から、00H、00H、01H、00H を送信します。

## 13. デバイス → コントローラ

51 バイト目 ~ 54 バイト目の送信データは、フラッシュメモリ終了アドレスになります。51 バイト目から、FFH、FFH、02H、00H を送信します。

## 14. デバイス → コントローラ

55 バイト目 ~ 56 バイト目の送信データは、フラッシュメモリのセクタ分割数になります。55 バイト目から、20H、00H を送信します。

## 15. デバイス → コントローラ

57 バイト目 ~ 65 バイト目の送信データは、フラッシュメモリのセクタ情報になります。セクタ情報は、フラッシュメモリ開始アドレスから見たとき、同一セクタサイズが何セクタ続いているかを一単位とし、同一セクタサイズの先頭の開始アドレスとセクタサイズ(ハーフワード)およびセクタの個数で表します。なお、セクタサイズはワードで表します。

57 バイト目 ~ 65 バイト目の送信データは、4K バイトのセクタ (Sector0~Sector31) を表します。

送信データについては、表 3.2.10、表 3.2.11を参照してください。

## 16. デバイス → コントローラ

66 バイト目の送信データは、CHECK SUM データになります。5 バイト目から 65 バイト目の送信データを符号なし 8 ビット加算 (オーバーフローを無視) して、得られた下位 8 ビット値の 2 の補数を送信します。

## 17. コントローラ → デバイス

67 バイト目の受信データは、次の動作コマンドデータになります。

## 3.2.4.11 フラッシュメモリチップ消去コマンド (表 3.2.12参照)

1. 1バイト目~2バイト目までの送受信データはRAM転送コマンドの場合と同一になります。
2. コントローラ → デバイス  
3 バイト目の受信データは動作コマンドデータになります。この場合は、フラッシュメモリチップ消去コマンドデータ (40 H) になります。
3. デバイス → コントローラ  
4 バイト目の送信データは、3 バイト目の動作コマンドデータに対する ACK 応答データになります。最初に、3 バイト目の受信データに受信エラーがあるかどうかをチェックします。受信エラーがある場合、通信異常の ACK 応答データ (bit 3) x8H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。送信データの上位 4 ビットは不定値になります(直前の動作コマンドデータの上位 4 ビットになります)。次に、3 バイト目の受信データが、表 3.2.7 の動作コマンドデータのいずれかに該当する場合は、受信データをエコーバック送信 (正常ACK応答データ) します。この場合、40H をエコーバック送信します。該当しない場合は、動作コマンドエラーの ACK 応答データ (bit 0) x1H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。送信データの上位 4 ビットは不定値になります(直前の動作コマンドデータの上位 4 ビットになります)。
4. コントローラ → デバイス  
5 バイト目の受信データは消去イネーブルコマンドデータ(54H)になります。
5. デバイス → コントローラ  
6 バイト目の送信データは、5 バイト目の消去イネーブルコマンドデータに対する ACK 応答データになります。  
最初に、5 バイト目の受信データに受信エラーがあるかどうかをチェックします。受信エラーがある場合、通信異常の ACK 応答データ (bit 3) x8H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。送信データの上位 4 ビットは不定値になります(直前の動作コマンドデータの上位 4 ビットになります)。  
次に、5 バイト目の受信データが、消去イネーブルコマンドデータに該当する場合は、受信データをエコーバック送信 (正常 ACK 応答データ) します。この場合、54H をエコーバック送信して、フラッシュメモリチップ消去処理ルーチンに分岐します。該当しない場合は、動作コマンドエラーの ACK 応答データ (bit 0) x1H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。送信データの上位 4 ビットは不定値になります(直前の動作コマンドデータの上位 4 ビットになります)。
6. デバイス → コントローラ  
7 バイト目の送信データは消去が正常に終了したかどうかを示します。  
正常に終了した時は、終了コード(4FH)を返します。  
消去 Error が起きた場合は、エラーコード(4CH)を返します。
7. デバイス → コントローラ  
8 バイト目の送信データは ACK 応答データです。  
正常に終了した時は、消去 ACK 応答コード(5DH)を返します。  
消去 Error の場合は、消去 ErrorACK 応答コード (60H)を返します。
8. コントローラ → デバイス  
9 バイト目の受信データは、次の動作コマンドデータになります。

## 3.2.4.12 フラッシュメモリプロテクト設定コマンド (表 3.2.13参照)

1. 1バイト目~2バイト目までの送受信データはRAM転送コマンドの場合と同一になります。
2. コントローラ → デバイス  
3 バイト目の受信データは、動作コマンドデータになります。この場合は、フラッシュメモリプロテクト設定コマンドデータ (60H) になります。
3. デバイス → コントローラ  
4バイト目の送信データは、3バイト目の動作コマンドデータに対する ACK 応答データになります。最初に、3 バイト目の受信データに受信エラーがあるかをチェックします。受信エラーがある場合、通信異常の ACK 応答データ (bit 3) x8H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。送信データの上位4ビットは、不定値になります(直前の動作コマンドデータの上位4ビットになります)。  
次に、3バイト目の受信データが、表 3.2.7の動作コマンドデータのいずれかに該当する場合は、受信データをエコーバック送信 (正常ACK応答データ) します。フラッシュメモリプロテクト設定コマンドを選択した場合には、60Hをエコーバック送信してフラッシュメモリプロテクト設定処理ルーチンに分岐します。  
このルーチンに分岐後、パスワードエリアのデータをチェックします。パスワードエリアのデータのチェック方法は、3.2.4.15「パスワードについて」を参照してください。  
3 バイト目の受信データがいずれのコマンドにも該当しない場合は、動作コマンドエラーの ACK 応答データ (bit 0) x1H を送信して、再び動作コマンド (3 バイト目) データ待ちになります。送信データの上位4ビットは、不定値になります(直前の動作コマンドデータの上位4ビットになります)。
4. コントローラ → デバイス  
5バイト目~16バイト目の受信データは、パスワードデータ (12バイト) になります。  
5 バイト目の受信データはフラッシュメモリの 02FEF4H のデータと照合し、6 バイト目の受信データはフラッシュメモリの 02FEF5H のデータと照合します。同様に 16 バイト目の受信データはフラッシュメモリの 02FEFFH のデータと照合します。
5. コントローラ → デバイス  
17バイト目の受信データは、CHECK SUMデータになります。5バイト目から 16 バイト目の送信データを符号なしの8ビット加算 (オーバーフローを無視) して得られた下位 8 ビット値の 2 の補数をコントローラから送信してください。CHECK SUMデータの計算方法は、3.2.4.17「CHECK SUMの計算方法」を参照してください。

## 6. デバイス → コントローラ

18 バイト目の送信データは、5 バイト目～17 バイト目のデータに対する ACK 応答データ (CHECK SUM 値に対する ACK 応答) になります。最初に、5 バイト目～17 バイト目の受信データに受信エラーがあるかをチェックします。受信エラーがある場合、通信異常の ACK 応答データ (bit 3) 68H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。送信データの上位 4 ビットは、直前の動作コマンドデータの上位 4 ビットになるので、“6” になります。

次に、17 バイト目の CHECK SUM データをチェックします。CHECK SUM データのチェック方法は、5 バイト目～17 バイト目までの受信データを符号なし 8 ビット加算 (オーバフローを無視) して得られた値の下位 8 ビットが、00H かどうかをチェックしています。00H 以外の場合、CHECK SUM エラーの ACK 応答データ (bit 0) 61H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。

最後に、パスワードの照合結果をチェックします。5 バイト目～16 バイト目のパスワードデータの照合がすべて一致しない場合、パスワードエラーの ACK 応答データ (bit 0) 61H を送信して、次の動作コマンド (3 バイト目) データ待ちになります。

上記のチェックを終えて、すべて正常なら、正常 ACK 応答データ 60H を送信します。

## 7. デバイス → コントローラ

19 バイト目の送信データはプロテクト設定が正常に終了したかどうかを示します。正常に終了した時は、終了コード(6FH)を返します。

Error が起きた場合は、エラーコード(6CH)を返します。

## 8. デバイス → コントローラ

20 バイト目の送信データは ACK 応答データです。

正常に終了した時は、ACK 応答コード(31H)を返します。

Error の場合は、ErrorACK 応答コード (34H)を返します。

## 9. デバイス → コントローラ

21 バイト目の受信データは、次の動作コマンドデータになります。

## 3.2.4.13 ACK応答データ

ブートプログラムは処理状況を各種コードによってコントローラに送信します。表 3.2.14から表 3.2.19に各受信データに対するACK応答データを示します。ACK応答データの上位 4 ビットは、動作コマンドデータの上位 4 ビットになります。また 3 ビット目は受信エラーを表し、0 ビット目は動作コマンドエラー、CHECK SUMエラー、パスワードエラーの状態を表します。

表 3.2.14 シリアル動作モード判定データに対する ACK 応答データ

送信データ	送信データの意味
86H	UART での通信が可能と判定した。(注)

注) ボーレートの設定が不可能と判定したら、何も送信しないで動作を停止します。

表 3.2.15 動作コマンドデータに対する ACK 応答データ

送信データ	送信データの意味
x8H (注)	動作コマンドデータに受信エラーが発生した。
x6H (注)	プロテクトが設定されており受信コマンドの動作を中止した。
x1H (注)	未定義の動作コマンドデータを正常受信した。
10H	RAM 転送コマンドと判定した。
20H	フラッシュメモリ SUM コマンドと判定した。
30H	製品情報読み出しコマンドと判定した。
40H	フラッシュメモリチップ消去コマンドと判定した。
60H	フラッシュメモリプロテクト設定コマンドと判定した。

注) 上位 4 ビットは、直前の動作コマンドデータの上位 4 ビットになります。

表 3.2.16 RAM 転送コマンド中の CHECK SUM データに対する ACK 応答データ

送信データ	送信データの意味
18H	受信エラーが発生していた。
11H	CHECK SUM エラーが発生した。あるいは、パスワードエラーが発生した。
10H	CHECK SUM 値は正常な値と判定した。

表 3.2.17 フラッシュメモリチップ消去動作に対する ACK 対応データ

送信データ	送信データの意味
54H	消去イネーブルコマンドと判定した。
4FH	消去終了
4CH	消去エラーが発生した。
5DH(注)	消去の再確認
60H(注)	消去エラーの再確認

注) 再確認表記は通信の確認

表 3.2.18 フラッシュメモリプロテクト設定コマンド中の CHECK SUM データに対する ACK 応答データ

送信データ	送信データの意味
68H	受信エラーが発生していた。
61H	CHECK SUM エラーが発生した。あるいは、パスワードエラーが発生した。
60H	CHECK SUM 値は正常な値と判定した。

表 3.2.19 フラッシュメモリプロテクト設定動作に対する ACK 対応データ

送信データ	送信データの意味
6FH	プロテクト(リード/ライト)設定終了
6CH	プロテクト(リード/ライト)設定エラーが発生した。
31H 注)	プロテクト(リード/ライト)設定の再確認
34H 注)	プロテクト(リード/ライト)設定エラーの再確認

注) 再確認表記は通信の確認
----------------

## 3.2.4.14 シリアル動作モード判定

コントローラは、UARTで通信したい場合、所望のボーレートで1バイト目を86Hにて送信してください。図3.2.7に波形を示します。

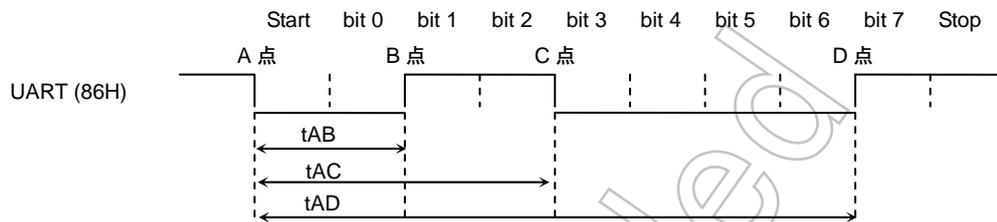


図 3.2.7 シリアル動作モード判定データ

ブードプログラムは、リセット解除後の1バイト目のデータ(86H)はコントローラからシリアル通信のデータとして受信せず、ボーレート判定のため、図3.2.8のフローチャートにあるように、図3.2.7のtAB、tACと、tADの出力期間を判別しています。

CPUが端子のレベルをモニタしてレベルの変化があると、その時のタイム値を取り込みボーレートを判定します。

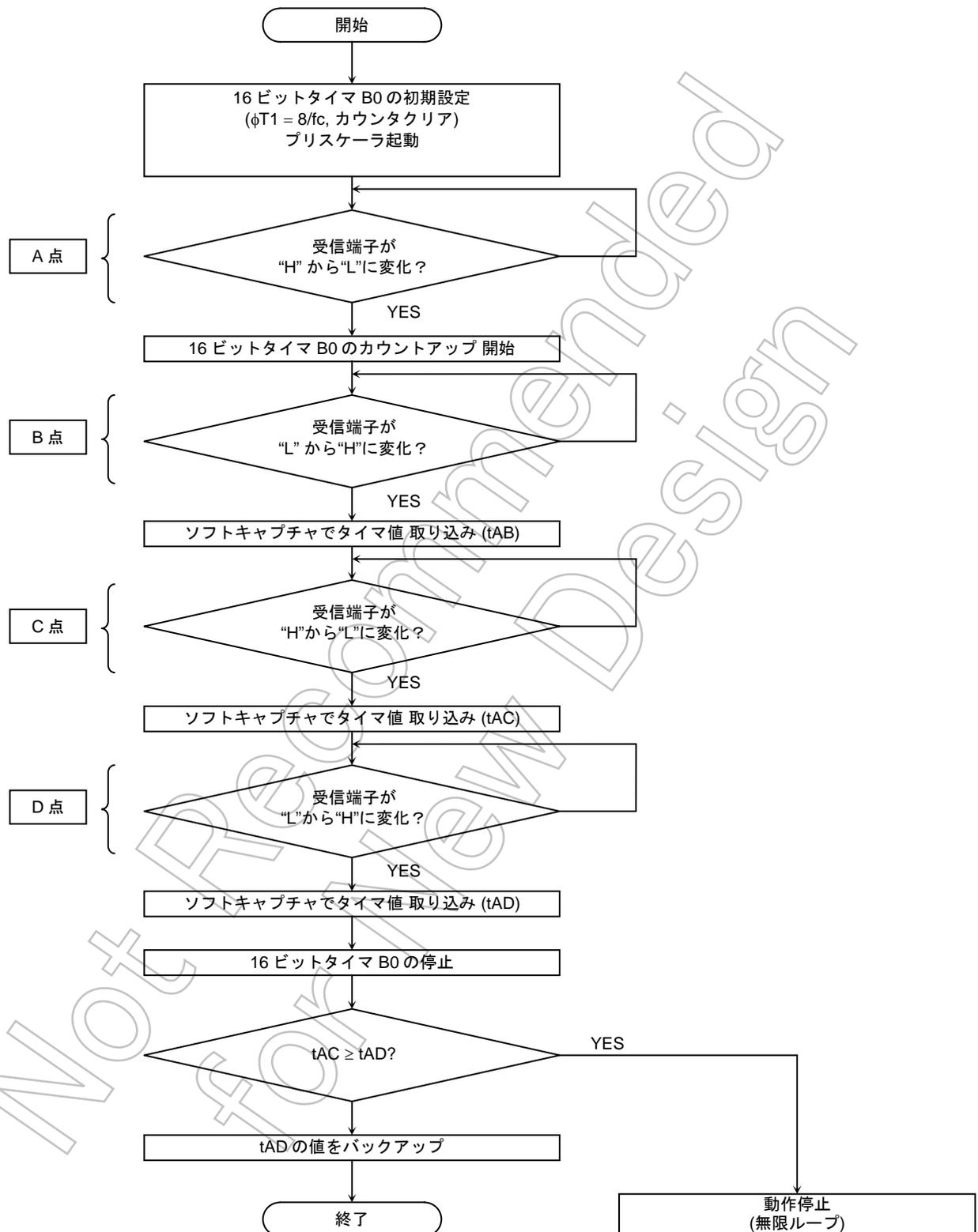


図 3.2.8 シリアル動作モード受信フローチャート

## 3.2.4.15 パスワードについて

動作コマンドデータが RAM 転送コマンド (10H) とフラッシュメモリプロテクト設定コマンド (60H) の場合、パスワードのチェックを行います。まず、動作コマンドデータをエコーバック送信 (10H または 60H) 後、パスワードエリア (02FEF4 番地~02FEFF 番地) のデータ (12 バイト) をチェックします。

次に、5 バイト目 ~ 16 バイト目の受信データ (パスワードデータ) の照合を行います。表 3.2.20 に対応表を示します。

パスワードが 12 バイト分すべて一致しないと、パスワードエラーになります。

また、パスワードが 12 バイト連続して同一データの場合もパスワードエラーになります。但し、パスワードが全て "FFH" で一致し、且つリセットベクタ (02FF00 番地~02FF02 番地) が全て "FFH" の場合は、ブランク品と判断し、パスワードエラーにはなりません。

パスワードエラーと判定された場合、18 バイト目の CHECKSUM 値に対する ACK 応答は、パスワードエラーとなります。

表 3.2.20 受信データと照合するデータの対応

受信データ	照合するデータ
5 バイト目	02FEF4 番地のデータ
6 バイト目	02FEF5 番地のデータ
7 バイト目	02FEF6 番地のデータ
8 バイト目	02FEF7 番地のデータ
9 バイト目	02FEF8 番地のデータ
10 バイト目	02FEF9 番地のデータ
11 バイト目	02FEFA 番地のデータ
12 バイト目	02FEFB 番地のデータ
13 バイト目	02FEFC 番地のデータ
14 バイト目	02FEFD 番地のデータ
15 バイト目	02FEFE 番地のデータ
16 バイト目	02FEFF 番地のデータ

## パスワードに指定できないデータ例

## ブランク品の場合 (注)

・ ALL "FF" (FFH, FFH, FFH) 以外のデータはパスワードに指定できません。ブランク品は必ず ALL "FF" を指定します。

注) ブランク品とはパスワードエリア (02FEF4 番地~02FEFF 番地) とリセットベクタ (02FF00 番地~02FF02 番地) が全て "FFH" となっている製品を示しています。

## 書き込み品の場合

・ 12 バイト連続の同一データはパスワードに指定できません。

下表に書き込み品のパスワードエラー例を示します。

書き込み品	1	2	3	4	5	6	7	8	9	10	11	12	備考
エラー例 1	FFH	ALL "FF"											
エラー例 2	00H	ALL "00"											
エラー例 3	5AH	ALL "5A"											

### 3.2.4.16 SUMの計算方法

SUMの計算方法は、バイト+バイト+バイト+……+バイトの結果をワードで返します。つまり、バイトでデータを読み出して符号なし8ビット加算を行い、計算結果を16ビットで求めています。

コントローラへは、SUMの上位8ビットデータ、下位8ビットデータの順番で送信します。SUMの計算対象のデータは、フラッシュメモリ全エリア(128Kバイト)のデータになります。フラッシュメモリSUMコマンドを実行したときに返されるSUMは、本計算方法を使用しています。

例)	A1H	左記4バイトが計算対象データの場合、SUMの値は、 $A1H + B2H + C3H + D4H = 02EAH$ となるので、 SUMの上位のデータは、02H SUMの下位のデータは、EAH になります。 したがって、コントローラには02H、EAHの順番で送信します。
	B2H	
	C3H	
	D4H	

### 3.2.4.17 CHECK SUMの計算方法

CHECK SUMの計算方法は、送信データを符号なし8ビット加算(オーバフローを無視)して得られた下位8ビット値の2の補数値を求めています。フラッシュメモリSUMコマンド、製品情報読み出しコマンドを実行したときに返されるCHECK SUMは、本計算方法を使用しています。また、コントローラはCHECK SUM値を送信するときは、本計算方法を使用してください。

例) フラッシュメモリSUMコマンドのときを例に説明します。

SUMの上位8ビットデータがE5H、下位8ビットデータがF6Hの場合のCHECK SUM値を求めます。

まず、符号なし8ビット加算して得られた値を求めます。

$$E5H + F6H = 1DBH$$

この値の下位8ビットに対しての2の補数をとると以下のようになり、この値がCHECK SUM値になります。したがって、コントローラには25Hを送信します。

$$0 - DBH = 25H$$

### 3.2.5 ユーザブートモード (シングルチップモード上)

ユーザブートモードでは、ユーザ作成のフラッシュメモリ書き込み/消去プログラムを使用してフラッシュメモリを書き替えることが出来ます。シングルチップモードにおいて、通常のユーザアプリケーションプログラムが動作しているノーマルモードから、フラッシュメモリを書き替えるためのユーザブートモードに移行させることでフラッシュメモリの書き替えが行えます。

例えば、シングルチップモード起動時にノーマル/ユーザブートモードを確定したい場合は、条件判定を行うプログラムをユーザアプリケーションプログラムのリセット処理ルーチンの中に組み込みます。なお、モード切り替えの条件設定は、本デバイスの I/O を使用してユーザのボード条件に合わせて独自に構築してください。

従って、ユーザブートモードにてフラッシュメモリの書き替えを行う場合には、あらかじめフラッシュメモリ書き込み/消去プログラムをユーザアプリケーションプログラムに組み込んでおいてください。なお、内蔵フラッシュメモリは消去/書き込み動作中はフラッシュメモリのデータを読み出せないため、フラッシュメモリ書き込み/消去プログラムはフラッシュメモリ外に格納して実行させる必要があります。また、ユーザブートモードで内蔵フラッシュメモリの消去/書き込み動作中は、割り込み発生を禁止してください。

書き込み/消去ルーチンを内蔵フラッシュメモリに置く場合と、外部から転送する場合の 2 ケース を例に、以下(1-A)、(1-B) にその手順を説明します。

Not Recommended for New Design

3.2.5.1 (1-A)書き込み／消去方法手順例1

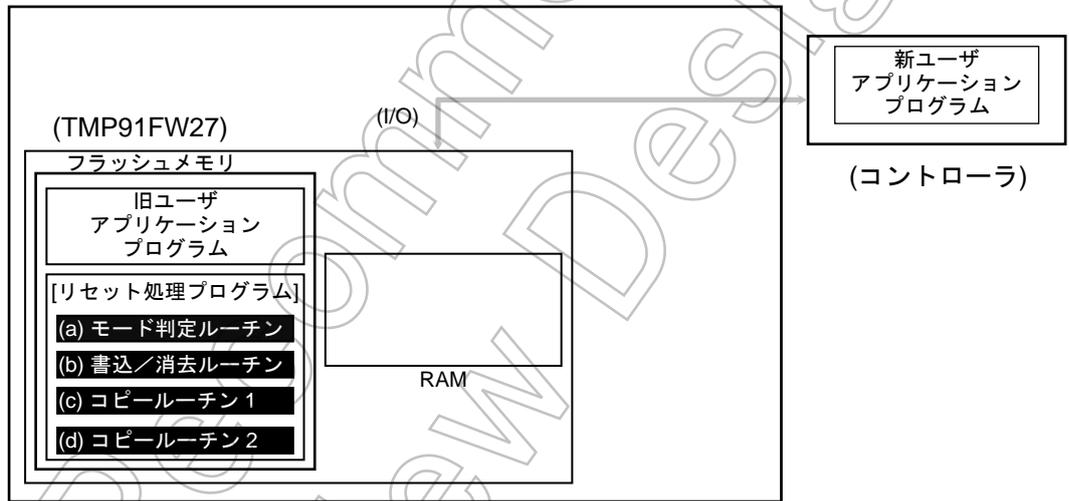
書き込み／消去プログラムをフラッシュメモリにあらかじめ内蔵している場合の手順例

(Step-1)環境準備

ユーザは、あらかじめどのような条件(例えば端子状態)に設定されたらユーザブート動作に移行するか、どの I/O バスを使用してデータ転送を行うかを決め、それに合ったデバイス周辺回路の設計、プログラムの作成を行います。ユーザは、あらかじめフラッシュメモリ上の任意のブロックに、以下に示す4つのプログラムを書き込んでおきます。

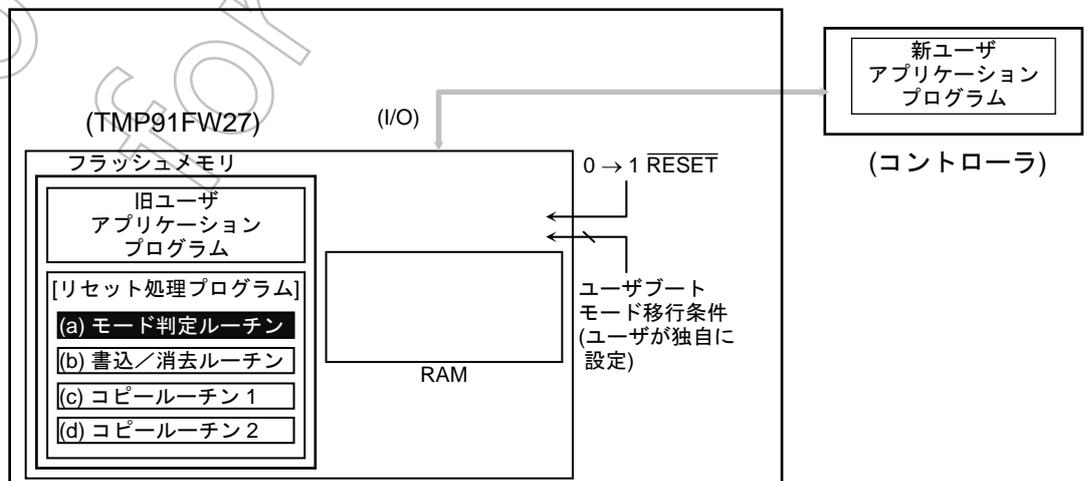
- (a) モード判定ルーチン: 書き込み／消去動作に移るためのプログラム
- (b) フラッシュ書き替えルーチン: 書き込み／消去データを外部から取り込み、フラッシュメモリを書き込み／消去するためのプログラム
- (c) コピー処理プログラム 1: (a)~(d) を内蔵 RAM または外部メモリにコピーするためのプログラム
- (d) コピー処理プログラム 2: 内蔵 RAM または外部メモリに書かれた(a)~(d)をフラッシュメモリへコピーするためのプログラム

注) (d)は書き替えプログラムをフラッシュ上に復元するプログラムです。常にフラッシュメモリの全領域が書き替えになり、書き替えデータのプログラムに同様のプログラムが存在する場合は不要になります。



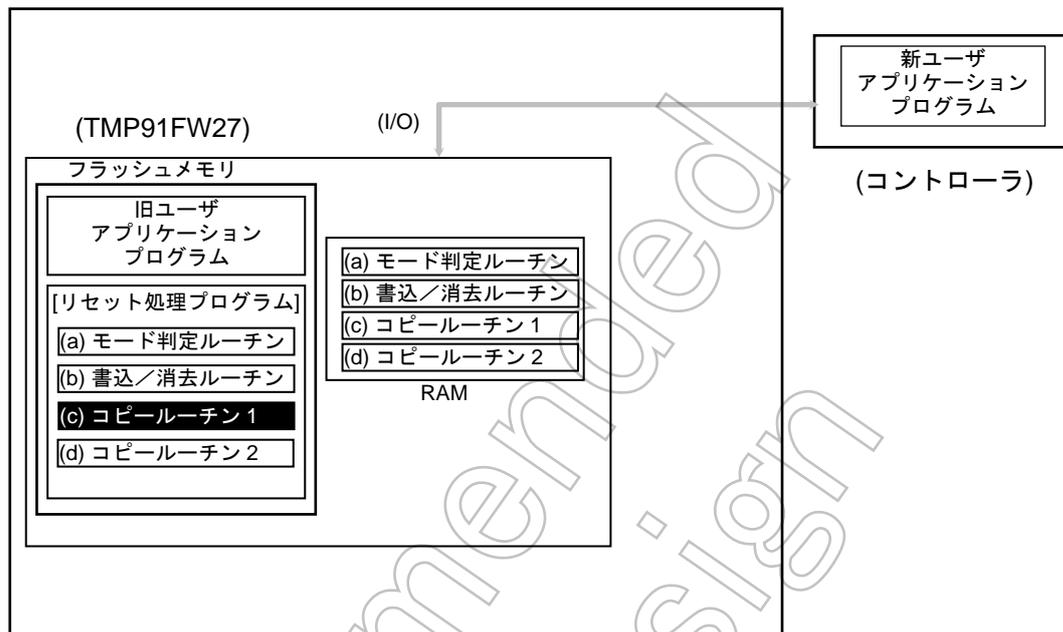
(Step-2) ブートの起動(モード判定を RESET 処理に組み込んだ場合)

リセット解除後のリセット処理プログラムにおいてユーザブートモードへの移行を判定します。このとき、移行条件(端子設定等)が整っていれば、プログラムは書き替えのためのユーザブートモードに移ります。



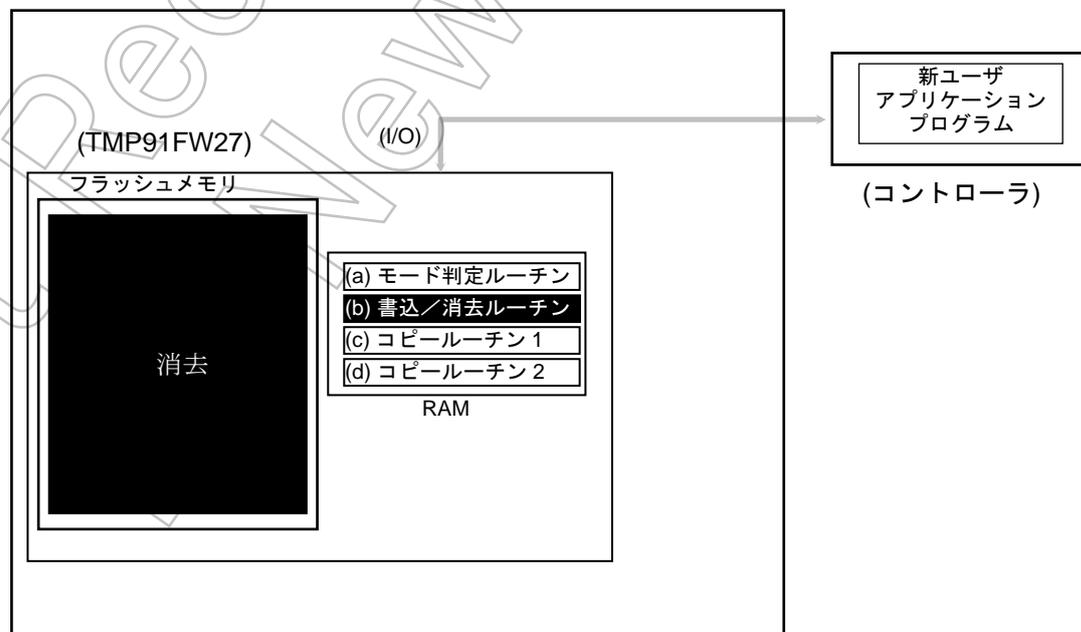
**(Step-3) ユーザブートの起動&RAM への書き込み/消去ルーチンのコピー**

ユーザブートモードに移ると、(c) コピールーチン 1 を使用して、(a)~(d) を内蔵 RAM もしくは外部メモリにコピーします。(下図は内蔵 RAM へコピーした場合を示します。)

**(Step-4) 書き換えルーチンによるフラッシュの消去**

RAM 上の書き込み/消去ルーチンへジャンプし、旧ユーザプログラムエリアの消去(セクタ単位、もしくはチップ消去)を行います。(RAM 上からフラッシュメモリに消去コマンドを与える場合。)

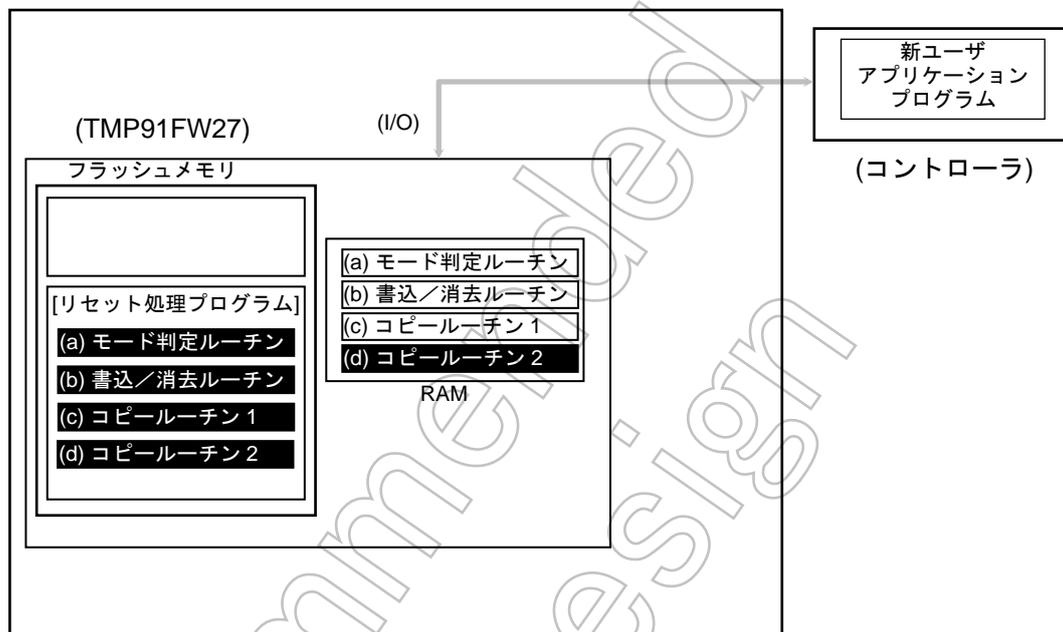
注)セクタ単位消去により、(a)~(d)のルーチンをフラッシュメモリに残す場合は(b)の書き込み/消去ルーチンだけ、RAM 上にコピーします。



(Step-5) ユーザブートプログラムのフラッシュメモリ上への復帰

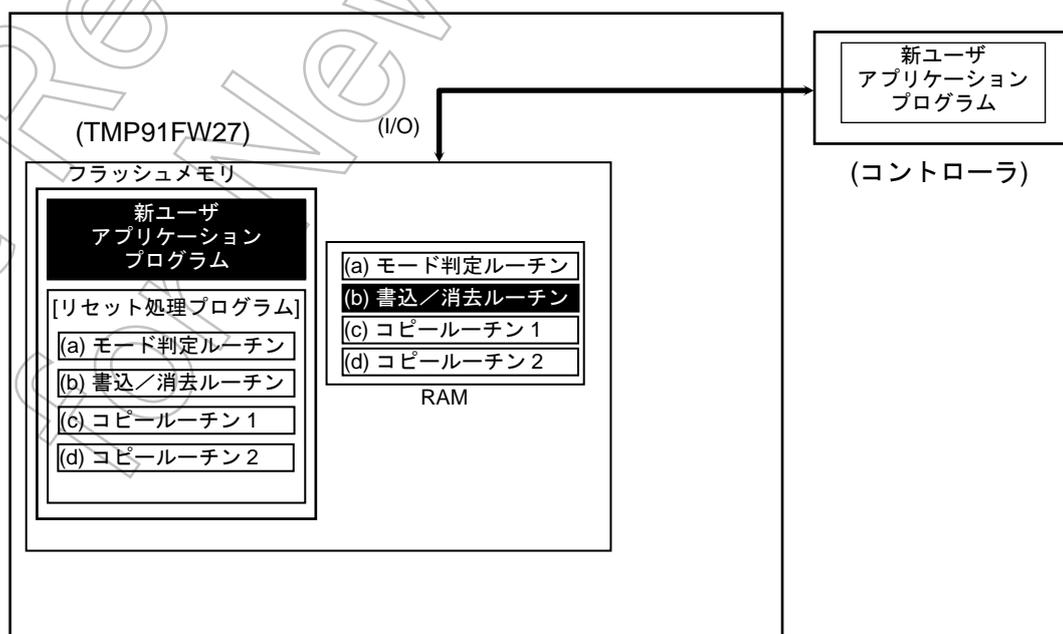
RAM 上のコピールーチン 2 を使用して、(a)~(d) をフラッシュメモリにコピーします。

注)セクタ単位消去により、(a)~(d)のルーチンをフラッシュメモリに残す場合、Step5 は不要です。



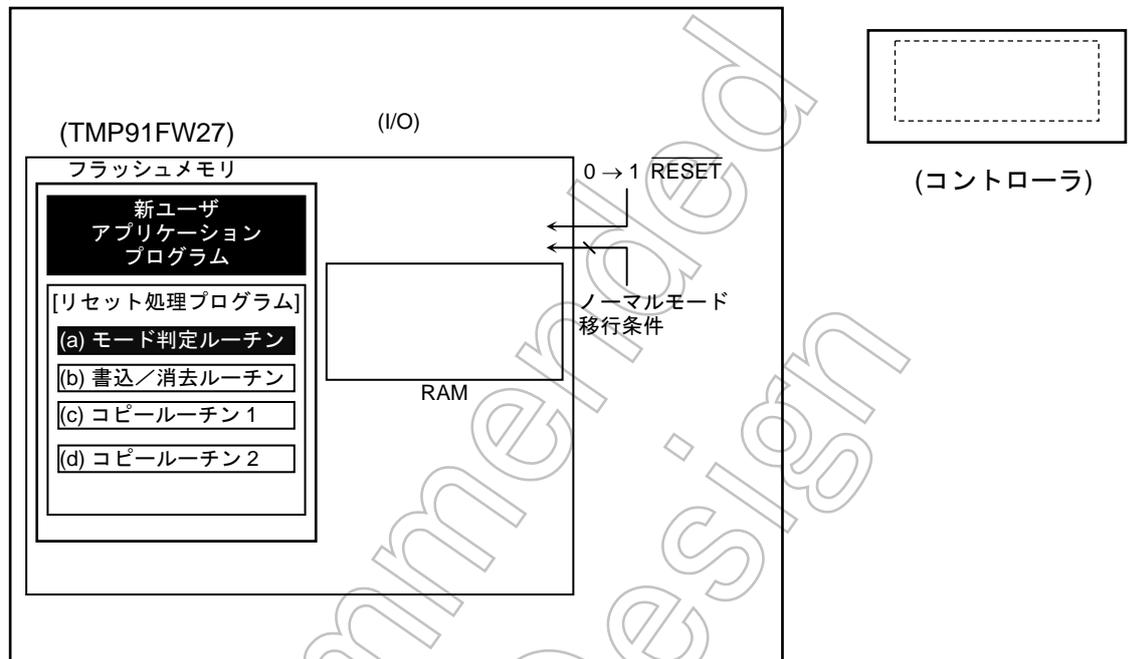
(Step-6) 新ユーザアプリケーションプログラムのフラッシュへの書き込み

さらに、RAM 上の書き替えルーチンを実行して、転送元 (コントローラ) より新ユーザアプリケーションプログラムのデータをロードし、フラッシュメモリの消去したエリアに書き込みを行います。



(Step-7) 新ユーザアプリケーションプログラムの起動

$\overline{\text{RESET}}$  入力端子を“0”にしてリセットを行い、設定条件をノーマルモードに設定します。  
リセット解除後、新ユーザアプリケーションプログラムで動作を開始します。



Not Recommended for New Design

3.2.5.2 (1-B)書き込み/消去方法手順例 2

以下は(1-A)とは異なり、必要最低限のブートプログラムをフラッシュ上に載せ、他に必要なプログラムはコントローラから供給する場合の手順です。

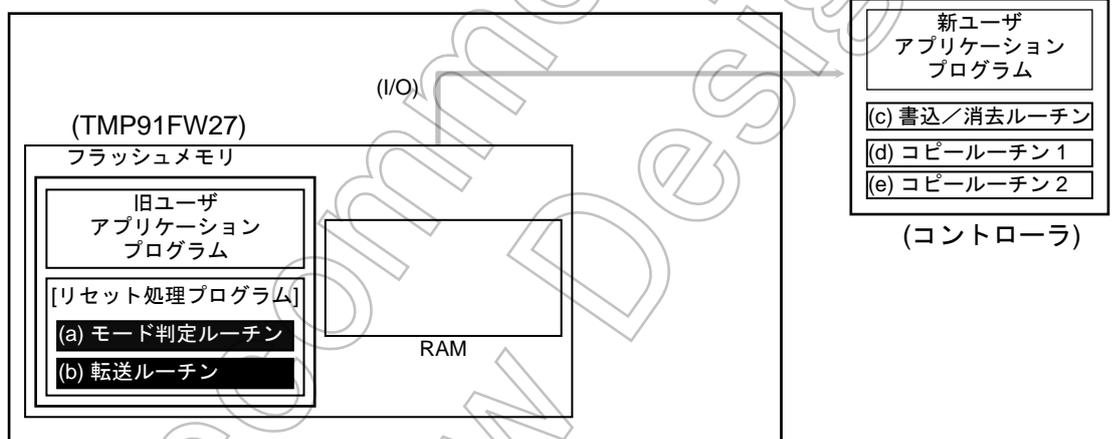
(Step-1)環境準備

ユーザは、あらかじめどのような条件(例えば端子状態)に設定されたらユーザブート動作に移行するか、どの I/O バスを使用してデータ転送を行うかを決め、それに合ったデバイス周辺回路の設計、プログラムの作成を行います。ユーザは、あらかじめフラッシュメモリ上の任意のブロックに、以下に示す 2つのプログラムを書き込んでおきます。

- (a) モード判定ルーチン: 書き込み/消去動作に移るためのプログラム
- (b) 転送ルーチン: 書き込み/消去プログラムを外部から取り込むためのプログラム

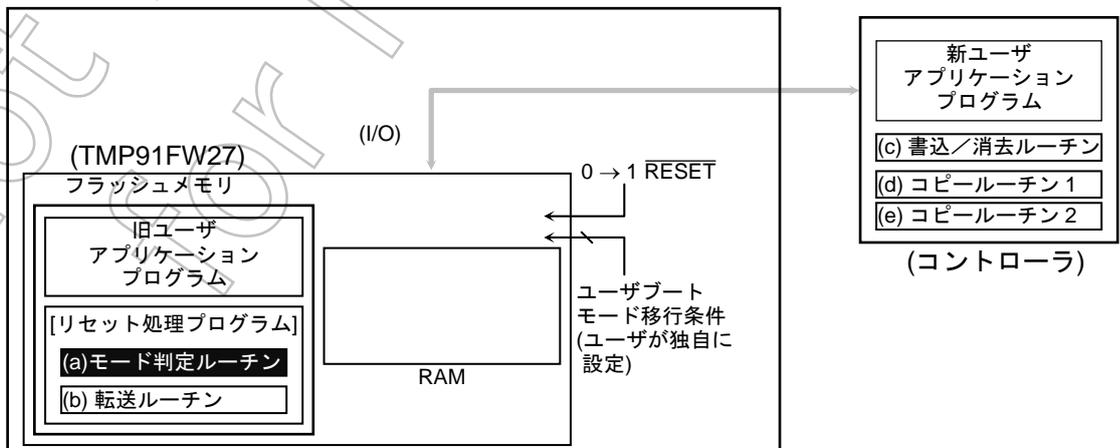
また、下記に示すルーチンはコントローラ上に用意します。

- (c) 書き込み/消去ルーチン: 書き込み/消去を行うためのプログラム
- (d) コピールーチン 1: (a),(b)を内蔵 RAM または外部メモリにコピーするためのプログラム
- (e) コピールーチン 2: 内蔵 RAM または外部メモリに書かれた(a),(b)をフラッシュメモリへコピーするためのプログラム



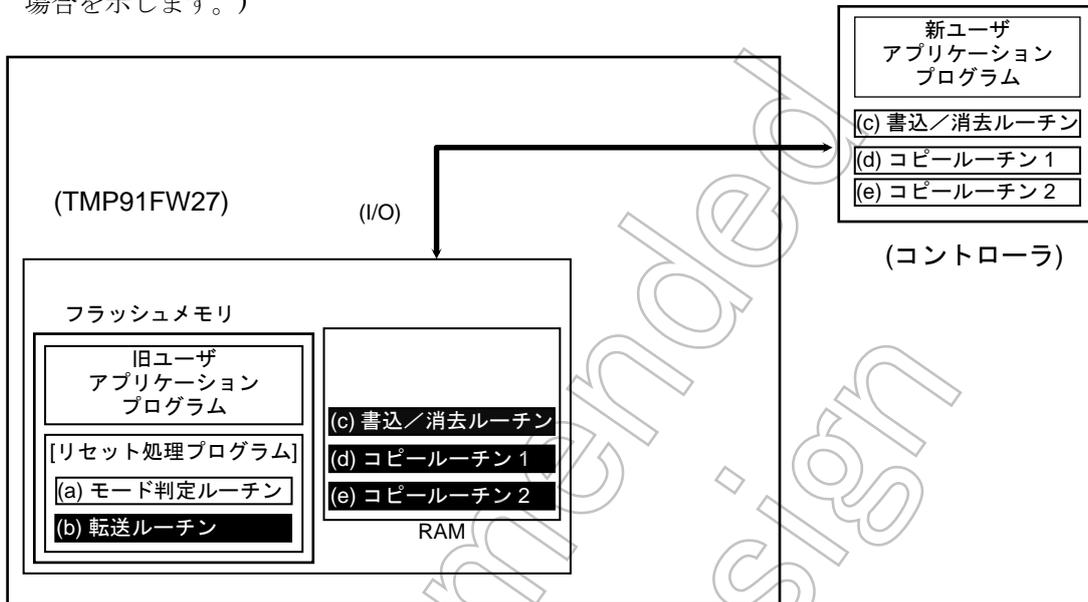
(Step-2) ブートの起動(モード判定を RESET 処理に組み込んだ場合)

リセット解除後のリセット処理プログラムにおいてユーザブートモードへの移行を判定します。このとき、移行条件(端子設定等)が整っていれば、プログラムは書き替えのためのユーザブートモードに移ります。



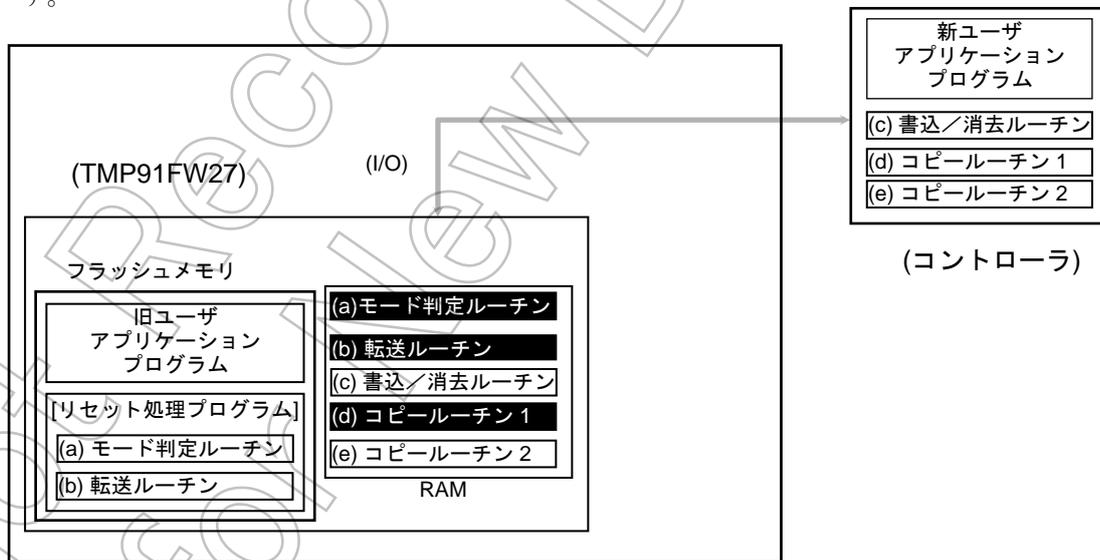
(Step-3) ユーザブートの起動&RAM への書き込み/消去ルーチンのコピー

ユーザブートモードに移ると、(b) 転送ルーチンを使用して、転送元(コントローラ)より (c)~(e)を内蔵 RAM(又は外部メモリ)にロードします。(下図は内蔵 RAM へコピーした場合を示します。)



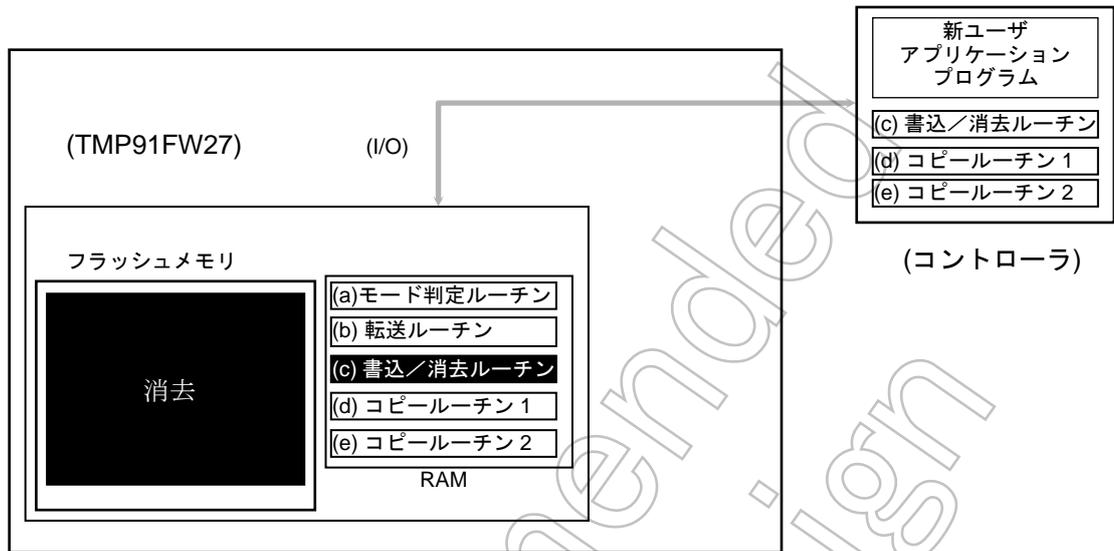
(Step-4) RAM 上のルーチンを起動

RAM 上に制御を移し、(d) コピールーチン 1 を使用して、(a),(b) を RAM にコピーします。

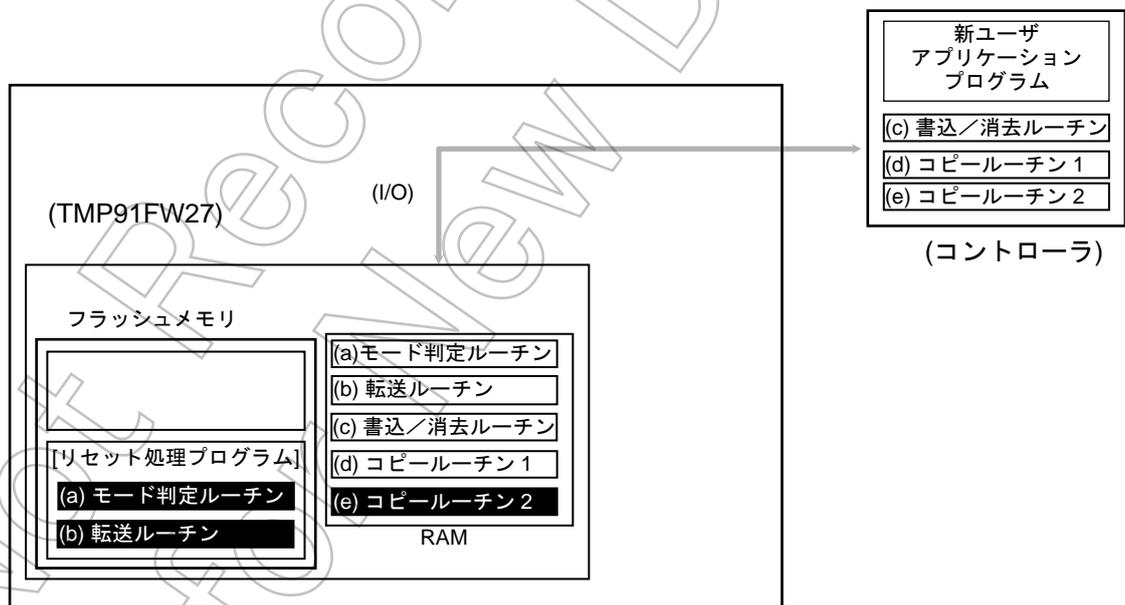


*(Step-5)* 書き替えルーチンによるフラッシュの消去

(c)の書き替えルーチンを使用して、旧ユーザプログラムエリアの消去を行います。

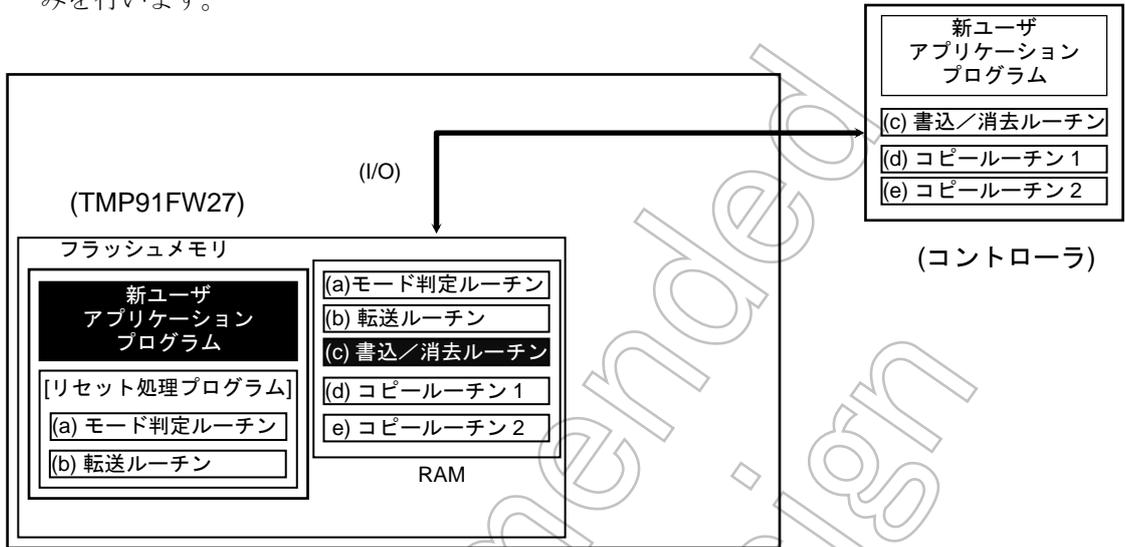
*(Step-6)* ユーザブートプログラムのフラッシュメモリ上への復帰

(e)のコピールーチン2を使用して、RAM上の(a),(b)をフラッシュメモリにコピーします。



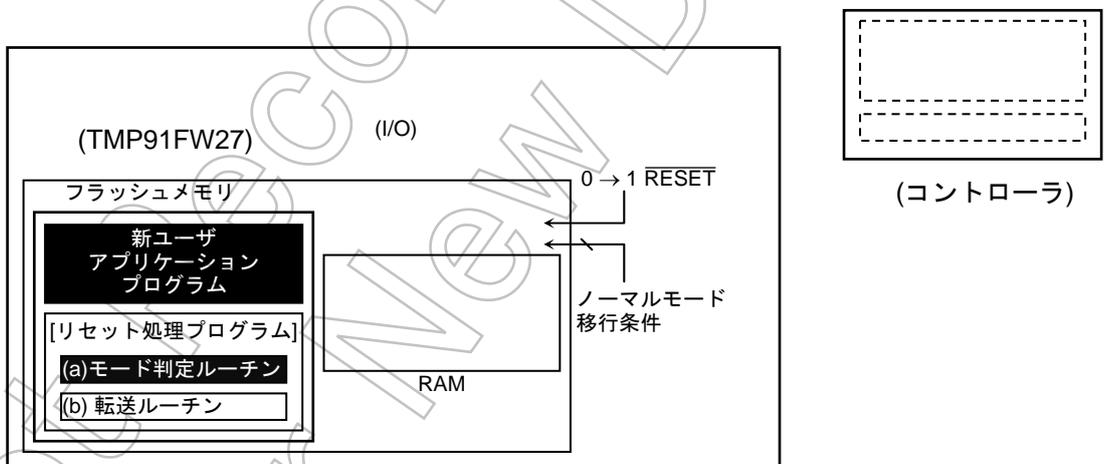
(Step-7) 新ユーザアプリケーションプログラムのフラッシュへの書き込み

さらに、RAM 上の (c) 書き込み/消去ルーチンを実行して、転送元 (コントローラ) より新ユーザアプリケーションプログラムのデータをロードし、消去したエリアに書き込みを行います。



(Step-8) 新ユーザアプリケーションプログラムの起動

RESET 入力端子を“0”にしてリセットを行い、設定条件をノーマルモードに設定します。リセット解除後、新ユーザアプリケーションプログラムで動作を開始します。



### 3.2.6 フラッシュメモリコマンドシーケンス

フラッシュメモリの動作は、6つのコマンドから構成されています。表 3.2.21にコマンドシーケンスの詳細を示します。コマンドシーケンスで指定するアドレスは、フラッシュメモリがマッピングされている領域の何れかを指定する必要があります。詳細は、表 3.2.3を参照してください。

表 3.2.21 コマンドシーケンス

	コマンド シーケンス	第1バス ライトサイクル		第2バス ライトサイクル		第3バス ライトサイクル		第4バス ライトサイクル		第5バス ライトサイクル		第6バス ライトサイクル	
		Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data	Addr.	Data
1	1ワード書き込み	AAAH	AAH	554H	55H	AAAH	A0H	PA (注1)	PD (注1)				
2	セクタイレース (4KB単位の部分消去)	AAAH	AAH	554H	55H	AAAH	80H	AAAH	AAH	554H	55H	SA (注2)	30H
3	チップイレース (全面消去)	AAAH	AAH	554H	55H	AAAH	80H	AAAH	AAH	554H	55H	AAAH	10H
3	Product ID Entry	AAAH	AAH	554H	55H	AAAH	90H						
5	Product ID Exit	xxH	F0H										
	Product ID Exit	AAAH	AAH	554H	55H	AAAH	F0H						
6	リードプロテクト設定	AAAH	AAH	554H	55H	AAAH	A5H	77EH	F0H (注3)				
	ライトプロテクト設定	AAAH	AAH	554H	55H	AAAH	A5H	77EH	0FH (注3)				

注1) PA=プログラムワードアドレス、PD=プログラムワードデータ

書き込みを行うアドレスとデータを設定してください。アドレスは偶数アドレスを設定してください。

注2) SA=セクタアドレス

アドレスのA23～A12で個々のセクタ消去範囲が選択されます。

注3) リードプロテクトとライトプロテクトの両方を設定する場合は、データ 00Hを書き込み必ず一度に設定してください。

表 3.2.22 ハードウェアシーケンスフラグー覧

状態		D7	D6
自動動作実行中	1ワード書き込み	D7反転	トグル
	セクタ/チップイレース	0	トグル
	リード/ライトプロテクト設定	使用不可	トグル

注) D15～D8、D5～D0はDon't care。

### 3.2.6.1 1ワード書き込み

1ワード単位でフラッシュメモリの書き込みを行います。第4バスライトサイクルで書き込みを行うアドレスとデータを指定します。1ワードあたりの書き込み時間は最大60 $\mu$ sです。書き込みが終了するまでは、他のコマンドシーケンスを実行することができません。書き込み終了を確認するには、フラッシュメモリの同一アドレスをリードし同一データが読み出せるまでポーリングを行います。書き込み中はリードするたびにデータビット6が反転します。

**注) 既にデータ (FFFFH を含む) が書き込まれたフラッシュメモリのアドレスに対して、再度データの書き込みを行う場合は、セクタイレースまたはチップイレースによって、必ずそのアドレスのデータを消去した後に書き込みを実行してください。**

### 3.2.6.2 セクタイレース(4KB単位の部分消去)

4Kバイト単位でフラッシュメモリの消去を行います。消去範囲は第6バスライトサイクルのアドレスで指定します。アドレスの指定は表3.2.3を参照してください。なお、セクタイレースは、ライターモード時は動作しません。

4Kバイトあたりの消去時間は、最大75msです。消去が終了するまでは、他のコマンドシーケンスを実行することができません。消去終了を確認するには、フラッシュメモリの同一アドレスをリードし同一データが読み出せるまでポーリングを行います。消去中はリードするたびにデータビット6が反転します。

### 3.2.6.3 チップイレース(全面消去)

フラッシュメモリの全領域を消去します。

全領域の消去時間は、最大300msです。消去が終了するまでは、他のコマンドシーケンスを実行することができません。消去終了を確認するには、フラッシュメモリの同一アドレスをリードし同一データが読み出せるまでポーリングを行います。消去中はリードするたびにデータビット6が反転します。

なお、消去された領域のデータはFFHとなります。

### 3.2.6.4 Product ID Entry

Product ID Entryを実行するとProduct IDモードが起動します。Product IDモード中、フラッシュメモリに対してリード命令を実行するとベンダーID、フラッシュマクロID、フラッシュサイズID、リード/ライトプロテクトステータスを読み出すことができます。なお、Product IDモード中は、フラッシュメモリのデータは読み出せません。

### 3.2.6.5 Product ID Exit

Product IDモードを終了します。

### 3.2.6.6 リードプロテクト設定

フラッシュメモリに対してリードプロテクトを設定します。リードプロテクトを設定するとライターモードのときフラッシュメモリのリードが出来なくなります。シングルブートモードのときは、RAM 転送コマンドが実行できなくなります。

リードプロテクト設定を解除するには、チップイレースを実行する必要があります。リードプロテクトが設定されているか確認するには Product ID モードで xxx77EH をリードします。リードプロテクトの設定時間は最大 60 $\mu$ s です。リードプロテクトの設定が終了するまでは、他のコマンドシーケンスを実行することができません。リードプロテクト設定の終了を確認するには、フラッシュメモリの同一アドレスをリードし同一データが読み出せるまでポーリングを行います。リードプロテクト設定中はリードするたびにデータビット 6 が反転します。

### 3.2.6.7 ライトプロテクト設定

フラッシュメモリに対してライトプロテクトを設定します。ライトプロテクトを設定するとライターモードのときフラッシュメモリのライトが出来なくなります。シングルブートモードのときは、RAM 転送コマンドが実行できなくなります。

ライトプロテクト設定を解除するには、チップイレースを実行する必要があります。ライトプロテクトが設定されているか確認するには Product ID モードで xxx77EH をリードします。ライトプロテクトの設定時間は最大 60 $\mu$ s です。ライトプロテクトの設定が終了するまでは、他のコマンドシーケンスを実行することができません。ライトプロテクト設定の終了を確認するには、フラッシュメモリの同一アドレスをリードし同一データが読み出せるまでポーリングを行います。ライトプロテクト設定中はリードするたびにデータビット 6 が反転します。

### 3.2.6.8 ハードウェアシーケンスフラグ

フラッシュメモリの自動動作実行状態を、ハードウェアシーケンスフラグにより確認できます。

#### 1) DATA ポーリング(D7)

フラッシュメモリの書き込みを実行すると、書き込み処理が完了するまでの間、D7に書き込んだデータの反転データを出力し、完了後はD7のセルデータを出力します。D7を読み出すことで動作状態の識別ができます。セクタ/チップイレース処理実行中はD7から“0”を出力し、完了後は“1”(セルデータ)を出力します。その後1 $\mu$ s待って読み出すと全ビットの書き込みデータが読み出せます。

リード/ライトプロテクト設定を実行した場合は、DATAポーリング機能は使用できませんので、トグルビット(D6)にて動作状態を識別してください。

#### 2) トグルビット(D6)

フラッシュメモリの書き込み、セクタ/チップイレース、リード/ライトプロテクト設定を実行すると、これらの処理が完了するまでの間、リードオペレーションによって読み出されるデータの6ビット目(D6)の値はリードするたびに0と1が交互に出力されます。これを利用すると各処理の完了をソフト的に確認することができます。通常はフラッシュメモリの同一アドレスに対しリードを行い同一データが読み込まれるまでポーリングを行います。なお、トグルビットの最初の値はかならず“1”になります。

注) 内蔵されているフラッシュメモリには、内部タイマ超過ビット(D5)機能が無いため、タイマにてDATAポーリングおよびトグルビットのポーリング時間を設定して、書き込み動作、イレース動作の最大時間を超えた場合にポーリングを中止するようにプログラムを設定する必要があります。

### 3.2.6.9 データリード

フラッシュメモリからのデータリードは、バイト単位またはワード単位で行います。なお、データリードの場合はコマンドシーケンスの実行は必要ありません。

### 3.2.6.10 内部CPUによるフラッシュメモリ書き替え

内部 CPU によるフラッシュメモリ書き替えは、上述のコマンドシーケンス、ハードウェアシーケンスフラグを使って行います。ただし、内蔵フラッシュメモリは、自動動作モード中はメモリデータを読み出せないため、書き替えプログラムをフラッシュメモリエリア外で実行する必要があります。

内部 CPU によるフラッシュメモリ書き替えには、2通りの方法があります。あらかじめ用意されたシングルブートモードを使う方法と、シングルチップモード上でユーザ独自のプロトコルを使う方法(ユーザブート)です。

#### 1) シングルブート :

マイコンをシングルブートモードで起動させ、内蔵ブート ROM プログラムにより、フラッシュメモリを書き替える方法です。このモードでは、内蔵ブート ROM が割り込みベクタテーブルを含む領域にマッピングされ、ブート ROM プログラムが実行されます。また、フラッシュメモリはブート ROM 領域とは別のアドレス空間にマッピングされます。ブート ROM プログラムは、シリアル転送による書き替え用データ取り込み、およびフラッシュメモリの書き替えを行います。シングルブートは割り込み禁止状態で行います。また、ノンマスクカブル割り込み (NMI 等) も発生しないように処理しておく必要があります。

詳細は 3.2.4 「シングルブートモード」を参照してください。

#### 2) ユーザブート :

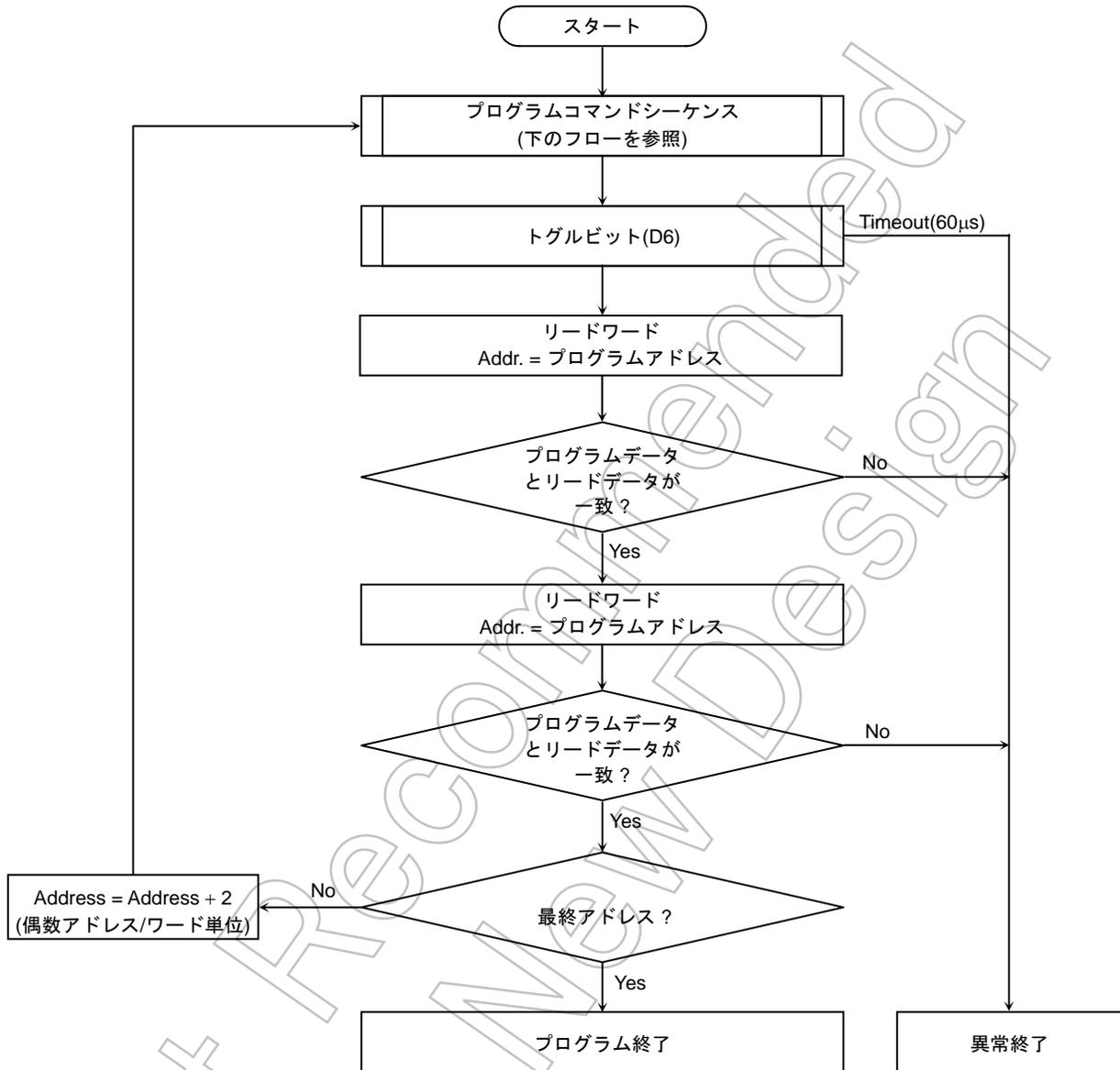
ユーザ独自のフラッシュメモリ書き替えプログラムを使う方法です。シングルチップモード(通常動作モード)で実行します。このモードでも、フラッシュメモリ領域とは別のアドレス空間上でフラッシュメモリ書き替えプログラムを実行させる必要があります。また、シングルブートと同様に、ノンマスクカブルも含めたすべての割り込み発生を禁止する必要があります。

フラッシュメモリ書き替えプログラムは、書き替え用データ取り込みルーチン、フラッシュメモリ書き替えルーチンを含めて、あらかじめ用意しておきます。メインプログラム上で、通常動作からフラッシュメモリ書き替え動作へ切り替え、用意しておいたフラッシュメモリ書き替えプログラムをフラッシュメモリ領域外に展開して実行します。例えば、フラッシュメモリ書き替えプログラムをフラッシュメモリ上から内蔵 RAM へ展開して実行したり、外部メモリ上に用意して実行したりできます。

詳細は 3.2.5 「ユーザブートモード(シングルチップモード上)」を参照してください。

フローチャート：内部 CPU によるフラッシュメモリアクセス

1 ワード書き込み



プログラムコマンドシーケンス (アドレス/データ)

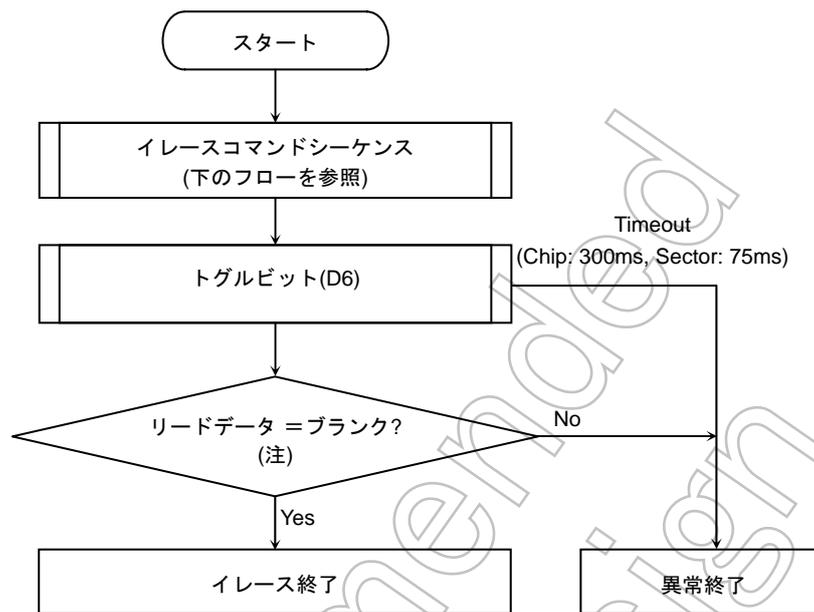
xxxAAAH/AAH

xxx554H/55H

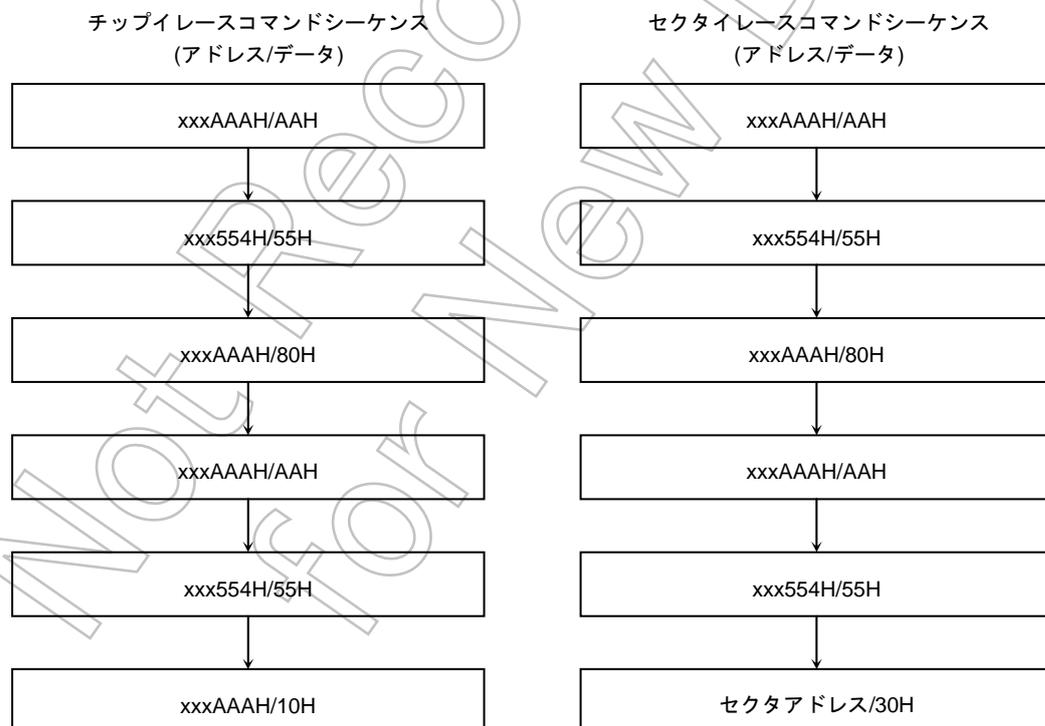
xxxAAAH/A0H

偶数プログラムアドレス (A0 = 0)  
/ プログラムデータ (ワード単位)

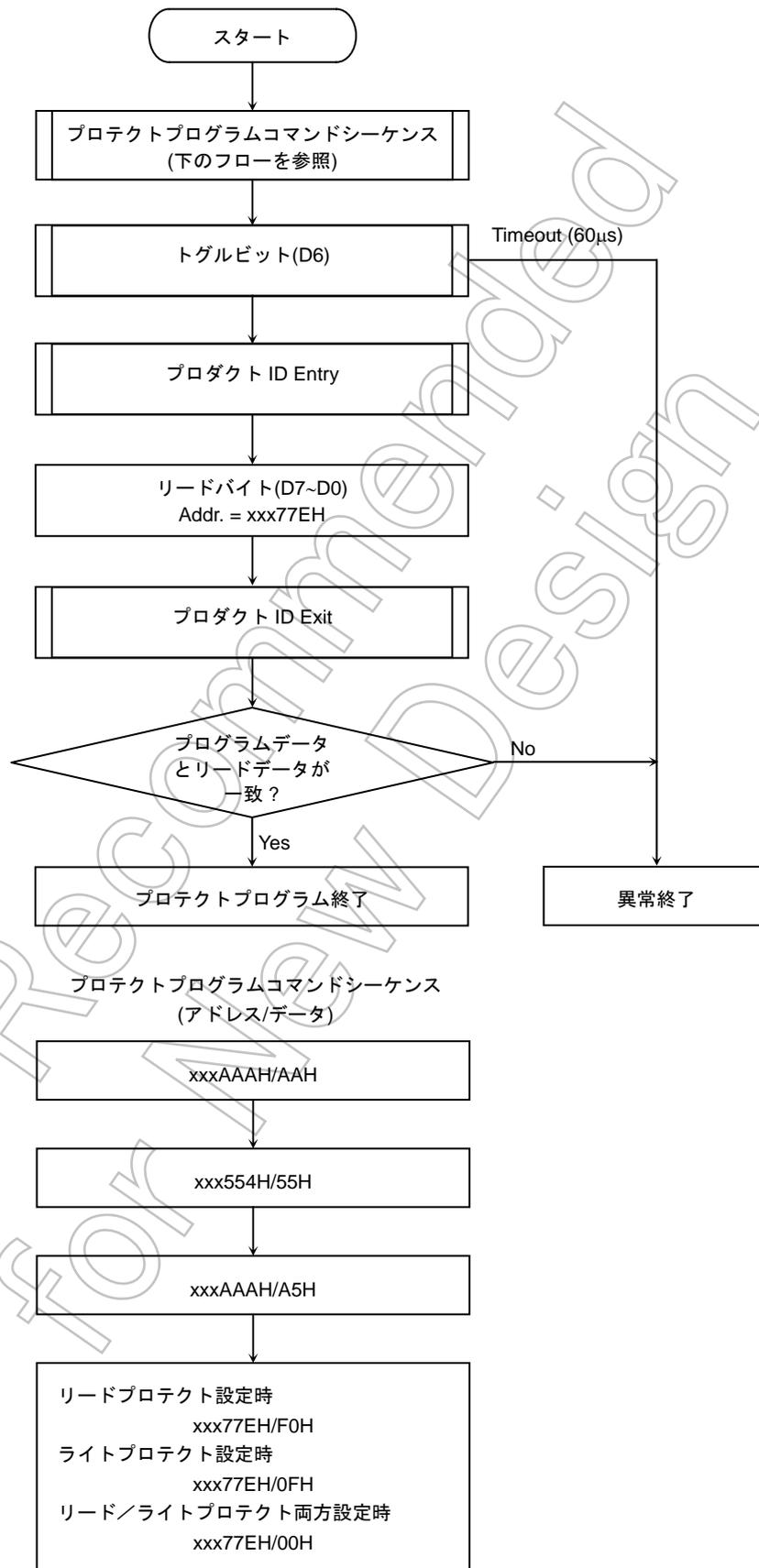
チップイレース/セクタイレース

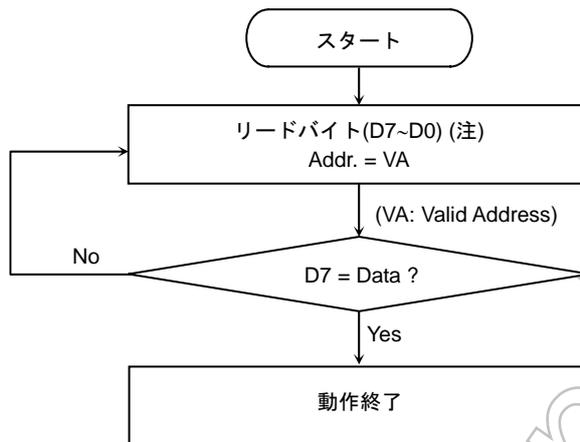
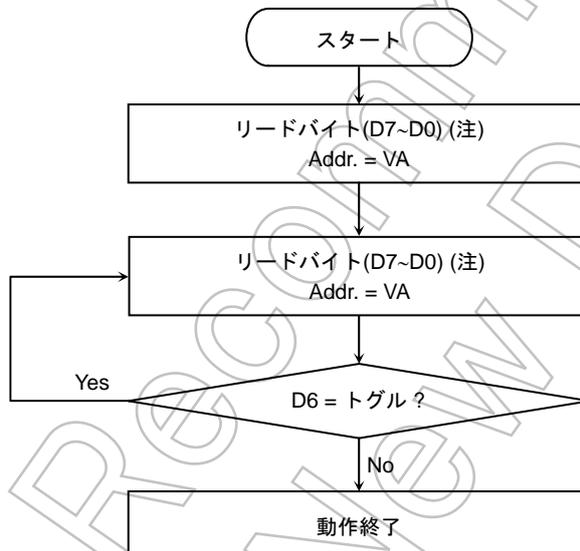


注) チップイレース時は、フラッシュメモリの全エリアがブランクであることを確認する。  
セクタイレース時は、選択したセクタエリアがブランクであることを確認する。



## リード/ライトプロテクト設定



DATA ポーリング(D7)トグルビット(D6)

注) ハードウェアシーケンスフラグの読み出しは、バイト単位、またはワード単位で行います。

VA: 1ワード書き込み時は、書き込みを行っているアドレス。

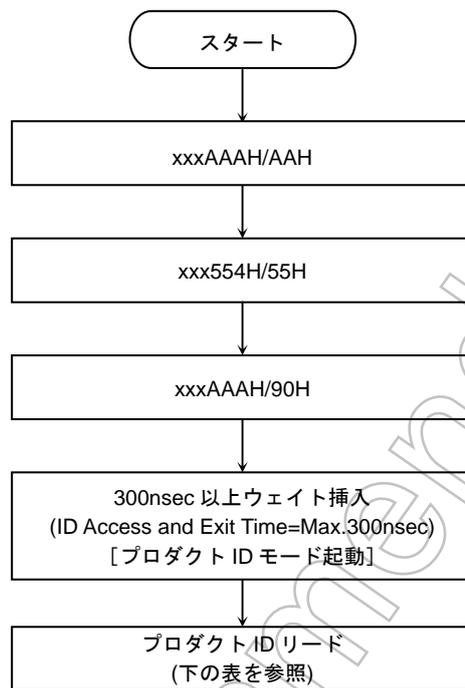
セクタイレース時は、選択したセクタ内の任意アドレス。

チップイレース時は、フラッシュメモリの任意アドレス。

リードプロテクト時は、プロテクト設定アドレス(0x77EH)。

ライトプロテクト時は、プロテクト設定アドレス(0x77EH)。

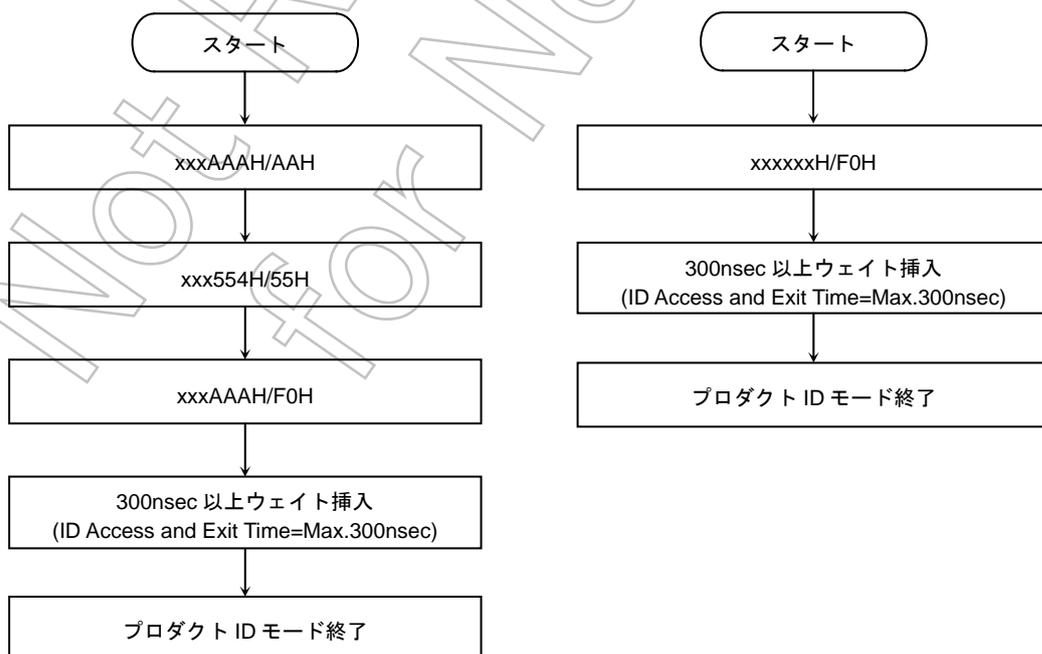
プロダクト ID Entry



プロダクト ID Entry 後のリード値

ベンダーID	アドレス	読み出される値
フラッシュマクロ ID	xxx00H	98H
フラッシュサイズ ID	xxx02H	42H
リード/ライト	xxx04H	1FH
プロテクトステータス	xxx77EH	プロテクト設定時は書き込んだデータ プロテクト未設定時は FFH

プロダクト ID Exit



(プログラム例：RAMに展開するプログラム)

チップイレースを実行した後、FE0000Hに0706Hのデータを書き込む

```

##### フラッシュメモリチップイレース処理 #####
ld      XIX, 0xFE0000      ;スタートアドレスの設定
CHIPERASE:
ld      (0xFE0AAA), 0xAA      ;1st Bus Write Cycle
ld      (0xFE0554), 0x55      ;2nd Bus Write Cycle
ld      (0xFE0AAA), 0x80      ;3rd Bus Write Cycle
ld      (0xFE0AAA), 0xAA      ;4th Bus Write Cycle
ld      (0xFE0554), 0x55      ;5th Bus Write Cycle
ld      (0xFE0AAA), 0x10      ;6th Bus Write Cycle

cal     TOGGLECHK          ;トグルビットの確認

CHIPERASE_LOOP:
ld      WA, (XIX+)          ;フラッシュメモリからデータ読み込み
cp      WA, 0xFFFF          ;ブランクデータか?
j       ne, CHIPERASE_ERR    ;ブランクデータ以外の場合はエラー処理へジャンプ
cp      XIX, 0xFFFFFFFF      ;エンドアドレス(0xFFFFFFFF)か?
j       j, CHIPERASE_LOOP    ;全メモリエリア確認後にループ処理を終了

##### フラッシュメモリ書き込み処理 #####
ld      XIX, 0xFE0000      ;書き込みアドレスの設定
ld      WA, 0x0706          ;書き込みデータの設定
PROGRAM:
ld      (0xFE0AAA), 0xAA      ;1st Bus Write Cycle
ld      (0xFE0554), 0x55      ;2nd Bus Write Cycle
ld      (0xFE0AAA), 0xA0      ;3rd Bus Write Cycle
ld      (XIX), WA            ;4th Bus Write Cycle

cal     TOGGLECHK          ;トグルビットの確認

ld      BC, (XIX)           ;フラッシュメモリからデータ読み込み
cp      WA, BC              ;書き込んだデータが読み込めなければエラー
j       ne, PROGRAM_ERR     ;フラッシュメモリからデータ読み込み
ld      BC, (XIX)           ;フラッシュメモリからデータ読み込み
cp      WA, BC              ;書き込んだデータが読み込めなければエラー
j       ne, PROGRAM_ERR     ;書き込んだデータが読み込めなければエラー

PROGRAM_END:
j       j, PROGRAM_END      ;書き込み完了

##### トグルビット(D6)確認処理 #####
TOGGLECHK:
ld      L, (XIX)            ;トグルビット(D6)の確認
and     L, 0y01000000        ;1回目のトグルビットデータを保存
ld      H, L

TOGGLECHK1:
ld      L, (XIX)            ;トグルビット(D6)の確認
and     L, 0y01000000        ;トグルビット=トグル?
cp      L, H                ;トグルビットが前回の状態と同じ場合は処理を終了
j       z, TOGGLECHK2       ; 今回の状態を保存
ld      H, L                ; 再チェック
j       j, TOGGLECHK1

TOGGLECHK2:
ret

##### エラー処理 #####
CHIPERASE_ERR:
j       j, CHIPERASE_ERR     ;チップイレースエラー

PROGRAM_ERR:
j       j, PROGRAM_ERR       ;書き込みエラー

```

(プログラム例：RAMに展開するプログラム)

FF0000H~FF0FFFHのセクタイレースを実行した後、FF0000Hに0706Hのデータを書き込む

```

##### フラッシュメモリセクタイレース処理 #####
ld      XIX, 0xFF0000      ;スタートアドレスの設定
SECTORERASE:
ld      (0xFE0AAA), 0xAA      ;1st Bus Write Cycle
ld      (0xFE0554), 0x55      ;2nd Bus Write Cycle
ld      (0xFE0AAA), 0x80      ;3rd Bus Write Cycle
ld      (0xFE0AAA), 0xAA      ;4th Bus Write Cycle
ld      (0xFE0554), 0x55      ;5th Bus Write Cycle
ld      (XIX), 0x30          ;6th Bus Write Cycle

cal      TOGGLECHK          ;トグルビットの確認

SECTORERASE_LOOP:
ld      WA, (XIX+)          ;フラッシュメモリからデータ読み込み
cp      WA, 0xFFFF          ;ブランクデータか?
j       ne, SECTORERASE_ERR ;ブランクデータ以外の場合はエラー処理へジャンプ
cp      XIX, 0xFF0FFF        ;エンドアドレス(0xFF0FFF)か?
j       j, SECTORERASE_LOOP ;該当セクタエリア確認後にループ処理を終了

##### フラッシュメモリ書き込み処理 #####
ld      XIX, 0xFF0000      ;書き込みアドレスの設定
ld      WA, 0x0706          ;書き込みデータの設定
PROGRAM:
ld      (0xFE0AAA), 0xAA      ;1st Bus Write Cycle
ld      (0xFE0554), 0x55      ;2nd Bus Write Cycle
ld      (0xFE0AAA), 0xA0      ;3rd Bus Write Cycle
ld      (XIX), WA            ;4th Bus Write Cycle

cal      TOGGLECHK          ;トグルビットの確認

ld      BC, (XIX)           ;フラッシュメモリからデータ読み込み
cp      WA, BC              ;書き込んだデータが読み込めなければエラー
j       ne, PROGRAM_ERR     ;フラッシュメモリからデータ読み込み
ld      BC, (XIX)           ;フラッシュメモリからデータ読み込み
cp      WA, BC              ;書き込んだデータが読み込めなければエラー
j       ne, PROGRAM_ERR     ;書き込んだデータが読み込めなければエラー

PROGRAM_END:
j       PROGRAM_END         ;書き込み完了

##### トグルビット(D6)確認処理 #####
TOGGLECHK:
ld      L, (XIX)
and     L, 0y01000000        ;トグルビット(D6)の確認
ld      H, L                 ;1回目のトグルビットデータを保存

TOGGLECHK1:
ld      L, (XIX)
and     L, 0y01000000        ;トグルビット(D6)の確認
cp      L, H                 ;トグルビット=トグル?
j       z, TOGGLECHK2       ;トグルビットが前回の状態と同じ場合は処理を終了
ld      H, L                 ; 今回の状態を保存
j       j, TOGGLECHK1       ; 再チェック

TOGGLECHK2:
ret

##### エラー処理 #####
SECTORERASE_ERR:
j       SECTORERASE_ERR     ;セクタイレースエラー

PROGRAM_ERR:
j       PROGRAM_ERR         ;書き込みエラー

```

(プログラム例 : RAMに展開するプログラム)  
 リードプロテクトとライトプロテクトの両方を設定する

```

##### フラッシュメモリプロテクト設定処理 #####
ld      XIX, 0xFE077E          ;プロテクトアドレスの設定
PROTECT:
ld      (0xFE0AAA), 0xAA      ;1st Bus Write Cycle
ld      (0xFE0554), 0x55      ;2nd Bus Write Cycle
ld      (0xFE0AAA), 0xA5      ;3rd Bus Write Cycle
ld      (XIX), 0x00           ;4th Bus Write Cycle

cal     TOGGLECHK             ;トグルビットの確認
cal     PID_ENTRY             ;
ld      A, (XIX)              ;プロテクトアドレスリード
cal     PID_EXIT              ;
cp      A, 0x00               ;(0xFE077E)=0x00?
j       ne, PROTECT_ERR       ;プロテクト状態?

PROTECT_END:
j       PROTECT_END           ;プロテクト設定完了

PROTECT_ERR:
j       PROTECT_ERR           ;プロテクト設定エラー

##### プロダクト ID Entry 処理 #####
PID_ENTRY:
ld      (0xFE0AAA), 0xAA      ;1st Bus Write Cycle
ld      (0xFE0554), 0x55      ;2nd Bus Write Cycle
ld      (0xFE0AAA), 0x90      ;3rd Bus Write Cycle
; --- 300nSEC 以上ウェイト(NOP 命令[148nsec/@fPPH=27MHz]を 3 回実行) ---
nop
nop
nop                               ;444nSEC ウェイト
ret

##### プロダクト ID Exit 処理 #####
PID_EXIT:
ld      (0xFE0000), 0xF0      ;1st Bus Write Cycle
; --- 300nSEC 以上ウェイト(NOP 命令[148nsec/@fPPH=27MHz]を 3 回実行) ---
nop
nop
nop                               ;444nSEC ウェイト
ret

##### トグルビット(D6)確認処理 #####
TOGGLECHK:
ld      L, (XIX)
and     L, 0y01000000         ;トグルビット(D6)の確認
ld      H, L                  ;1 回目のトグルビットデータを保存
TOGGLECHK1:
ld      L, (XIX)
and     L, 0y01000000         ;トグルビット(D6)の確認
cp      L, H                  ;トグルビット=トグル?
j       z, TOGGLECHK2         ;トグルビットが前回の状態と同じ場合は処理を終了
ld      H, L                  ; 今回の状態を保存
j       TOGGLECHK1           ; 再チェック
TOGGLECHK2:
ret

```

(プログラム例 : RAMに展開するプログラム)  
 FE0000H からデータを読み込む

```

##### フラッシュメモリ読み込み処理 #####
READ:
ld      WA, (0xFE0000)        ;フラッシュメモリからデータ読み込み

```

## 4. 電気的特性

### 4.1 絶対最大定格

項目	記号	定格	単位
電源電圧	V <sub>CC</sub>	-0.5~4.0	V
入力電圧	V <sub>IN</sub>	-0.5~V <sub>CC</sub> + 0.5	V
出力電流 (1 端子当たり)	I <sub>OL</sub>	2	mA
出力電流 (1 端子当たり)	I <sub>OH</sub>	-2	mA
出力電流 (合計)	Σ I <sub>OL</sub>	80	mA
出力電流 (合計)	Σ I <sub>OH</sub>	-80	mA
消費電力 (Ta = 85°C)	PD	600	mW
はんだ付け温度 (10 s)	TSOLDER	260	°C
保存温度	TSTG	-65~150	°C
動作温度	TOPR	-40~85	°C
プログラム消去回数	N <sub>EW</sub>	100	Cycle

注) 絶対最大定格とは、瞬時たりとも超えてはならない規格であり、どの1つの項目も超えることができない規格です。絶対最大定格を超えると、破壊や劣化の原因となり、破裂・燃焼による傷害を負うことがあります。従って、必ず絶対最大定格を超えないように、応用機器の設計を行ってください。

#### 鉛フリー品 (G 付製品) へのはんだ濡れ性についての注意事項

試験項目	試験条件	備考
はんだ付け性	230°C 5 秒間 1 回 R タイプフラックス使用 (鉛はんだ使用時) 245°C 5 秒間 1 回 R タイプフラックス使用 (鉛フリーはんだ使用時)	フォーミングまでの半田付着率 95% を良品とする

## 4.2 DC 電気的特性 (1/2)

項目		記号	条件		Min	Typ. 注)	Max	単位
電源電圧 ( AVCC = DVCC AVSS = DVSS = 0 V )		VCC	fc = 4~27 MHz	fs = 30~ 34 kHz	2.7		3.6	V
			fc = 2~16 MHz		2.2			
電源電圧 ( AVCC = DVCC AVS = DVSS = 0 V ) Flash へのプログラムおよび 消去動作時		VCC	fc = 4~27 MHz Ta = -10~40°C		2.7		3.6	V
低レベル 入力電圧	P00~P17 (AD0~AD15)	VIL	Vcc ≥ 2.7 V		-0.3		0.6	V
			Vcc < 2.7 V				0.2 Vcc	
	P20~P97 (P63 を除く)	VIL1	Vcc ≥ 2.7 V				0.3 Vcc	
			Vcc < 2.7 V				0.2 Vcc	
	RESET, NMI, P63 (INT0)	VIL2	Vcc ≥ 2.7 V				0.25 Vcc	
			Vcc < 2.7 V				0.15 Vcc	
	AM0, AM1	VIL3	Vcc ≥ 2.7 V				0.3	
			Vcc < 2.7 V				0.3	
X1	VIL4	Vcc ≥ 2.7 V		0.2 Vcc				
		Vcc < 2.7 V		0.1 Vcc				
高レベル 入力電圧	P00~P17 (AD0~AD15)	VIH	Vcc ≥ 2.7 V		2.0	Vcc + 0.3	V	
			Vcc < 2.7 V		0.7 Vcc			
	P20~P97 (P63 を除く)	VIH1	Vcc ≥ 2.7 V		0.7 Vcc			
			Vcc < 2.7 V		0.8 Vcc			
	RESET, NMI, P63 (INT0)	VIH2	Vcc ≥ 2.7 V		0.75 Vcc			
			Vcc < 2.7 V		0.85 Vcc			
	AM0, AM1	VIH3	Vcc ≥ 2.7 V		Vcc - 0.3			
			Vcc < 2.7 V		Vcc - 0.3			
X1	VIH4	Vcc ≥ 2.7 V		0.8 Vcc				
		Vcc < 2.7 V		0.9 Vcc				
低レベル出力電圧		VOL	IOL = 1.6 mA	Vcc ≥ 2.7 V			0.45	V
			IOL = 0.4 mA	Vcc < 2.7 V			0.15 Vcc	
高レベル出力電圧		VOH	IOH = -400 μA	Vcc ≥ 2.7 V	Vcc - 0.3			V
			IOH = -200 μA	Vcc < 2.7 V	0.8 Vcc			

注) Typ.値は特に指定のない限り、Ta = 25°C, Vcc = 3.0 V です。

## DC 電気的特性 (2/2)

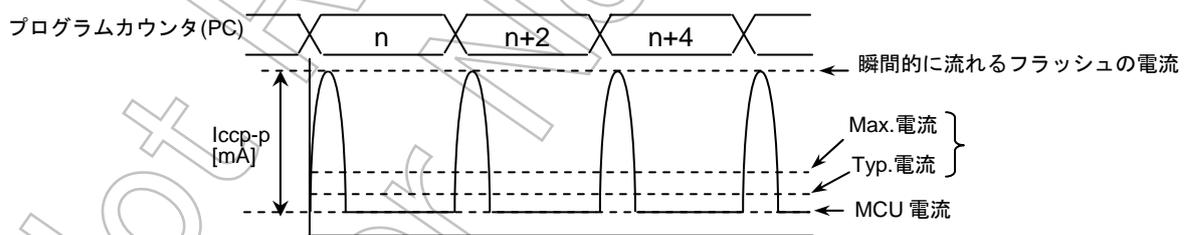
項目	記号	条件	Min	Typ.(注 1)	Max	単位		
入力リーク電流	ILI	$0.0 \leq V_{IN} \leq V_{CC}$		0.02	$\pm 5$	$\mu\text{A}$		
出力リーク電流	ILO	$0.2 \leq V_{IN} \leq V_{CC} - 0.2$		0.05	$\pm 10$			
パワーダウン電圧 (@STOP, RAMバックアップ)	VSTOP	$V_{IL2} = 0.2 V_{CC}$ , $V_{IH2} = 0.8 V_{CC}$	2.2		3.6	V		
RESET プルアップ抵抗	RRST	$V_{CC} = 2.7 \text{ V} \sim 3.6 \text{ V}$	100		400	$\text{k}\Omega$		
		$V_{CC} = 2.2 \text{ V}$	200		1000			
端子容量	CIO	$f_c = 1 \text{ MHz}$			10	pF		
シュミット幅 RESET, NMI, INTO	VTH	$V_{CC} \geq 2.7 \text{ V}$	0.4	1.0		V		
		$V_{CC} < 2.7 \text{ V}$	0.3	0.8				
プログラマブル プルアップ抵抗	RKH	$V_{CC} = 2.7 \text{ V} \sim 3.6 \text{ V}$	100		400	$\text{k}\Omega$		
		$V_{CC} = 2.2 \text{ V}$	200		1000			
NORMAL (注 2)	I <sub>CC</sub>	$V_{CC} = 2.7 \text{ V} \sim 3.6 \text{ V}$ $f_c = 27 \text{ MHz}$		13	20	mA		
IDLE2				7	9			
IDLE1				3.4	4.5			
NORMAL (注 2)			$V_{CC} = 2.2 \text{ V}$ (Typ.値は 2.2V) $f_c = 16 \text{ MHz}$		5.5		8	mA
IDLE2					3.0		4.8	
IDLE1					1.5		2.9	
SLOW (注 2)		$V_{CC} = 2.2 \text{ V} \sim 3.6 \text{ V}$ $f_s = 32.768 \text{ kHz}$		20	55	$\mu\text{A}$		
IDLE2				10	44			
IDLE1				6	40			
STOP				1	25			
間欠動作ピーク電流		I <sub>CCP-p</sub>	$V_{CC} = 2.2 \text{ V} \sim 3.6 \text{ V}$		20		mA	

注 1) Typ.値は特に指定のない限り、 $T_a = 25^\circ\text{C}$ ,  $V_{CC} = 3.0 \text{ V}$  です。

注 2) I<sub>CC</sub> NORMAL, SLOW の測定条件: すべて動作、出力端子は開放、入力端子はレベル固定。

フラッシュメモリでプログラムが動作しているとき、またはフラッシュメモリからデータをリードしているとき、フラッシュメモリは間欠動作を行いますので、瞬間的に下記タイミング図のようなピーク電流が流れます。よってこの場合の電源電流 I<sub>CC</sub>(NORMAL/SLOW モード時)は、ピーク電流を平均化した電流値と MCU 電流値の和となります。

電源設計の際はピーク電流が供給可能な回路設計にしてください。SLOW モードではピーク電流と平均化された電流の差が大きくなります。



フラッシュメモリの間欠動作

## 4.3 AC 電気的特性

(1)  $V_{CC} = 2.7\text{ V} \sim 3.6\text{ V}$ 

No.	項目	記号	計算式		$f_{\text{FPH}} = 27\text{ MHz}$		単位
			Min	Max	Min	Max	
1	$f_{\text{FPH}}$ 周期 (= x)	$t_{\text{FPH}}$	37.0	31250	37.0		ns
2	A0~A15 有効 → ALE 立ち下がり	$t_{\text{AL}}$	$0.5x - 6$		12		ns
3	ALE 立ち下がり → A0~A15 保持	$t_{\text{LA}}$	$0.5x - 16$		2		ns
4	ALE 高パルス幅	$t_{\text{LL}}$	$x - 20$		17		ns
5	ALE 立ち下がり → $\overline{\text{RD}} / \overline{\text{WR}}$ 立ち下がり	$t_{\text{LC}}$	$0.5x - 14$		4		ns
6	$\overline{\text{RD}}$ 立ち上がり → ALE 立ち上がり	$t_{\text{CLR}}$	$0.5x - 10$		8		ns
7	$\overline{\text{WR}}$ 立ち上がり → ALE 立ち上がり	$t_{\text{CLW}}$	$x - 10$		27		ns
8	A0~A15 有効 → $\overline{\text{RD}} / \overline{\text{WR}}$ 立ち下がり	$t_{\text{ACL}}$	$x - 23$		14		ns
9	A0~A21 有効 → $\overline{\text{RD}} / \overline{\text{WR}}$ 立ち下がり	$t_{\text{ACH}}$	$1.5x - 26$		29		ns
10	$\overline{\text{RD}}$ 立ち上がり → A0~A21 保持	$t_{\text{CAR}}$	$0.5x - 13$		5		ns
11	$\overline{\text{WR}}$ 立ち上がり → A0~A21 保持	$t_{\text{CAW}}$	$x - 13$		24		ns
12	A0~A15 有効 → D0~D15 入力	$t_{\text{ADL}}$		$3.0x - 38$		73	ns
13	A0~A21 有効 → D0~D15 入力	$t_{\text{ADH}}$		$3.5x - 41$		88	ns
14	$\overline{\text{RD}}$ 立ち下がり → D0~D15 入力	$t_{\text{RD}}$		$2.0x - 30$		44	ns
15	$\overline{\text{RD}}$ 低パルス幅	$t_{\text{RR}}$	$2.0x - 15$		59		ns
16	$\overline{\text{RD}}$ 立ち上がり → D0~D15 保持	$t_{\text{HR}}$	0		0		ns
17	$\overline{\text{RD}}$ 立ち上がり → A0~A15 出力	$t_{\text{RAE}}$	$x - 15$		22		ns
18	$\overline{\text{WR}}$ 低パルス幅	$t_{\text{WW}}$	$1.5x - 15$		40		ns
19	D0~D15 有効 → $\overline{\text{WR}}$ 立ち上がり	$t_{\text{DW}}$	$1.5x - 35$		20		ns
20	$\overline{\text{WR}}$ 立ち上がり → D0~D15 保持	$t_{\text{WD}}$	$x - 25$		12		ns
21	A0~A21 有効 → ポート入力	$t_{\text{APH}}$		$3.5x - 89$		40	ns
22	A0~A21 有効 → ポート保持	$t_{\text{APH2}}$	$3.5x$		129		ns
23	A0~A21 有効 → ポート有効	$t_{\text{AP}}$		$3.5x + 80$		209	ns

## AC 測定条件

- 出力レベル: High  $0.7 \times V_{CC}$  / Low  $0.3 \times V_{CC}$ , CL = 50 pF
- 入力レベル: High  $0.9 \times V_{CC}$  / Low  $0.1 \times V_{CC}$

注) 表中の「x」は、クロック  $f_{\text{FPH}}$  の周期を示します。 $f_{\text{FPH}}$  の周期は、CPU コアで使用されるシステムクロック  $f_{\text{SYS}}$  周期の 1/2 です。

クロック  $f_{\text{FPH}}$  の周期は、クロックギアの設定や、高速発振器/低速発振器の切り替えなどに依存します。

(2)  $V_{CC} = 2.2\text{ V} \sim 3.6\text{ V}$ 

No.	項目	記号	計算式		$f_{FPH} = 16\text{ MHz}$		単位
			Min	Max	Min	Max	
1	$f_{FPH}$ 周期 (= x)	$t_{FPH}$	62.5	31250	62.5		ns
2	A0~A15 有効→ALE 立ち下がり	$t_{AL}$	$0.5x - 18$		13		ns
3	ALE 立ち下がり→A0~A15 保持	$t_{LA}$	$0.5x - 25$		6		ns
4	ALE 高パルス幅	$t_{LL}$	$x - 30$		32		ns
5	ALE 立ち下がり→ $\overline{RD}$ / $\overline{WR}$ 立ち下がり	$t_{LC}$	$0.5x - 25$		6		ns
6	$\overline{RD}$ 立ち上がり→ALE 立ち上がり	$t_{CLR}$	$0.5x - 15$		16		ns
7	$\overline{WR}$ 立ち上がり→ALE 立ち上がり	$t_{CLW}$	$x - 15$		47		ns
8	A0~A15 有効→ $\overline{RD}$ / $\overline{WR}$ 立ち下がり	$t_{ACL}$	$x - 30$		32		ns
9	A0~A21 有効→ $\overline{RD}$ / $\overline{WR}$ 立ち下がり	$t_{ACH}$	$1.5x - 30$		63		ns
10	$\overline{RD}$ 立ち上がり→A0~A21 保持	$t_{CAR}$	$0.5x - 21$		10		ns
11	$\overline{WR}$ 立ち上がり→A0~A21 保持	$t_{CAW}$	$x - 25$		37		ns
12	A0~A15 有効→D0~D15 入力	$t_{ADL}$		$3.0x - 50$		137	ns
13	A0~A21 有効→D0~D15 入力	$t_{ADH}$		$3.5x - 56$		162	ns
14	$\overline{RD}$ 立ち下がり→D0~D15 入力	$t_{RD}$		$2.0x - 50$		75	ns
15	$\overline{RD}$ 低パルス幅	$t_{RR}$	$2.0x - 28$		97		ns
16	$\overline{RD}$ 立ち上がり→D0~D15 保持	$t_{HR}$	0		0		ns
17	$\overline{RD}$ 立ち上がり→A0~A15 出力	$t_{RAE}$	$x - 25$		37		ns
18	$\overline{WR}$ 低パルス幅	$t_{WW}$	$1.5x - 29$		64		ns
19	D0~D15 有効→ $\overline{WR}$ 立ち上がり	$t_{DW}$	$1.5x - 60$		33		ns
20	$\overline{WR}$ 立ち上がり→D0~D15 保持	$t_{WD}$	$x - 40$		22		ns
21	A0~A21 有効→ポート入力	$t_{APH}$		$3.5x - 100$		68	ns
22	A0~A21 有効→ポート保持	$t_{APH2}$	$3.5x$		218		ns
23	A0~A21 有効→ポート有効	$t_{AP}$		$3.5x + 150$		368	ns

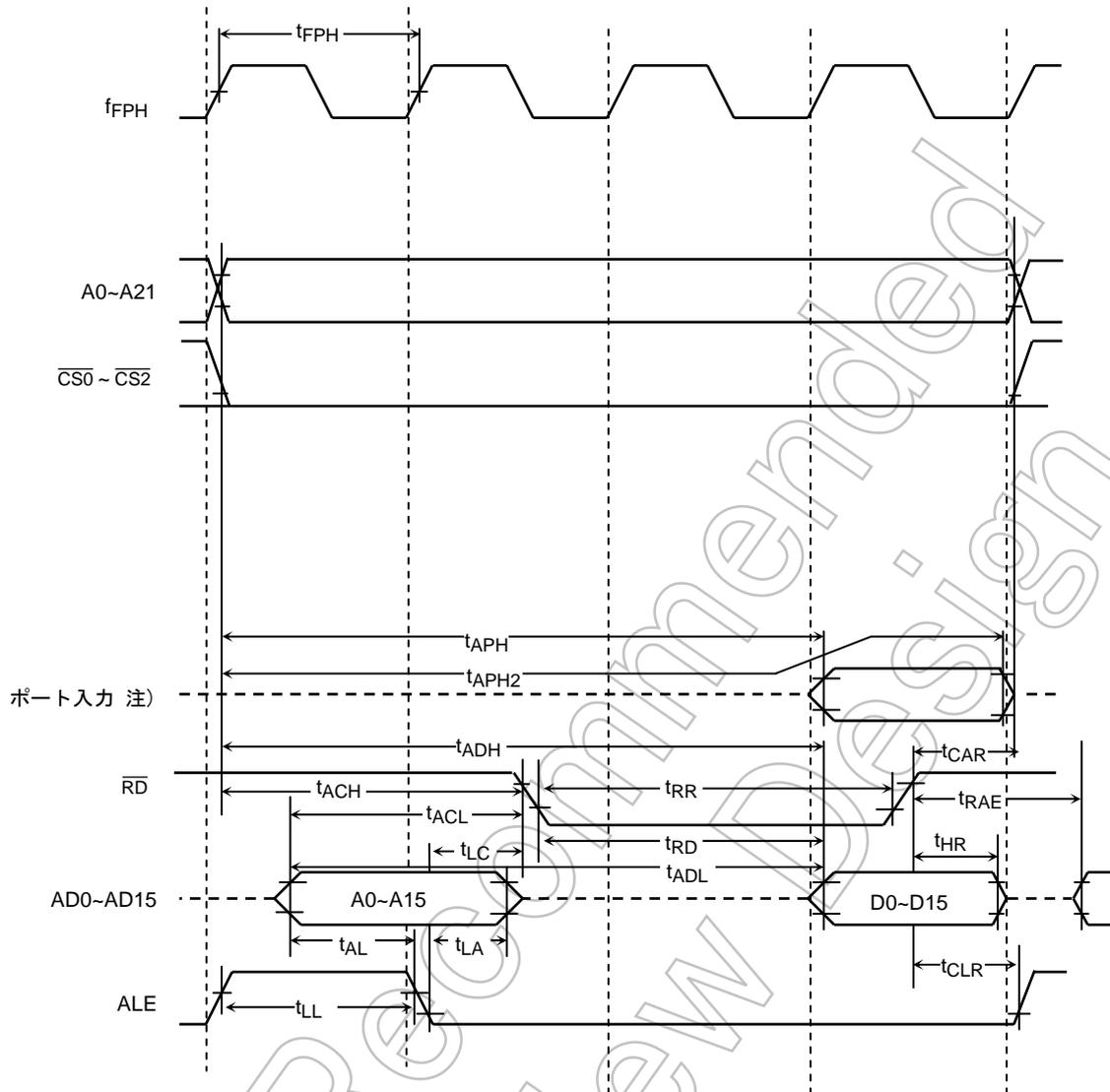
## AC 測定条件

- 出力レベル: High  $0.7 \times V_{CC}$ /Low  $0.3 \times V_{CC}$ , CL = 50 pF
- 入力レベル: High  $0.9 \times V_{CC}$ /Low  $0.1 \times V_{CC}$

注) 表中の「x」は、クロック  $f_{FPH}$  の周期を示します。 $f_{FPH}$  の周期は、CPU コアで使用されるシステムクロック  $f_{SYS}$  周期の 1/2 です。

クロック  $f_{FPH}$  の周期は、クロックギアの設定や、高速発振器/低速発振器の切り替えなどに依存します。

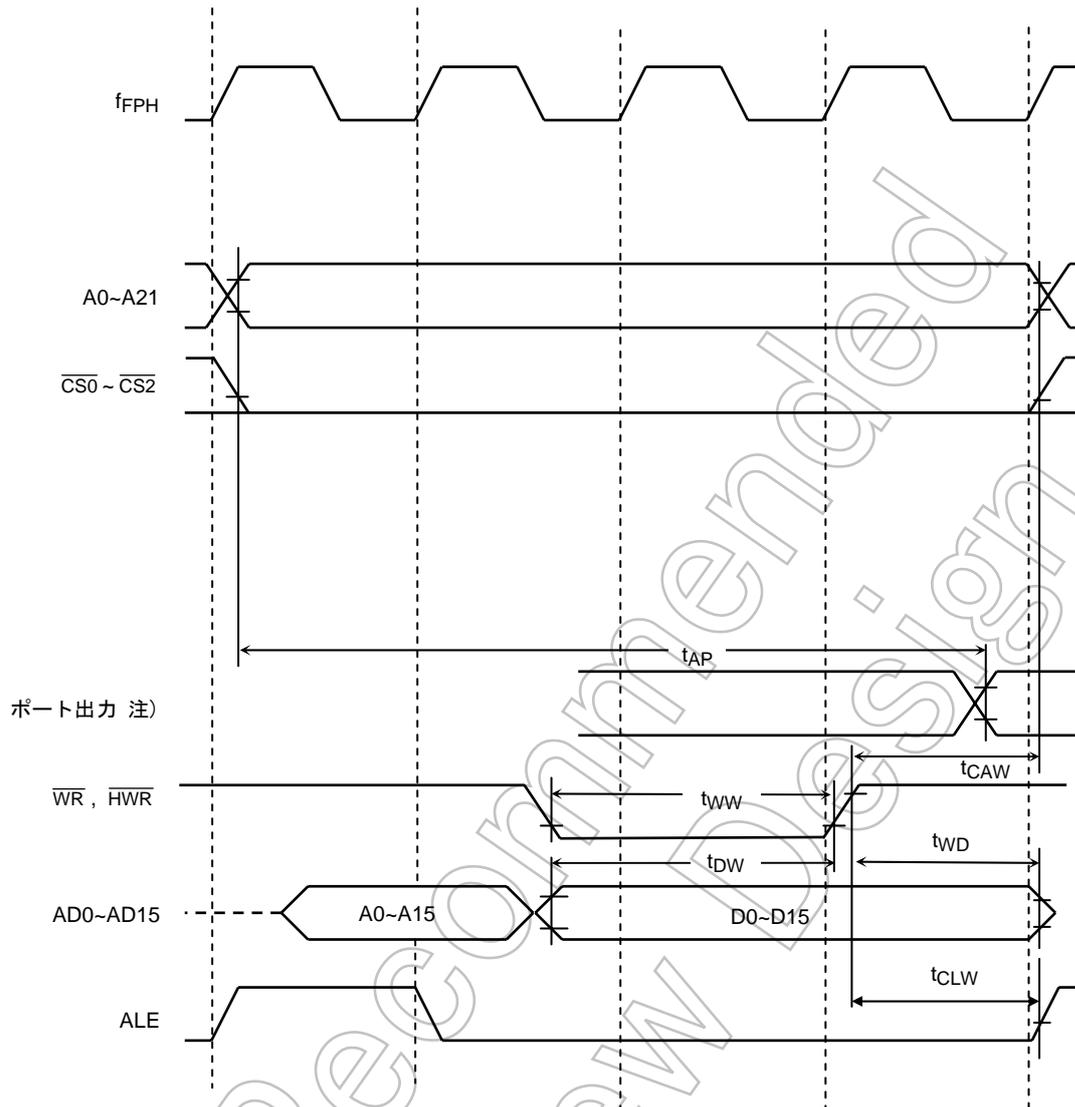
## (2) リードサイクル



注) ポートのデータリードは内蔵エリアへのアクセスとなるため、外部端子の制御信号  $\overline{RD}$ 、 $\overline{CS}$  信号などはイネーブル出力されません。上記波形は内部信号の動きとして認識してください。

また、上記ポート入出力タイミングおよび AC 特性は、代表的なタイミングを示しています。詳細は弊社営業担当までお問い合わせください。

## (3) ライトサイクル



注) ポートのデータライトは内蔵エリアへのアクセスとなるため、外部端子の制御信号  $\overline{WR}$ 、 $\overline{CS}$  信号などはインネーブル出力されません。上記波形は内部信号の動きとして認識してください。

また、上記ポート入出力タイミングおよび AC 特性は、代表的なタイミングを示しています。詳細は弊社営業担当までお問い合わせください。

## 4.4 AD 変換特性

AVCC = VCC , AVSS = VSS

項目	記号	条件	Min	Typ.	Max	単位
アナログ入力電圧	VAIN		AVSS		AVCC	V
総合誤差 (量子誤差を含まず)	-	VCC = 2.2 V~3.6 V		±1.0	±4.0	LSB

注 1) 1LSB = (AVCC - AVSS)/1024 [V]

注 2) 最低動作周波数について

AD コンバータの動作は、fc (高速発振器) 使用時のみ保証します (fs では保証しません)。ただし、fc 使用時に  
クロックギアで選択されたクロックの周波数が 4 MHz 以上で保証します。

注 3) AVCC 端子より流れる電源電流は、VCC 端子の電源電流 (Icc) に含まれます。

Not Recommended for New Design

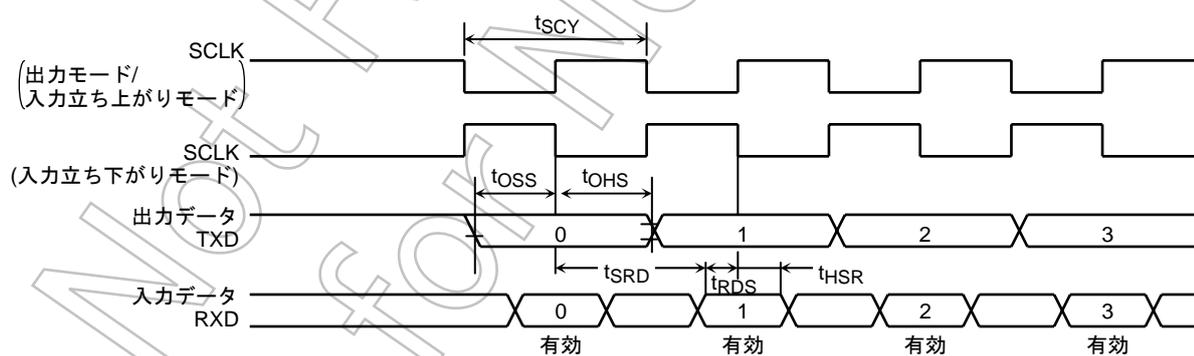
## 4.5 シリアルチャネルタイミング-I/O インタフェースモード

## (1) SCLK 入力モード

項目	記号	計算式		10 MHz		27 MHz		単位
		Min	Max	Min	Max	Min	Max	
SCLK 周期	t <sub>SCY</sub>	16X		1.6		0.59		μs
出力データ → SCLK 立ち上がり/立ち下がり	t <sub>OSS</sub>	$t_{SCY}/2 - 4X - 110$ (V <sub>CC</sub> = 2.7 V~3.6 V)		290		38		ns
		$t_{SCY}/2 - 4X - 180$ (V <sub>CC</sub> = 2.2 V)		220		-		
SCLK 立ち上がり/立ち下がり → 出力データ 保持	t <sub>OHS</sub>	$t_{SCY}/2 + 2X + 0$		1000		370		ns
SCLK 立ち上がり/立ち下がり → 入力データ 保持	t <sub>HSR</sub>	3X + 10		310		121		ns
SCLK 立ち上がり/立ち下がり → 有効データ入力	t <sub>SRD</sub>				1600		592	ns
有効データ入力 → SCLK 立ち上がり/立ち下がり	t <sub>RDS</sub>	0		0		0		ns

## (2) SCLK 出力モード

項目	記号	計算式		10 MHz		27 MHz		単位
		Min	Max	Min	Max	Min	Max	
SCLK 周期	t <sub>SCY</sub>	16X	8192X	1.6	819	0.59	303	μs
出力データ → SCLK 立ち上がり/立ち下がり	t <sub>OSS</sub>	$t_{SCY}/2 - 40$		760		256		ns
SCLK 立ち上がり/立ち下がり → 出力データ 保持	t <sub>OHS</sub>	$t_{SCY}/2 - 40$		760		256		ns
SCLK 立ち上がり/立ち下がり → 入力データ 保持	t <sub>HSR</sub>	0		0		0		ns
SCLK 立ち上がり/立ち下がり → 有効データ入力	t <sub>SRD</sub>	$t_{SCY} - 1X - 180$			1320		375	ns
有効データ入力 → SCLK 立ち上がり/立ち下がり	t <sub>RDS</sub>	1X + 180		280		217		ns



注 1) SCLK 立ち上がり/立ち下がり…SCLK 立ち上がりモードの場合は SCLK 立ち上がり、SCLK 立ち下がりモードの場合は SCLK 立ち下がりモードのタイミングです。

注 2) 27 MHz, 10 MHz の値は、t<sub>SCY</sub> = 16X のときの値です。

注 3) 表中の「x」は、クロック f<sub>FPH</sub> の周期を示します。f<sub>FPH</sub> の周期は、CPU コアで使用されるシステムクロック f<sub>sys</sub> 周期の 1/2 です。

クロック f<sub>FPH</sub> の周期は、クロックギアの設定や、高速発振器/低速発振器の切り替えなどに依存します。

## 4.6 イベントカウンタ (TA0IN, TA4IN, TB0IN0, TB0IN1)

項目	記号	計算式		10 MHz		27 MHz		単位
		Min	Max	Min	Max	Min	Max	
クロック周期	t <sub>VCK</sub>	8X + 100		900		396		ns
クロック低レベルパルス幅	t <sub>VCKL</sub>	4X + 40		440		188		ns
クロック高レベルパルス幅	t <sub>VCKH</sub>	4X + 40		440		188		ns

## 4.7 割り込み、キャプチャ

(1)  $\overline{\text{NMI}}$ , INTO 割り込み

項目	記号	計算式		10 MHz		27 MHz		単位
		Min	Max	Min	Max	Min	Max	
$\overline{\text{NMI}}$ , INTO 低レベルパルス幅	t <sub>INTAL</sub>	4X + 40		440		188		ns
$\overline{\text{NMI}}$ , INTO 高レベルパルス幅	t <sub>INTAH</sub>	4X + 40		440		188		ns

## (2) INT5~INT6 割り込み、キャプチャ

INT5~INT6 入力パルス幅はシステムクロック選択、およびプリスケラ用クロック選択により異なります。下記に動作クロック別パルス幅を示します。

システム クロック選択 SYSCR1 <SYSCK>	プリスケラ用 クロック選択 SYSCR0 <PRCK1:0>	t <sub>INTBL</sub> (INT5~INT6 低レベルパルス幅)		t <sub>INTBH</sub> (INT5~INT6 高レベルパルス幅)		単位
		計算式	f <sub>FPH</sub> = 27MHz	計算式	f <sub>FPH</sub> = 27MHz	
		Min	Min	Min	Min	
0 (fc)	00 (f <sub>FPH</sub> )	8X + 100	396	8X + 100	396	ns
	10 (fc/16)	128Xc + 0.1	4.8	128Xc + 0.1	4.8	μs
1 (fs)	00 (f <sub>FPH</sub> )	8X + 0.1	244.3	8X + 0.1	244.3	

注 1) Xc は、高速発振器側のクロック fc の周期を示します。

注 2) 表中の「x」は、クロック f<sub>FPH</sub> の周期を示します。f<sub>FPH</sub> の周期は、CPU コアで使用されるシステムクロック f<sub>sys</sub> 周期の 1/2 です。

クロック f<sub>FPH</sub> の周期は、クロックギアの設定や、高速発振器/低速発振器の切り替えなどに依存します。

## 4.8 フラッシュ特性

## (1) 書き込み特性

項目	条件	Min	Typ	Max	単位
フラッシュメモリ書き替え保証回数	V <sub>cc</sub> = 2.7V~3.6V fc = 4~27 MHz Ta = -10~40°C	-	-	100	回

## 5. ポート部等価回路図

- 回路図の見方

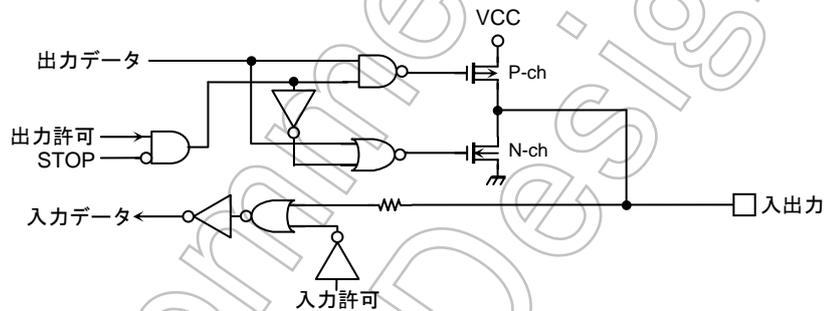
基本的に、標準 CMOS ロジック IC 「74HCxx」 シリーズと同じゲート記号を使って書かれています。

信号名の中で特殊なものについては、下記に示します。

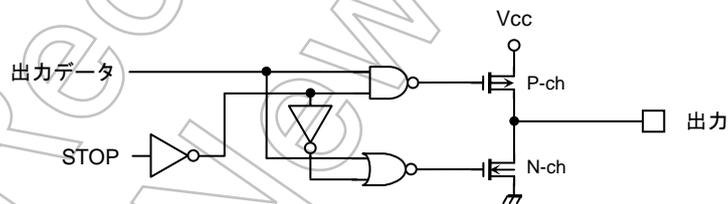
**STOP:** この信号は、HALT モード設定レジスタを「STOP」モード (SYSCR2<HALTM1:0> = 0, 1) にして、CPU が「HALT」命令を実行したとき、アクティブ“1”になります。ただし、ドライブイネーブルビット SYSCR2<DRVE> が“1”にセットされているとき、STOP は“0”のままです。

- 入力保護抵抗は、数十Ω~数百Ω程度です。

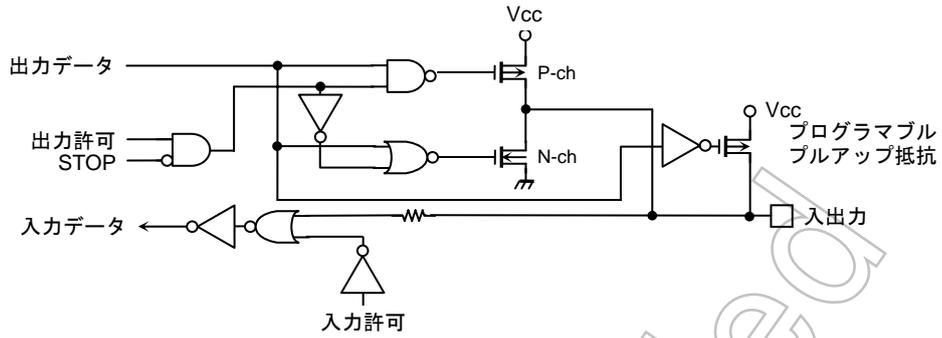
- P0 (AD0~AD7), P1 (AD8~AD15, A8~A15), P2 (A16~A21, A0~A5), P60, P70~P74, P80~P83, P91~P92, P94~P95



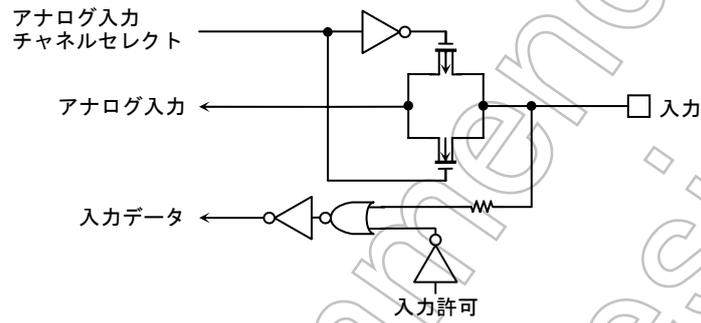
- P30 ( $\overline{RD}$ ), P31 ( $\overline{WR}$ )



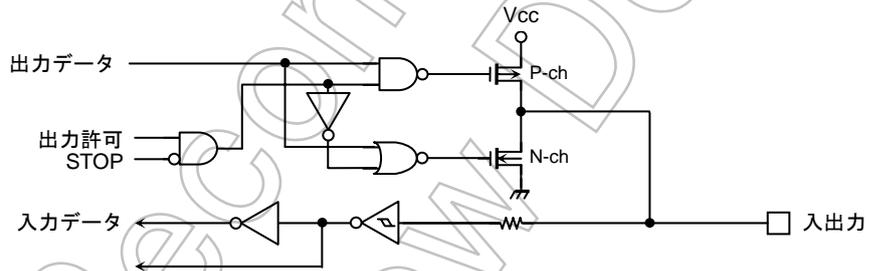
■ P32, P40~P42



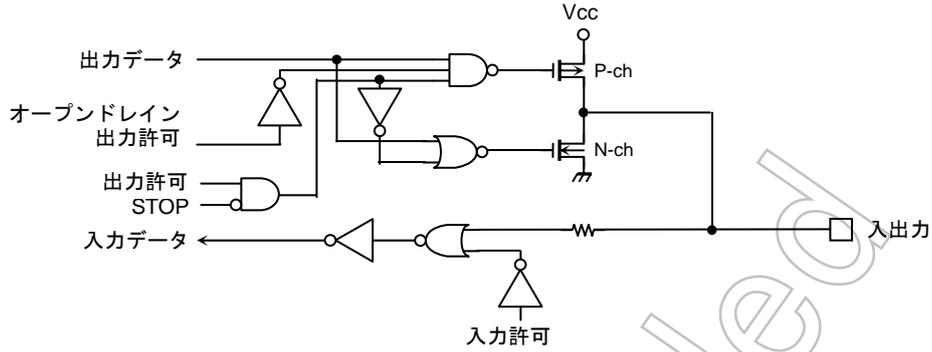
■ P5 (AN0~AN3)



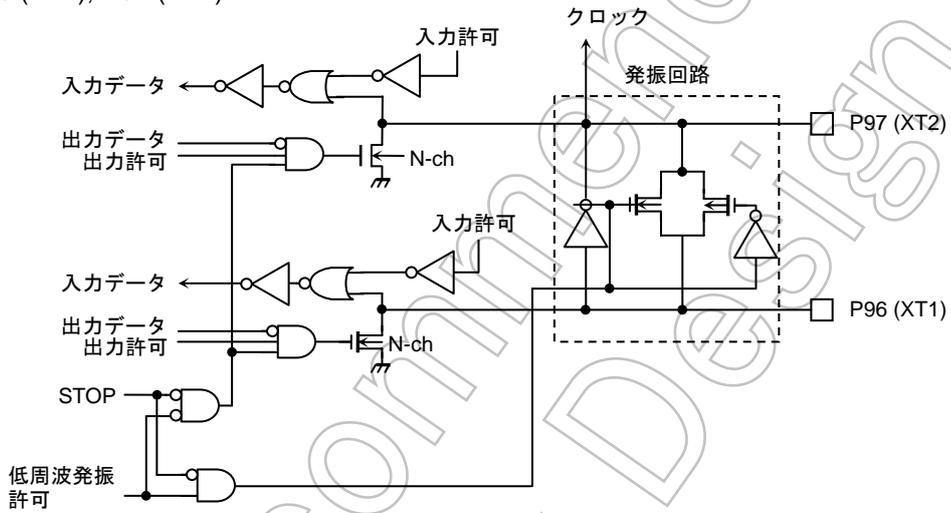
■ P63 (INT0)



■ P61 (SO/SDA), P62 (SI/SCL), P90 (TXD0), P93 (TXD1)



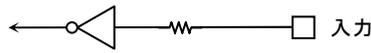
■ P96 (XT1), P97 (XT2)



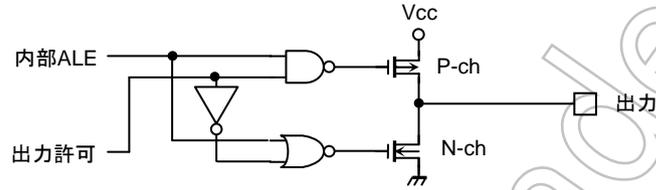
■ NMI



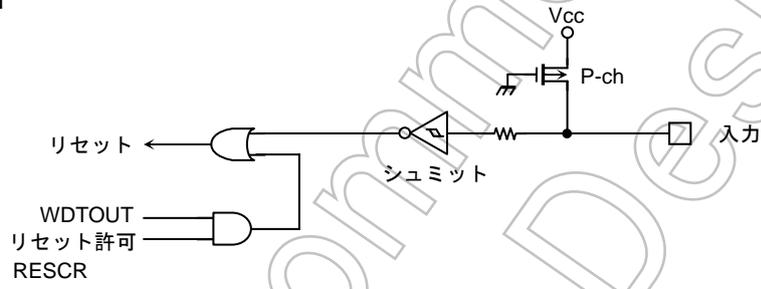
■ AM0~AM1



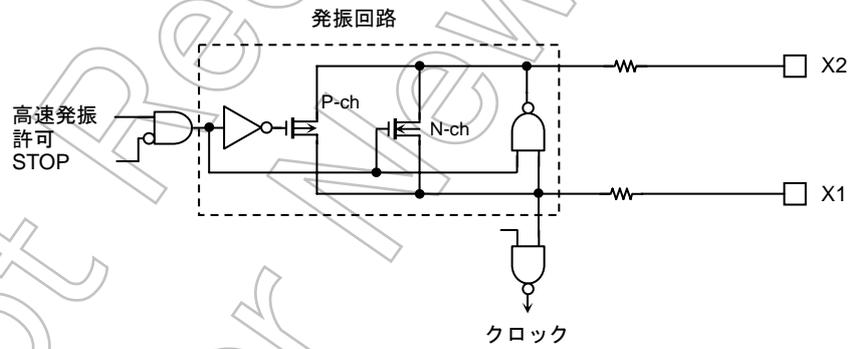
■ ALE



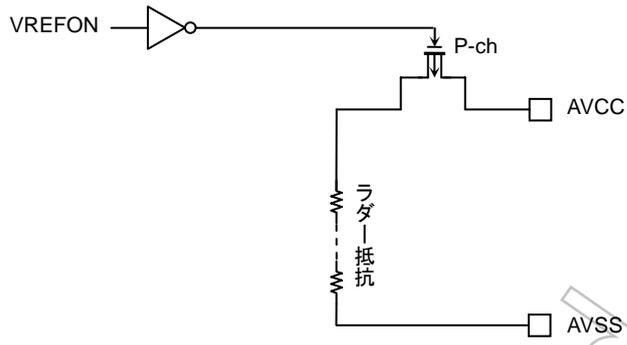
■  $\overline{\text{RESET}}$



■ X1, X2



■ AVCC, AVSS

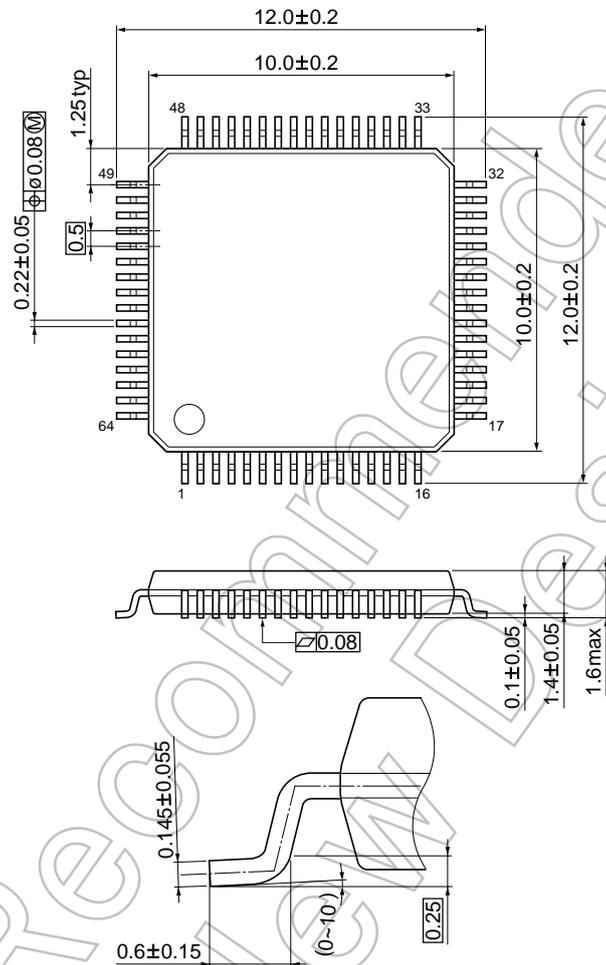


Not Recommended  
for New Design

## 6. パッケージ外形寸法図

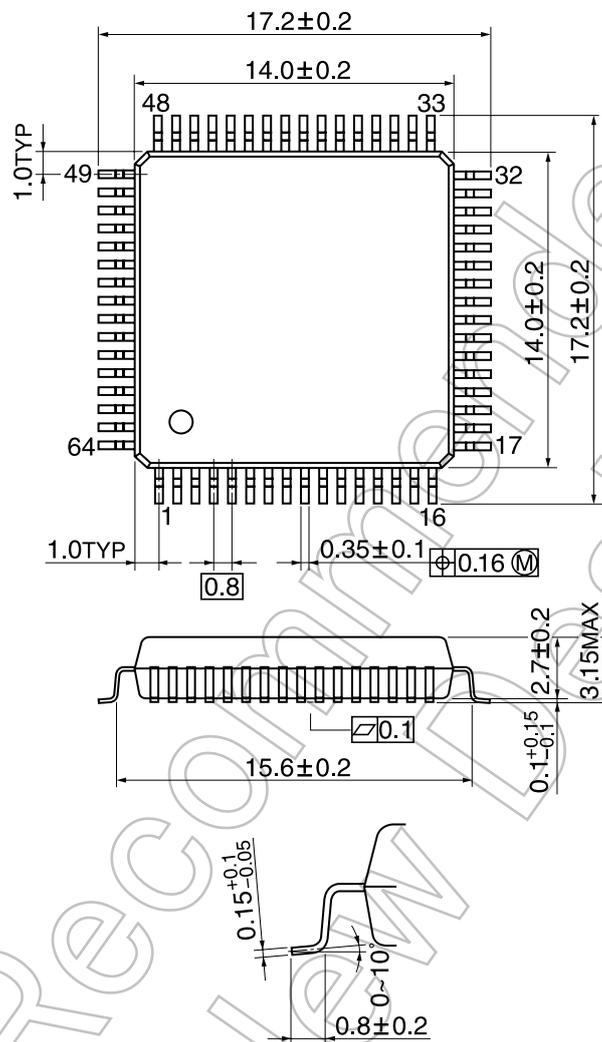
パッケージ名: LQFP64-P-1010-0.50D

Unit: mm



パッケージ名: QFP64-P-1414-0.80A

Unit: mm



Not Recommended for New Design