

AVR1305 : XMEGAの割り込みと 設定可能な多段割り込み制御器

要点

- 3つの割り込みレベル
- 低レベル割り込みに対するラウトロビン計画
- 低レベル割り込みに対する設定可能な優先権

1. 序説

マイクロコントローラはタスク間の優先付けと或る周辺機能部署が高速に処理されるのを保証するために割り込みを使用します。更に割り込みはマイクロコントローラの消費電力を低減するのに使用することができます。それでデバイスは或る割り込みを引き起こす条件が起きるまで低電力動作です。

XMEGA[®]の割り込み機構と設定可能な多段割り込み制御器(PMIC)が本応用記述で記述されます。本応用記述はPMICがどうアクセスされ得るかを示すコード例も提供します。

2. 動作の理屈

殆どの(周辺機能)部署は状態フラグまたは割り込みの実行を起動できる割り込み要求フラグを持っています。割り込みはプログラムコードの流れが割り込まれて、特定の割り込み処理ルーチン(ISR)プログラムコードが実行されるべきことを部署がAVR[®] CPUに合図する時です。割り込みは次の必要条件の全てに出会う場合に生成されます。

- ・ 割り込み要求フラグを設定(1)する条件が全て満たされている。
- ・ 指定割り込みが部署内で許可されている。
- ・ 割り込みレベルが割り込み制御器で許可されている。
- ・ 全体割り込みがCPUに対して許可されている。

割り込みの実行順序は先にそれらの割り込みレベルによって、そしてその後にそれらの割り込み優先権によって決められます。割り込みレベルはそれらが源を発する部署内の割り込みに対して個別に指定され、一方割り込み優先権はそれらの固定の割り込みベクタアドレスによって決められます。代わりに、低レベル割り込みに対する割り込み優先権は割り込み制御器によって制御されるラウトロビンの仕組みに基くことができます。

遮蔽不可割り込み(NMI)は禁止することができない特別な割り込みで、システムの重要な割り込みに関して使用されます。NMIはクリスタル用発振器の停止のような重要な不具合に対して使用されます。

2.1. 割り込みレベル - 部署割り込み許可

周辺機能部署が割り込みの使用を必要とするとき、対応する割り込みの割り込みレベルを"OFF"以外の何かに設定します。割り込みは、低、中、高の3つのレベルの1つを生じることができます。割り込みのレベルと許可を制御するビットは代表的にINTCTRLレジスタに配置されています。

複数の割り込みレベルを持つとき、与えられたレベルでの割り込み実行時に、より高いレベルの割り込みによって割り込むことができると理解すべきです。これは低割り込みレベルの割り込みが中または高レベルの割り込みによって割り込まれることを意味します。明らかに、入れ子にされた割り込みが起き得ることは、ファームウェアによって必要とされるスタックの深さを査定する時にこれを考慮に入れるべきです。

また、8ビットより多いレジスタと変数へのアクセスの保護も考慮しなければなりません。AVRのデータバスが8ビット幅で複数バイトの変数へのアクセスが1つより多くの命令実行を必要とするため、16ビット(または32ビット)変数の先頭と第2バイト間に割り込みが起き得ます。コードの主繰り返しで使用されている16ビット変数が割り込みによって変更される場合、その変数が不正にされ得ます。同じ状態は割り込みがより高いレベルの別の割り込みによって割り込まれる場合にも適用します。この不正は多くの方法で避けることができます。1つの方法は割り込み処理ルーチン(ISR)によって変更され得る変数が非分断操作でアクセスされるのを保証することです。これは変数をアクセスする間、全体割り込みの禁止を通して達成することによって行えます。別の方法、それは16ビットレジスタ(または、むしろ共にで16ビット値を表す2つの8ビットレジスタ)のアクセス時に適用し、これらのレジスタのアクセス時にハードウェアによって使用される一時レジスタで保存と復元を行うことです。この方法は例えば16ビットタイマレジスタのアクセス時に適切となり得ます。一時レジスタとレジスタへの16ビットアクセスについてのより多くの情報はデータシートで得られます。



8ビット AVR[®]
マイクロコントローラ

応用記述

本書は一般の方々の便宜のため有志により作成されたもので、ATMEL社とは無関係であることを御承知ください。しおりのはじめにでの内容にご注意ください。

Rev. 8043A-02/08, 8043AJ3-03/14

2.2. 遮蔽不可割り込み(NMI)

XMEGA A1デバイスには1つのNMI元、クリスタル用発振器停止NMIを持っています。

クリスタルがシステムクロックとして使用される場合、クリスタル用発振器からのクロックは監視されます。いくつかの理由でクリスタルクロックが停止(例えば、クリスタルへの物理的な損傷のため)する場合、NMIが生成されて、システムクロックとして内部2MHz RC発振器がそれを引き継ぎます。

禁止できないとの定義によるNMIなので、一旦許可されると、ソフトウェアからNMIを禁止するための機構を利用できません。NMIを禁止する方法はMCUをリセットすることだけで、それはNMIが禁止された処から初期状態に戻ります。クリスタル停止検出と一般的なクロックシステムについてのより多くの情報に関してはXMEGAのクロックシステムを網羅するAVR1003応用記述を参照してください。

2.3. PMICでの割り込みレベル許可

各割り込みレベルは個別に許可することができます。これは例えば上で言及した16ビットレジスタ読み込みの例のような非分断操作の場合に望まれ得、高レベル割り込みを防ぐべきではありません。低レベル割り込みだけが、保護されるべき変数を書くことを考察してください。中と高のレベルの割り込みはどうしてもこの変数に影響を及ぼさないで、これらを防ぐ理由はありません。PMICのCTRLレジスタ内のHILVLEN, MEDLVLEN, LOLVLENのビットが個別レベルを許可します。

megaAVR™マイクロコントローラでの単一レベル割り込みシステムと異なり、全体割り込み(SREGのI)ビットが割り込み移行時に解除(0)されないことを考慮に入れることが重要です。PMICは同じレベルを持つ割り込みが互いに割り込めないことを保証する機構を含みます。より多くの情報に関してデータシートで得られるPMIC STATUSレジスタの記述を参照してください。

2.4. 割り込みの全体許可

例え割り込みレベルが部署とPMICの両方で許可されていたとしても、CPUに対して全体割り込みビットが設定(1)されていないと、割り込みは実行されません。アセンブリ言語命令SEIとCLIが各々全体割り込みの許可と禁止を行います(これらの命令についてのより多くの情報に関してはAVR命令一式を参照してください)。

C言語でのプログラミング時、許可/禁止は代表的にインラインアセンブリ言語またはコンパイラ指定の固有関数を使用して行われます。これについての情報を探すにはコンパイラの参考指針を参照してください。

2.5. 割り込みベクタ

割り込みベクタは割り込みが起動される時に実行されるプログラムメモリ内の位置(アドレス)です。メモリ位置は代表的に対応する割り込み処理ルーチン(ISR)が配置されているプログラムメモリアドレスへのJMP(またはRJMP)命令を保持します。個別の割り込みベクタのアドレスはデータシートで指定されます。

安全の予防処置として、未使用割り込みベクタを非プログラムのままにしないことが推奨されます。非プログラムのメモリは不正命令オペコードを含み、その影響は未定義です。RETI命令で“満たされた”未使用割り込みベクタは、ISRを持たない割り込みの実行結果がより少ない危期になることを保証するでしょう。けれども、開発中にこれを用いる場合、それは未定義割り込みが実行されている事実を隠すかもしれません。このバグの種類を通知することが可能なように、開発及びデバッグの間、全ての未使用割り込みに対して擬似ISRの実装を考慮すべきです。

プログラムメモリの応用領域を(再)プログラミングするためにプログラムメモリのブート領域を使用する一方で、割り込みが未だ動作すべきなら、割り込みベクタをブート領域へ移動する必要があります。ベクタが移動されず、一方で応用領域が書かれつつある場合、割り込みは(書き込み処理のために)読むことができないプログラムメモリアドレスの実行を引き起こすでしょう。従って、割り込みが許可されている一方でプログラムメモリの応用領域が書かれる場合、割り込みが正しく実行されるのを保証するためにベクタをブート領域へ移動しなければなりません。ベクタはPMICのCTRLレジスタ内のIVSELビットの設定(1)によってブート領域へ移動されます。ブートロータについてのより多くの情報はAVR109とAVR1316応用記述で得られます。

2.6. 割り込み優先権

割り込みの優先権はそれらのベクタアドレス(またはベクタ番号)によって決められます。ベクタアドレスは与えられたデバイスに対して固定化されています。割り込み優先権は同じ割り込みレベルの2つの割り込みが同時に起きる場合にどちらが先に実行されるかを決めます。

PMICは低レベル割り込みに対してラウンドロビン計画の仕組みを提供し、これはPMICのCTRLレジスタ内のRRENビットの設定(1)によって許可されます。ラウンドロビン優先権の仕組みを許可すると、CPUは最後に実行された割り込みのベクタ番号を保持するように、INTPRIレジスタを自動的に更新します。INTPRIの値はどの低レベル割り込みが最低優先権を持つかを定義します。この方法でのラウンドロビンの仕組みは、優先権が継続的に変更するために、“兵糧攻め”にされる低レベル割り込みがないことを保証します。別の言葉では、最後に実行された低レベル割り込みは、他の低レベル割り込みも処理されることを保証するために、常に最低優先権にされます。

ラウンドロビンの仕組みが禁止される時に、INTPRIレジスタが既定値(\$00)に戻らないことに注意してください。これは最後に実行された低レベル割り込みの番号を含んでいます。既定の優先順への復帰を望むなら、ラウンドロビンの仕組みを禁止した後でINTPRIレジスタに\$00を書かなければなりません。例えラウンドロビンの仕組みが使用されていないと、INTPRIレジスタの値は未だ低レベル割り込みの優先順に影響を及ぼします。

2.7. 割り込み要求フラグ

AVR部署内のいくつかの状態ビットは割り込み要求フラグとして参照されます。これらの割り込み要求フラグは部署内の状態変化、例えばSPIでバイトが受信されたこと、についての情報を提供します。例え対応する割り込みが許可されていなくても、割り込み要求フラグは設定(1)されます。これはファームウェアの主繰り返しでの割り込み要求フラグのポーリングを可能にします。ポーリングはいくつかの場合で好まれ得ますが、事象発生時に実行されるISRによって度々扱われます。ISR処理の1つの利点はより低い重要な作業を一時的に停止することによって重要な部署が直ぐに処置を与えられることです。この場合での割り込み制御器は複数の割り込みレベル持ち、“より低い重要な作業”は“重要な”より高いレベルの割り込みよりも低いレベルの割り込みを含みます。別の利点は消費電力低減のためにAVRが休止形態へ移行し、そして割り込み発生時に起動できることです。どの割り込みが各種休止形態からCPUを起動できるのかを探するにはデータシートを参照してください。

割り込みは代表的にはフラグへの論理1書き込みによって手動で解除(0)でき、または割り込みベクタが実行される時に自動的に解除(0)されます。その他の割り込み要求フラグは或るレジスタ、例えばUSARTデータレジスタ、がアクセスされる時に解除(0)されます。別の例はEEPROM書き込みに対する割り込み要求フラグで、これはEEPROMが書き込み作業中でない間中、継続的に割り込みを生成します。更に、いくつかのフラグは割り込みベクタの実行によって自動的に解除(0)されません。個別フラグの解除(0)についての情報に関してはデータシートを参照してください。

2.8. 割り込み 対 事象システム

CPUに割り込みを扱わせることが常に必要でないことに注意してください。XMEGAの事象システムはCPUの介在なしにハードウェアで代表的な割り込み条件を扱う代替方法を提供します。事象システムは、例えばタイマ上昇溢れが起こる度毎にADC変換を開始することができます。これはCPUの介在が必要とした場合に得られなかった独特なタイミングの機会を提供します。事象システムはAVR1001応用記述で記述されます。

3. ドライバ実装

本応用記述はCで実装された基本ドライバの一括ソースコードを含みます。それはIAR Embedded Workbench[®]コンパイラで書かれています。

このドライバが高い可読性と周辺機能部署の使用法の一般的な例として書かれていることに注意してください。このドライバを応用で使用するとき、関数呼び出し数を減らすために、それが必要とされる場所にコードの関連部分を複写することが好ましいかもしれません。これはコードの速度向上とコード量の低減の両方を行い得ます。

3.1. ファイル

一括ソースコードは次の3つのファイルから成ります。

- `pmic_driver.c` : ドライバ ソース ファイル
- `pmic_driver.h` : ドライバ ヘッダ ファイル
- `pmic_example.c` : ドライバを使用するコード例

利用可能なドライバ インターフェース関数とそれらの使用の完全な概要についてはソースコードの資料を参照してください。

3.2. Doxygen資料化

全てのソースファイルはDoxygenを使用する自動資料生成用に準備されています。Doxygenは特別なキーワードを使用してソースコードを分析することによって、ソースコードから資料を作成するツールです。Doxygenについてのより多くの詳細に関しては<http://www.doxygen.org>を訪ねてください。予めコンパイルされたDoxygen資料は本応用記述に伴うソースコードと共に供給され、ソースコードフォルダの`readme.html`ファイルから利用可能です。



本社

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131
USA
TEL 1(408) 441-0311
FAX 1(408) 487-2600

国外営業拠点

Atmel Asia

Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
TEL (852) 2245-6100
FAX (852) 2722-1369

Atmel Europe

Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
TEL (33) 1-30-60-70-00
FAX (33) 1-30-60-71-11

Atmel Japan

104-0033 東京都中央区
新川1-24-8
東熱新川ビル 9F
アトメル ジャパン株式会社
TEL (81) 03-3523-3551
FAX (81) 03-3523-7581

製品窓口

ウェブサイト

www.atmel.com

技術支援

avr@atmel.com

販売窓口

www.atmel.com/contacts

文献請求

www.atmel.com/literature

お断り: 本資料内の情報はATMEL製品と関連して提供されています。本資料またはATMEL製品の販売と関連して承諾される何れの知的所有権も禁反言あるいはその逆によって明示的または暗示的に承諾されるものではありません。ATMELのウェブサイトに位置する販売の条件とATMELの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適法性の暗黙保証に制限せず、ATMELはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえATMELがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益の損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してATMELに責任がないでしょう。ATMELは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。ATMELはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、ATMEL製品は車載応用に対して適当ではなく、使用されるべきではありません。ATMEL製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

© Atmel Corporation 2008. 全権利予約済 ATMEL[®]、ロゴとそれらの組み合わせ、AVR[®]とその他はATMEL Corporationの登録商標、XMEGA[®]とその他は商標またはその付属物です。他の用語と製品名は一般的に他の商標です。

© HERO 2014.

本応用記述はATMELのAVR1305応用記述(doc8043.pdf Rev.8043A-02/08)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。