ARMの非境界整列アドレスのデータアクセス

RISC系CPUでは16bitや32bitのデータアクセスを行う時に、それぞれ2バイト境界や4バイト境界アドレスでないと例外トラップする仕様も多いがARMv6T2以降では非境界アドレスのデータアクセスもサポートされている。

ARMv7の場合

ARMv7 以降では、非境界整列アクセスが義務付けられてる（デフォルト）。

ARMv7-A と ARMv7-R では、システム制御レジスタSCTLR の A ビット(bit1)によって、境界調整チェックのイネーブル(A=1)またはディセーブル(A=0)を制御できる。又、ARMv7-M では、コンフィギュレーションおよび制御レジスタCCRの UNALIGN_TRP ビット、ビット 3 がこれを制御する。

境界調整チェックがイネーブルされている場合、すべての非境界整列ワードまたはハーフワード転送が行われると、境界調整例外が発生する。

ディセーブルされている場合は、LDRLDRHSTRSTRHLDRSHLDRTSTRTLDRSHTLDRHTSTRHTおよび TBH 命令で非境界整列アクセスを行うことができる。他のデータアクセス命令が行われると常に非境界整列データの境界調整例外が発生する STRD および LDRD では、指定されたアドレスはワード境界で整列させる必要がある。

Abitは、デフォルトで「0」となっており、非境界アドレスのアクセスも可能となっているが、システム起動初期化時 A=1 として非境界アクセスをチェックし、安定して起動したら A=0 とするような使い方もできる。

```
    asm("   mrc    p15, 0, r0, c1, c0, 0");
    asm("   bic    r0, r0, #0x00000002"); /* clear A bit (disable alignment
fault) */
    asm("   mcr    p15, 0, r0, c1, c0, 0");
```

GCCコンパイラの動作

GCCは、デフォルトで -munaligned-access が有効となっているらしく、データアクセスが境界調整されている場合、コマンドラインオプション -mno_unaligned_access を使って非境界整列アクセスをサポートするライブラリ関数へのリンクを避けることもできる。

補足

---

1. 非境界アクセスが発生すると、実際のメモリバス上は複数回のバスアクセスとなり、多少アクセス時間がかかる。

参考

---

1. ARM® コンパイラ armasm ユーザガイド 6.18 アドレス境界調整 2. RealView Compilation Tools アセンブラガイド 4.2.1. アドレス境界調整 3. GCC(ARM)のunaligned data accessコード生成

From:
https://www.yokoweb.net/dokuwiki/ - **新石器Wiki**

Permanent link:
https://www.yokoweb.net/dokuwiki/programing/arm-programing/arm-unaligned-access?rev=1524717003

Last update: **2018/04/26 13:30**