

AVR1519 : XMEGA-A1 Xplain練習 – XMEGA 暗号エンジン

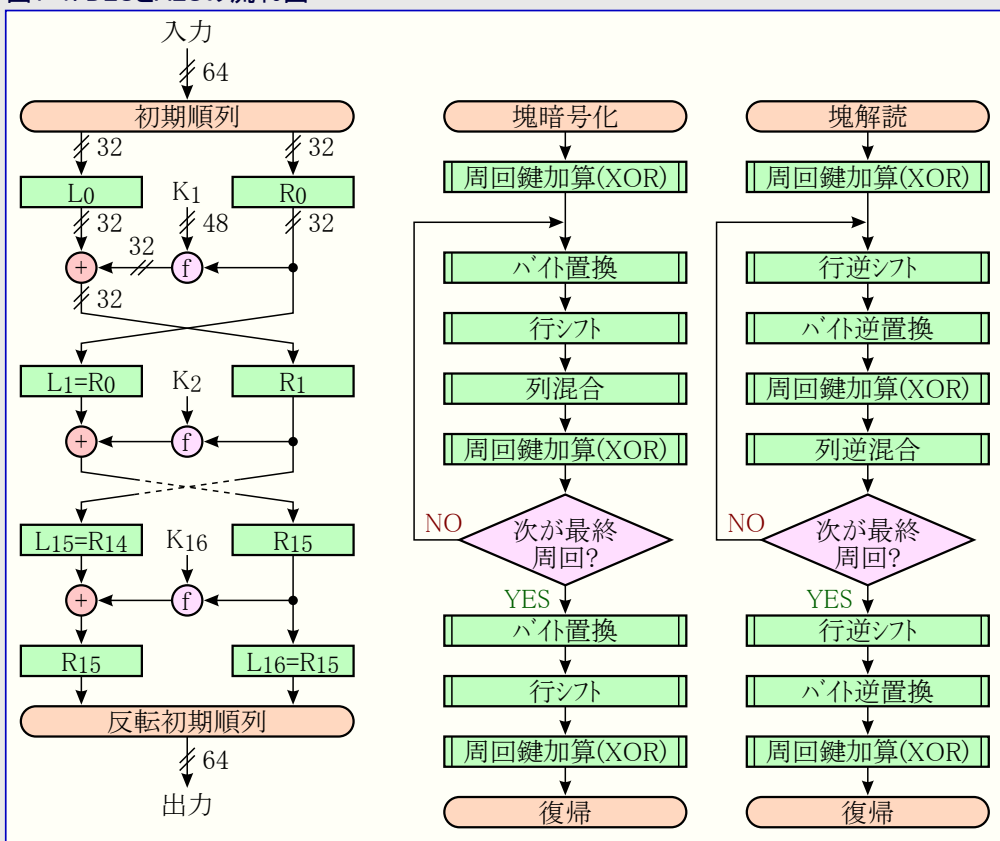
前提条件

- 必要な知識
 - ・ ATMEL® AVR Studio®の使い方
 - ・ ATMELのXMEGA基礎練習
- ソフトウェア必要条件
 - ・ ATMEL AVR Studio 5
- ハードウェア必要条件
 - ・ XMEGA-A1 Xplain
 - ・ JTAGICEmk3(またはJTAGICEmk II かAVR ONE!)
- 予想完了時間
 - ・ 1.5時間

1. 序説

この練習は言及される暗号算法の実用的な実装に集中して、ATMEL AVR® XMEGA®の強力なハードウェア支援を用いてそれらの容易な実装方法を説明します。DESとAESの算法の後ろにある演算の詳細はこの練習の範囲外で網羅されません。

図1-1. DESとAESの流れ図



8ビット ATMEL
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. 8405A-07/11, 8405AJ1-03/14



2. 暗号概要

本章は基本的な暗号形態設定任意選択とATMEL XMEGAの機能の概要を提供します。

2.1. DES - データ暗号化規格

XMEGAは命令1式に加えて付加的な命令、**DES**命令を持ちます。この命令はR0～R7のレジスタに格納されたデータを暗号化または解読するための鍵としてR8～R15の汎用レジスタに格納される鍵データを使用します。1つの**DES**命令は、DES算法での1周回を実行します。DES算法は16段階を用い、従って正しいDES暗号文または平文のために次数を増して16回の**DES**命令が実行されなければなりません。**DES**命令への有効な引数は\$00～\$0Fの範囲の数値ですが、正しい手順、即ち\$00,\$01,～,\$0E,\$0Fを用いた場合にだけ正しい結果を出力します。

2.1.1. DES暗号化

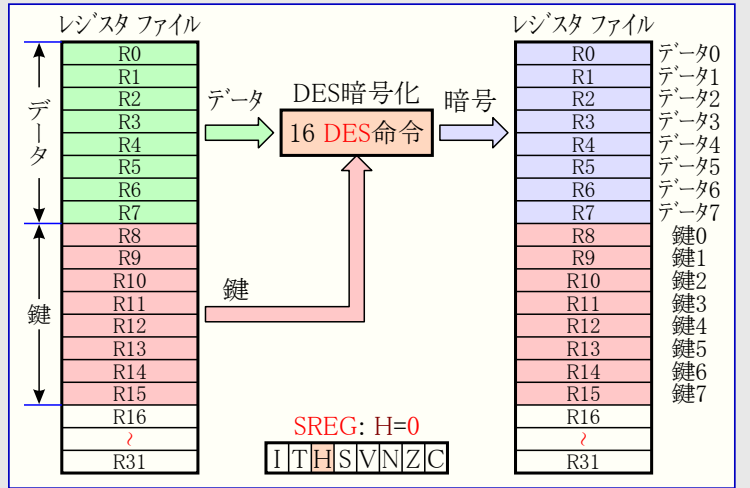
DES算法を用いる暗号データはXMEGA上で非常に簡単で、使用者は暗号化法の演算や操作について何も知る必要がありません。

実用的な実装は以下の4段階だけに依存します。

1. 平文データをR0～R7のレジスタ内に設定してください。
2. 鍵データをR8～R15のレジスタ内に設定してください。
3. **SREG**の**"H"**フラグを解除(0)してください。
4. **DES \$00～DES \$0F**で16回**DES**命令を実行してください。

今や暗号文がR0～R7のレジスタで利用可能です。この時点に於いて**DES**命令がコンパイラによって直接支援されない一方で、正しいDESの暗号化と復号を実行するアセンブリ言語ライブラリがこの応用記述で供給され、Cまたはアセンブリの言語で書かれた応用ソフトウェアで使用する準備が整っています。

図2-1. 暗号化に使用するレジスタ



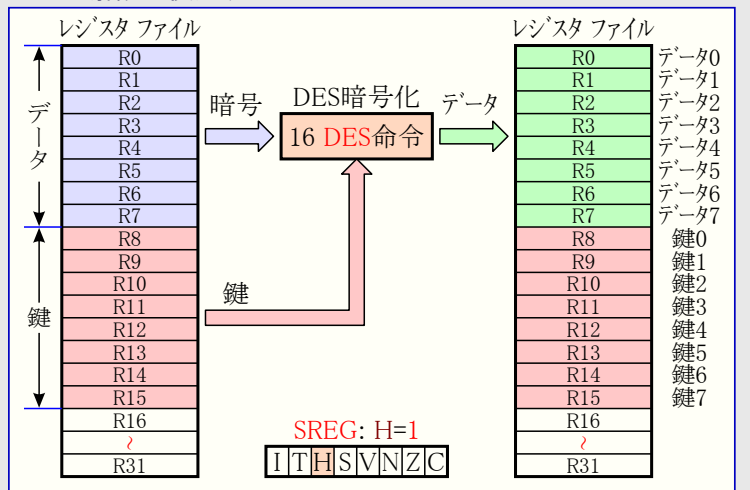
2.1.2. DES解読

DES算法でのデータ解読はまさに暗号化のように簡単です。同じ段階が適用され、今回は**SREG**の**"H"**フラグが設定(1)されなければならないだけです。

1. 暗号文データをR0～R7のレジスタ内に設定してください。
2. 鍵データをR8～R15のレジスタ内に設定してください。
3. **SREG**の**"H"**フラグを設定(1)してください。
4. **DES \$00～DES \$0F**で16回**DES**命令を実行してください。

今や平文がR0～R7のレジスタで利用可能です。

図2-2. 解読に使用するレジスタ



2.2. AES - 新暗号規格

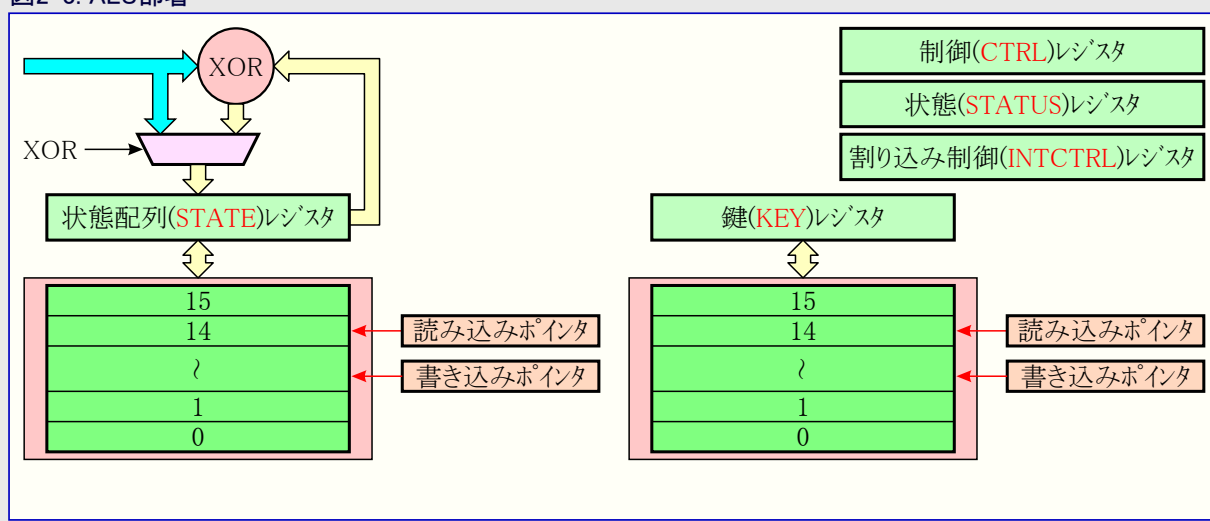
DES命令と異なり、**AES**はデータレジスタ、制御と状態のレジスタを持つハードウェア部署です。これは変換完了時に割り込みを生成することができます。

この部署は以下を含みます。

1. 制御レジスタ(**CTRL**): 部署のリセット、変換開始、暗号化/解読と状態メモリにデータを格納する時に使用されるべきどちらかのXOR形態の選択に使用されます。
2. 状態レジスタ(**STATUS**): 異常(**ERROR**)と状態配列準備可割り込み要求フラグ(**SRIF**)のフラグを含みます。**ERROR**は状態配列(**STATE**)レジスタまたは鍵(**KEY**)レジスタが変更されると同時に開始/走行(**START**)ビットが設定(1)された場合に設定(1)されます。メモリが完全に読み書きされていない時に変換が開始された場合にも設定(1)されます。**SRIF**は変換完了時に設定(1)されます。
3. 鍵メモリ: 鍵(**KEY**)レジスタを通してアクセスされる16バイトのメモリです。

4. 状態メモリ: 状態配列(STATE)レジスタを通してアクセスされる16バイトのメモリです。
5. 割り込み制御レジスタ(INTCTRL): 許可と割り込みレベルの設定に使用されます。

図2-3. AES部署



AES部署と関係するレジスタについての更なる情報はXMEGA手引書で得られます。

3. 概要

この実践作業はATMEL XMEGAの基本的な暗号機能のいくつかを網羅します。

課題1. DES暗号化と解読

この課題は簡単なメッセージを暗号化と解読をするためのDES応用記述ライブラリの使用法を示します。

課題2. 3DES暗号化と解読

この課題は3重DES(3DES)の使用法を示します。3DESライブラリ関数を使用しますが、3つの標準DES呼び出しを用いて自身の3DESも作ります。

課題3. AES暗号化と解読

この課題はAES部署を制御してメッセージを暗号化して解読するのにAES関数を使用するのが如何に簡単かを示します。AESを使用したメッセージの暗号化と解読にかかる周期数も考察します。

課題4. AESとCBC

この課題はAES暗号化に暗号塊連鎖(CBC)を追加します。この課題では正しい構文法を探し出すためにAESライブラリ ファイル内の関数を調べることが必要です。CBC AESを用いたメッセージを暗号化と解読にかかる周期数も考察して、直前の課題と比較します。



- ・この絵文字は任意選択の課題と質問を表します。答えはこの練習の最後の要約章で与えられます。

4. 課題1: DES暗号化と解読

この課題はDES応用記述と、応用内にDES暗号化と解読を素早く実装するためのライブラリ関数の使用法を紹介します。

この課題の目的はあなたが以下の方法を知ることです。

- ・DES暗号化と解読用の応用記述ライブラリ関数を使う。
- ・暗号化に使用されたのと僅かに異なる鍵で解読した場合に何が起きるかを見る。
- ・パリティビットとしてどの64ビット符号が使用されたかを探し出す。



すること

1. **Atmel XMEGA-Crypto**フォルダに位置付けてください。 `xmega_crypto.avrsln`解決ファイルを開き、始動(Startup)プロジェクトとしてそれを選択することによってTask1を活性に設定してください。

この課題では入力データ、出力データ、暗号データの格納に3つの文字列を使用します。これは解読したメッセージが入力データと同じであることを確認する簡単な検査を許します。

2つのDES鍵文字列が定義されていることに注意してください。これは僅かな違いが出力にどう影響を及ぼすかを見るのに、暗号鍵と僅かに異なる鍵でメッセージを解読するための簡単な方法を許すために行われます。

2. コードを学んで構造体に慣れてください。

全てのプログラムはメッセージを暗号化し、それを解読して入力と出力が同じことを確かめます。

3. AVR Studioでコードを構築して走らせて、入力データと出力データが本当に同じことを確かめてください。

4. 解読鍵値内の値の1つを変更してみて、これが解読にどう影響を及ぼすかを見てください。



- DES規格は実際の鍵として56ビットだけを使用します。解読鍵内の8ビットはパリティ検査に使用されます。これらのパリティビットが64ビット解読鍵内の何処に配置されているかを探し出してください。

助言: これらのビットの0/1切り替えはDES解読算法で出力に影響を及ぼしません。

5. 課題2: 3DES暗号化と解読

伝統的なDES暗号化は56ビットでこれが良好な安全を提供するとは言え、これは大いに安全な応用にはもはや推奨されません。3回のDES使用に基づく、3重データ暗号化規格(3DES)は56ビットから168ビットに鍵の長さを増加します。これはそれを遥かに安全にしますが、実際上各データ塊で3つの操作を実行するためにより遅くもします。

この課題の目的はあなたが以下の方法を知ることです。

- メッセージを暗号化と解読をするために応用記述3DESライブラリ関数を使用します。
- 3DES実装を”手動で”作るためにDESライブラリを使用します。



すること

1. Atmel XMEGA-Cryptoフォルダに位置付けてください。xmega_crypto.avrsln解決ファイルを開き、始動(Startup)プロジェクトとしてそれを選択することによってTask2を活性に設定してください。

コードはDES応用記述によって提供される関数を使用します。

2. Task2.c内のコードを理解してみてください。その後それをコンパイルして走らせてください。コード走行後に暗号データと出力データを監視するために監視(Watch)ウィンドウを使用してください。

3DESが3回のDES走行に基づくため、あなたの課題は直前の課題からのDES関数を使用して3DESを実装することです。3DES符号化は解読段階が後続する暗号化段階、その後の最後の暗号化段階によって行われます。各段階はそれ自身の鍵を使用します。

3. 主関数に於いてDEC_EncryptとDEC_Decryptの関数呼び出しを見るでしょう。それらが解読に関して正しい順で配置されているのを確かめてください。
4. 欠けているパラメータを挿入してください。参照として前の課題のコードを利用するかもしれません。
5. DEC_EncryptとDEC_Decryptの関数によって復号されたデータが初期入力データ、それとDEC_3DES_EncryptとDEC_3DES_Decryptの関数から解読されたデータとも等しいことを確かめてください。



- DEC_3DES_Encryptを実行するのに必要な周期数はDEC_Encrypt関数の3倍よりも大きいですか？。

6. 課題3: AES暗号化と解読

この課題ではAES部署、そしてAES暗号化と解読を実装するためにAVR1318応用記述のソフトウェアライブラリを使用法を考察します。

この課題の目的はあなたが以下を知ることです。

- メッセージを暗号化と解読をするためにAVR1318 AES応用記述ライブラリ関数を使用する方法。
- 暗号化と解読の両方に使用する鍵がどれか。
- AESの符号化と復号を実行するのにどのくらいのクロック周期が必要かを測定する方法。



すること

1. Atmel XMEGA-Cryptoフォルダに位置付けてください。xmega_crypto.avrsln解決ファイルを開き、始動(Startup)プロジェクトとしてそれを選択することによってTask3を活性に設定してください。

2. プログラムをコンパイルして走らせてください。

3. indata, cipherdata, outdata, aeskey, lastsubkeyの変数を監視(Watch)に追加してください。

4. プログラム全体を通して段階実行して変数の内容がどう変わるかを考察してください。

AESがハードウェア部署のため、AESソフトウェアはどんなインラインアセンブリ言語またはアセンブリ関数への呼び出しもなしに、C言語で容易に書くことができます。

5. プロセッサ(Processor)ウィンドウで動かないように見える周期計数器(Cycle Counter)があるのを見るでしょう。これはデバイスがハードウェアの周期計数を提供せず、周期計数からの追跡機能が生成され得ないからです。けれども、私たちはAES暗号化と解読を行うのにかかる周期数の測定を望みます。これを行うためにデバッグ基盤を一時的に変更します。"Debug"メニューで"Platform and device"を選び、デバッグに対して"JTAGICE3mk II"の代わりに"Simulator 2"を使用するように選択してください。

6. 暗号化と解読にどのくらいの周期(数)がかかるかを測定してください。それを右クリックすることによって周期計数器をリセットすることが可能なことを憶えて置いてください。



- AES_lastsubkey_generate()関数の目的は何ですか？。
- JTAGICE3(または他のデバッグ)で周期計数器が不可能なのは何故ですか？。

7. 課題4: CBC形態でのAESの使い方

要約:

暗号塊連鎖(CBC:Cipher Block Chaining)形態では平文の各塊が暗号化されつつある前に直前の暗号文の塊とXORされます。このように、各暗号文の塊はその時点までに処理された全ての平文の塊に依存します。また、各メッセージを独自にするため、最初の塊で初期化ベクトルを使用しなければなりません。

この課題ではCBCがどう実装され得るのかを考察します。

この課題の目的はあなたが以下の方法を知ることです。

- ドライブ ファイルを調べてその関数の構文法を見つけます。
- CBC形態を用いる3つのメッセージの塊の暗号化。
- CBCを用いる同じ3つのメッセージの塊の解読。

前の課題では全てのコードが既に書かれていました。これはもうそうではありません。コードの殆どは提供されますが、暗号化と解読の関数に対する引数の入力と、正しい引数が各関数呼び出しに渡されるのを確実にすることが必要です。



すること

1. **Atmel XMEGA-Crypto**フォルダに位置付けてください。 `xmega_crypto.avrsln` 解決ファイルを開き、始動(Startup)プロジェクトとしてそれを選択することによって**Task4**を活性に設定してください。
2. `main()`関数では、完成していない2つの関数呼び出しを持ちます。暗号化と解読のルーチンに関してコードを完成してください。コードをコンパイルして走らせてください。
3. シミュレータに変更して暗号化と解読の関数が費やす周期数を測定してください。



- 符号化と復号に費やされる周期(数)の違いを説明できますか?。

8. 要約

この実践練習ではあなたの応用にDESとAESの暗号の実装がどのくらい容易かを見ました。これらの特徴を完全に利用し得るのに算法の知識は全く必要とされません!。

9. 資料

- XMEAの手引書とデータシート
 - <http://www.atmel.com/xmega>
- ATMEL AVR Studio 5
 - <http://www.atmel.com/avrstudio>
- ATMEL用IAR Embedded Workbench®コンパイラ
 - <http://www.iar.com/>

10. ATMEL技術支援センター

ATMELは以下の利用可能な多数の支援チャネルを持ちます。

- ウェブ入り口 : <http://www.atmel.no/> 全てのATMELマイクロコントローラ
- Eメール : avr@atmel.com 全てのATMEL AVR製品
- Eメール : avr32@atmel.com 全ての32ビットAVR製品

以下のサービスへのアクセスを得るにはウェブ入り口で登録してください。

- 豊富なFAQデータベースへのアクセス
- 技術支援要請の容易な依頼
- あなたの過去の全支援要請の履歴
- ATMELマイクロコントローラ時事通信の受信のための登録
- 利用可能な練習と練習材料についての情報取得



Atmel Corporation

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

Atmel Asia Limited

Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
TEL (+852) 2245-6100
FAX (+852) 2722-1369

Atmel Munich GmbH

Business Campus
Parking 4
D-85748 Garching b. Munich
GERMANY
TEL (+49) 89-31970-0
FAX (+49) 89-3194621

Atmel Japan

141-0032 東京都品川区
大崎1-6-4
新大崎勸業ビル 16F
アトメル ジャパン合同会社
TEL (+81)(3)-6417-0300
FAX (+81)(3)-6417-0370

© 2011 Atmel Corporation. 全権利予約済

ATMEL®、ATMELロゴとそれらの組み合わせ、それとXMEGA®、AVR Studio®、AVR®、AVR®ロゴとその他はATMEL Corporationの登録商標また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。

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のウェブサイトに表示する販売の条件と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、または利益と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。

© HERO 2014.

本応用記述はATMELのAVR1519応用記述(doc8405.pdf Rev.8405A-07/11)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。