AVR001：条件付アセンブルと可搬性マクロ

要点

- 増した可搬性
- より容易なコード書き
- 簡単化されたI/Oレジスタ アクセス
- 改善されたアセンブル状態還元

1. 序説

この応用記述はAVRアセンブラ Ver. 1.74とそれ以降に存在する条件付アセンブル機能を記述します。AVRアセンブラ®はAtmelのウェブサイトのAVR部で得られるAVR Studio®と共に既定でインストールされます。

条件付アセンブルの使用法の例が提供されます。例の1つはデバイス定義ファイルの変更以外の他の修正なしにどのAVR®もアセンブルする標準的なコードを書くことをプログラマに許すマクロ群です。

2. 動作の理屈

条件付アセンブル(CA)はVer. 1.74のAVRアセンブラで導入されました。条件付アセンブルはC言語で利用可能なプリプロセッサ擬似命令と同様の一連の擬似命令に基いています。

Ver. 2.0以降のAVRアセンブラもCのプリプロセッサを模したプリプロセッサを持ちます。これは例えば、使われるデバイスに対するCヘッダ ファイルを許します。

通常のプロプロセッサと条件付きアセンブル擬似命令の違いは後者がアセンブル時に評価され、従ってマクロを上書きする意味として使われるかもしれないことです。通常のプロプロセッサは、例えば(例えそれが更に条件付アセンブル擬似命令で行われ得るとしても)デバイス依存コードの構成と選択に使われるかもしれません。(本応用記述は条件付きアセンブル擬似命令だけを扱います。)

擬似命令の全一覧についてはアセンブラのAVR sudioヘルプをご覧ください。

2.1. 条件付アセンブル

アセンブラは条件付アセンブル擬似命令式をアセンブル時に評価し、条件付アセンブル擬似命令によって囲まれたコードが含まれるべきか否かを決めます。

条件付アセンブル擬似命令の一覧は表2-1.で示されます。

表2-1. 条件付アセンブル擬似命令

擬似命令	内容
.ifdef 〈シンボル〉	シンボルが定義済みなら、.ifdefと対応する.elseまたは.endif間に存在するコードを含めます。.EQUと.SETによって宣言されたシンボルだけが.ifdefによって評価され、.DEF擬似命令で定義されたシンボルはアセンブラによって違うように扱われ、.ifdefと共に使えません。
.ifndef 〈シンボル〉	シンボルが未定義なら、.ifdefと対応する.elseまたは.endif間に存在するコードを含めます。.EQUと.SETによって宣言されたシンボルだけが.ifndefによって評価され、.DEF擬似命令で定義されたシンボルはアセンブラによって違うように扱われ、.ifndefと共に使えません。
.if 〈式〉	式を0以外(真)と評価したなら、.ifと対応する.else,.elifまたは.endif間に存在するコードを含めます。
.elif 〈式〉	.elifは.ifが先行しなければなりません。式を0以外(真)と評価したなら、.elifと対応する.else,.elifまたは.endif間に存在するコードを含めます。
.else	.elseは.ifまたは.elifが先行しなければなりません。対応する.ifまたは.elifに対して指定された式を0以外(真)と評価したなら、.elseと対応する.endif間に存在するコードを含めます。
.endif	.endifは.ifdef,.ifndef,.ifまたは.elifが先行しなければなりません。.endifは対応する対応する.ifdef,.ifndef,.ifまたは.elifに対する範囲の終りを定義します。

擬似命令は5段まで入れ子にできます。含まれた条件付アセンブル擬似命令に依存してマクロが違ってアセンブルされるために、条件付アセンブル擬似命令は.MACRO擬似命令と組み合わせることができます。これは本応用記述に対するコード例で使われています。

8-bit AVR® マイクロ コントローラ

応用記述

Rev. 2550E-04/08, 2550EJ4-01/21

2.2. ﾃﾞﾊﾞｯｸﾞ擬似命令

ｺｰﾄﾞ開発時、各種の還元形式はｱｾﾝﾌﾞﾙ時に有用です。この目的に対して2つの新しい擬似命令が追加されました。1つは出会った場合にﾒｯｾｰｼﾞ ｳｨﾝﾄﾞｳへﾒｯｾｰｼﾞを出力し、もう1つは(異常ﾒｯｾｰｼﾞを伴う)異常を発行します。ﾃﾞﾊﾞｯｸﾞ擬似命令は**表2-2.**で記述されます。

表2-2. ﾃﾞﾊﾞｯｸﾞ擬似命令

擬似命令	内容
.message "文字列"	ｺｰﾄﾞのｱｾﾝﾌﾞﾙ時に.message擬似命令に出会うと、ﾒｯｾｰｼﾞ ｳｨﾝﾄﾞｳに"文字列"が書かれます。
.error "文字列"	ｺｰﾄﾞのｱｾﾝﾌﾞﾙ時に.error擬似命令に出会うと、ｱｾﾝﾌﾞﾙ異常が発行され、ﾒｯｾｰｼﾞ ｳｨﾝﾄﾞｳに"文字列"が書かれます。

これらの擬似命令はｱｾﾝﾌﾞﾙ状態についての情報を提供するために、(例えば条件付ｱｾﾝﾌﾞﾙ)擬似命令と組み合わせることができます。.message擬似命令の使い方は本応用記述に対するｺｰﾄﾞ例に含まれています。AVRｱｾﾝﾌﾞﾗ 1.77.1が.error擬似命令に出会った時に更なるｱｾﾝﾌﾞﾙを止めることに注意してください。

2.3. 例1: 条件付ｱｾﾝﾌﾞﾙ擬似命令の一般的な使い方

或るﾌﾟﾛｼﾞｪｸﾄから次へのｺｰﾄﾞ部再使用は度々大いなる時間の節約になり得ます。異なる使用目的に仕立てることを必要とするとき、ｺｰﾄﾞ部の再使用はﾌﾟﾘﾌﾟﾛｾｯｻ擬似命令も、この場合は条件付ｱｾﾝﾌﾞﾙ擬似命令もなしで、結果としてｺｰﾄﾞ部の異なる色々な版を生じるでしょう。条件付ｱｾﾝﾌﾞﾙ擬似命令を使う1つの版は様々な目的ﾃﾞﾊﾞｲｽに対して使うことができます。

例としてUART通信に使う入出力ﾋﾟﾝの初期化を考察してください。

```
          .EQU    ATmega128  =1                    ; ATmega128ｼﾝﾎﾞﾙ宣言
;         .EQU    ATmega16   =1                    ; ATmega16ｼﾝﾎﾞﾙ宣言(現状注釈)
          .EQU    UART       =0                    ; UART番号指定(0=UART0または1=UART1)

          .ifdef ATmega128                         ; ATmega128なら、
            .message "UART部をATmega128用にｱｾﾝﾌﾞﾙ."  ; ATmega128使用表示
              .if UART == 0                        ; UART0指定なら、
                .message "UART0使用."              ; UART0使用表示
                  SBI       DDRE,PE1               ; UART0のTxDを出力として設定
              .elif UART == 1                      ; UART1指定なら、
                .message "UART1使用."              ; UART1使用表示
                  SBI       DDRD,PD3               ; UART1のTxDを出力として設定
              .else                                ; UART0,1以外で、
                .error "無効UART番号"              ; 無効UART番号表示
              .endif                               ;
            .elif ATmega16                         ; ATmega16なら、
              .message "UART部をATmega16用にｱｾﾝﾌﾞﾙ."  ; ATmega16使用表示
              .if UART == 0                        ; UART0指定なら、
                .message "UART0使用."              ; UART0使用表示
                  SBI       DDRD,PD1               ; UART0のTxDを出力として設定
              .else                                ; UART0以外で、
                .error "無効UART番号"              ; 無効UART番号表示
              .endif                               ;
            .endif                                 ;
```

このｺｰﾄﾞ例で見られるように、ﾃﾞﾊﾞｲｽ特有の初期化(例えばUART)を1つのﾌｧｲﾙに含めることができます。これは再使用するｺｰﾄﾞ部のより良い制御を可能にします。

2.4. 例2: マクロでの条件付アセンブル

本応用記述のコード例は正しくアセンブルするためにAVRアセンブラ Ver. 1.77またはそれ以降が必要です。AVRアセンブラはAVR Studioに含まれ、それはAtmelのウェブサイトのAVR部からダウンロードできます。

"macros.inc"ファイルはI/Oとデータ空間で容易にビットとバイトをアクセスするためのいくつかのマクロを含みます。マクロは表2-3.に於いて注釈付きで一覧されます。

表2-1. "macros.inc"ファイル内のマクロ定義

マクロ名	内容
SETB [アドレス, ビット番号, レジスタ]	"ビット設定" - I/O空間の'アドレス'位置で'ビット番号'のビットを設定(1)します。'レジスタ'はR16～R31が使えます。
CLRB [アドレス, ビット番号, レジスタ]	"ビット解除" - I/O空間の'アドレス'位置で'ビット番号'のビットを解除(0)します。'レジスタ'はR16～R31が使えます。
SKBS [アドレス, ビット番号, レジスタ]	"ビット=1ならスキップ" - I/O空間の'アドレス'位置で'ビット番号'によって指定されたビットが設定(1)なら、マクロに後続する命令をスキップします。
SKBC [アドレス, ビット番号, レジスタ]	"ビット=0ならスキップ" - I/O空間の'アドレス'位置で'ビット番号'によって指定されたビットが解除(0)なら、マクロに後続する命令をスキップします。
STORE [アドレス, レジスタ]	"レジスタ格納" - I/O空間の'アドレス'位置に'レジスタ'の内容を格納します。
LOAD [レジスタ, アドレス]	"レジスタ格納" - I/O空間の'アドレス'位置から内容を'レジスタ'に取得します。

I/O空間(と拡張I/O空間)をアクセスするのにこれらのマクロを使う理由は、アクセスするレジスタがI/O空間の何処に配置されているかをコードの書き手が考慮する必要がないからです。全ての命令がI/O空間の全アドレスに届く訳ではないので、(既存)マクロを使わない場合に、これが必要でしょう。故に利点は多数あります。

- 著者はI/O配置を知る必要がなく、只レジスタの名前だけです。
- レジスタ名とビット名に対する標準定義ファイルが使えます。
- レジスタ アクセスに最もコード量効率的な命令が使われます。
- アセンブリ コードがコード修正なしでどのデバイスにも移転できます。

2.4.1. ビット設定マクロ

全てのマクロが本質的に同様なため、SETBだけが詳細に記述されます。

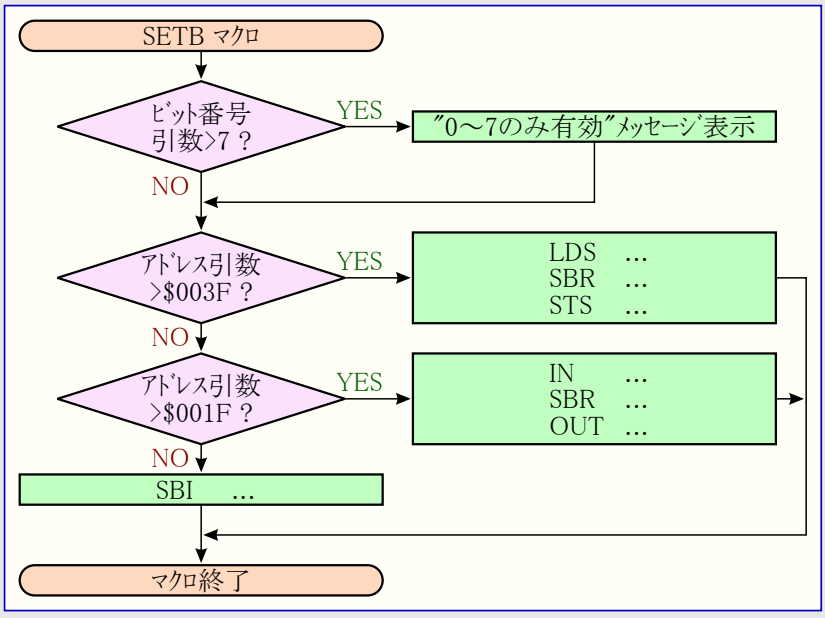
3つの引数、目的アドレス、ビット番号、レジスタがSETBマクロに対して指定されます。レジスタはアドレスが$001Fよりも大きな場合にだけ使われますが、正しいアセンブル結果と最良の可搬性の機会を保証するために、いつも指定することが推奨されます。

ビット番号の範囲は0～7の間であることを検証され、条件違反の場合は.error擬似命令を使って誤りが発行されます。

アドレスが$001F以下の場合はビットを設定(1)するのにSBI命令が使われます。アドレスが$0020～$003F間なら、そのアドレスのアクセスにIN命令とOUT命令が使われます。最後に、アドレスが$0040以上なら、LDS命令とSTS命令が使われます。

図2-1.は条件付アセンブル擬似命令を使うSETBマクロをアセンブラがどう取り扱うかを示します。

図2-1. SETBマクロ内側の条件付アセンブル擬似命令に対するアセンブルの流れ



2.5. 改良

YまたはZレジスタの1つの犠牲で、これらのレジスタの1つが間接アクセス用に予約されてLDS命令とSTS命令がLDD命令とSTD命令に置換できるなら、LOADとSTOREのマクロはより速い実行ともっと簡潔なコードに改良できます。これはコードの書き手を圧迫するため、提供した実装に付加されていません。

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
東熱新川ﾋﾞﾙ 9F
ｱﾄﾒﾙ ｼﾞｬﾊﾟﾝ株式会社
TEL (81) 03-3523-3551
FAX (81) 03-3523-7581

製品窓口

ｳｪﾌﾞｻｲﾄ
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

本応用記述はAtmelのAVR001応用記述(doc2550.pdf Rev.2550E-04/08)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訳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。