第4章

命令セット

このPDFは，CQ出版社発売の「ARM Cortex-M3システム開発ガイド」の一部分の見本です．
内容・購入方法などにつきましては以下のホームページをご覧下さい．
<http://shop.cqpub.co.jp/hanbai/books/36/36491.htm>

この章では以下の項目を紹介します．

- アセンブリの基本
- 命令リスト
- 命令の解説
- Cortex-M3にあるいくつかの便利な命令

本章では，Cortex-M3の命令セットと多くの命令の使用例を見ていきます．さらに，本書の**付録A**にはサポートされている命令のクイック・リファレンスがあります．各命令の詳細に関しては，*ARM v7-M Architecture Application Level Reference Manual*（Ref2）を参照してしてください．

4.1 アセンブリの基本

ここで，本書中の以降のコード例を理解しやすくするために，ARMアセンブリの基礎的な文法を紹介します．本書中のほとんどのアセンブラ・コード例は，GNUツール・チェーンにフォーカスしている第19章を例外として，ARMアセンブラ・ツールに基づきます．

4.1.1 アセンブリ言語：基礎的な文法

アセンブラ・コードでは，以下の命令フォーマットが一般に使用されます．

```
label
        opcode operand1, operand2,... ; Comments
```

ラベルはオプションです．命令のアドレスを決定するのにラベルを使用できるように，命令のうちのいくつかはそれらの前にラベルをもっている場合があります．その後，オペコード（命令）に続いてオペランドがきます．通常，第1オペランドは操作のデスティネーションです．命令のオペランドの数は，命令のタイプに依存します．また，オペランドの文法フォーマットはさらに異なる場合があります．たとえば，イミディエート値のデータは，以下のように，通常**#number**形式になります．

```
MOV R0, #0x12 ; Set R0 = 0x12 (hexadecimal)
MOV R1, #'A'  ; Set R1 = ASCII character A
```


見本

各セミコロン（;）の後のテキストはコメントです．これらのコメントはプログラム動作に影響しません．しかし，プログラムを理解するのに役立ちます．

EQUを使用して定数を定義でき，プログラム・コードの内部でそれらを使用できます．例を次に示します．

```
NVIC_IRQ_SETEN0 EQU 0xE000E100
NVIC_IRQ0_ENABLE EQU 0x1
    ...
    LDR R0,=NVIC_IRQ_SETEN0  ; LDR here is a pseudo instruction that
                             ; convert to a PC relative load by
                             ; assembler.
    MOV R1,#NVIC_IRQ0_ENABLE ; Move immediate data to register
    STR R1, [R0]             ; Enable IRQ 0 by writing R1 to address
                             ; in R0
```

アセンブラが正確な命令を生成することができないとき，その命令の機械語を知っていれば，DCIを使って命令をコード化できます．

```
DCI 0xBE00 ; Breakpoint (BKPT 0), a 16-bit instruction
```

コード中に2進データを定義するにはDCB（文字のようなバイト・サイズの定数値用）とDCD（ワード・サイズの定数値用）を使用できます．

```
    LDR R3,=MY_NUMBER ; Get the memory address value of MY_NUMBER
    LDR R4,[R3]       ; Get the value code 0x12345678 in R4
    ...
    LDR R0,=HELLO_TXT ; Get the starting memory address of
                      ; HELLO_TXT
    BL PrintText      ; Call a function called PrintText to
                      ; display string
    ...
MY_NUMBER
    DCD 0x12345678
HELLO_TXT
    DCB "Hello¥n",0 ; null terminated string
```

アセンブラ文法は，どのアセンブラ・ツールを使用しているかに依存することに注意してください．ここでは，ARMアセンブラ・ツールの文法を紹介します．ほかのアセンブラの文法については，ツールとともに提供されるコード例から始めることをお勧めします．

4.1.2　アセンブリ言語：Suffixesの使用

ARMプロセッサのアセンブリ言語では，命令の後ろに**表4.1**で示した，サフィックスを付けて実行条件を指定します．

Cortex-M3で，実行条件を指定するサフィックス（以下，条件実行サフィックスと呼ぶ）は，分岐命

見本