Cortex-M4 数据处理指令小结

Cortex-M4 对 Cortex-M3 完全向下兼容，并添加了一系列的数据处理指令共计 101 条，包括饱和算术指令、SIMD 算术指令，以及用于数字信号处理的 MAC 系列指令。Cortex-M4 还可选配浮点功能，称之为 Cortex-M4F，这是一个单精度浮点硬件处理引擎，含有指令共计 25 条。

1. 前置说明

1.1 指令助记符的进一步分解

指令的助记符其实可以拆分成更小的元素，例如，SMLALDX 指令相当于 S MLA L D X，拆分时按取大优先的原则。在下文中列出指令时，也根据拆分的元素添加额外的空格——在书写汇编程序时，可是不能加入这些

1.1.1 通用元素

S ：有符号
U ：无符号
Q ：**有符号数**饱和加减法
H ：两数之和或者之差除以 2
T ：高(Top)半字，即 32 位整数的[31:16]
B ：低(Bottom)半字，即 32 位整数的[15:0]
16 ：SIMD 半字运算
8 ：SIMD 字节运算
ASX：SIMD 半字先加后减
SAX：SIMD 半字先减后加
PK ：PacK，打包
D ：给某一寄存器的值翻倍

1.1.2 专用于 MAC 指令的元素

MUL：乘
MU ：同 MUL
MLA：乘加
MLS：乘减
M/W：乘法是为了求出被乘数的带权值，权的范围在[0,1)之间．M 是 32 位精度，W 是 16 位精度
R ：(配合 M 使用)求带权值时施加四舍五入处理
D ：两个寄存器按两对半字处理，进行乘加或乘减
AD ：两个寄存器按两对半字处理，进行乘加
SD ：两个寄存器按两对半字处理，进行乘减
X ：（必须和 D 联用）两个寄存器的高低半字交叉相乘
L ：和/差是 64 位整数

1.2 APSR.GE[3:0]

APSR.GE[3:0]是 4 个专用于 SIMD 的标志位，对应于 4 个字节。当 SIMD 以 16 位处理时，则 GE[1:0]/GE[3:2]同时设置或清除。

1.3 指令功能详解

下文中每一类的指令只有第一次出现时列出详细的操作数和指令功能，其它同类指令则只给出指令名，希望大家触类旁通。在详解时，使用以下简写符号：
单个字母，d，n，m，a：表示 32 位寄存器 Rd，Rn，Rm，Ra
后缀 t，如 dt,nt,mt 等：表示相应寄存器的高 16 位(Top)。
后缀 b，如 db,nb,mb 等：表示相应寄存器的低 16 位(Bottom)。
后缀 L,m,M,H:分别表示寄存器的[7:0]，[15:8]，[23:16]，[31:24] 4 个字节。
后缀.H#，如 m.H#：表示相应寄存器的高半字和低半字，意指 SIMD 指令同时处理 2 个 16 位整数。
后缀.B#，如 m.B#：表示相应寄存器中的某个字节，意指 SIMD 指令同时处理 4 个 8 位整数。
Q+，Q*：饱和加法符号，饱和乘法符号
符号“ || ”：表示 SIMD 指令中同时执行的操作，当操作不便以.H#和.B#格式表达时分别列出。

2.饱和算术、SIMD，以及其它位宽变换指令（59 条）

2.1 打包指令（2 条）

PK H B T Rd, Rn, Rm {,LSL #i5}
db = nb, dt = (m<<#i5)t
PK H T B Rd, Rn, Rm {,ASR #i5}
dt = nt, db = (m asr #i5)b

2.2 普通饱和算术指令（4 条）

Q ADD Rd, Rn, Rm
Q SUB
Q D ADD Rd, Rn, Rm
d = n Q+ (2 Q* m)
Q D SUB

2.3 SIMD 普通加减法指令（16 条）

S ADD 8 Rd, Rn, Rm
d.B# = n.B# + m.B#
S ADD 16
d.H# = n.H# + m.H#
U ADD 8
U ADD 16
Q ADD 8
Q ADD 16
U Q ADD 8
U Q ADD 16
S SUB 8
S SUB 16
U SUB 8
U SUB 16
Q SUB 8
Q SUB 16
U Q SUB 8
U Q SUB 16

2.4 SIMD 求平均数加减法指令（8 条）

```
S H ADD 8     Rd,    Rn,    Rm
    d.B# = (n.B# + m.B#) / 2
U H ADD 8
S H ADD 16
U H ADD 16
S H SUB 8
U H SUB 8
S H SUB 16
U H SUB 16
```

2.5 SIMD 半字交叉加减指令（8 条）

```
S ASX      Rd,    Rn,    Rm      (Q Add Sub with Exchange)
    dt = nt + mb  ||  db = nb - mt
U ASX
Q ASX
U Q ASX

S SAX
    dt = nt - mb  ||  db = nb + mt
U SAX
Q SAX
U Q SAX
```

2.6 SIMD 求平均数半字交叉加减指令（4 条）

```
S H ASX
U H ASX
S H SAX
U H SAX
```

2.7 字节与半字位宽扩展指令（6 条）

```
SXT B      Rd,    Rm,    <ror #8/16/24>
    d = SXT(旋转过的 Rm.B0, 32)
UXT B
SXT H      Rd,    Rm,    <ror #8/16/24>
    d = SXT(旋转过的 Rm.H0, 32)
UXT H
SXT B 16   Rd,    Rm,    <ror #8/16/24>
    db = SXT(旋转过的 Rm.B0,16)  ||  dt = SXT(旋转过的 Rm.B2,16)
UXT B 16
```

2.8 带加法的字节与半字位宽扩展指令（6 条）

```
SXT A B    Rd,    Rn,    Rm,    <ror #8/16/24>
    d = Rn + SXT(旋转过的 Rm.B0,32)
UXT A B
SXT A H
    d = Rn + SXT(旋转过的 Rm.H0,32)
UXT A H
SXT A B 16 Rd,       Rn,    Rm,    <ror #8/16/24>
```

```
    db = nb + SXT(旋转过的 Rm.B0,16)  ||  dt = nt + SXT(旋转过的 Rm.B2,16)
UXT A B 16
```

2.9 杂项 SIMD 指令（5 条）

```
SEL         Rd,     Rn,     Rm
    Rd.B# = (APSR.GE<#> ==1) ? n.B# : m.B#
S SAT 16    Rd,     #i4,    Rn
    Rn 的高低半字分别作带符号饱和
U SAT 16
U SAD 8     Rd,     Rn,     Rm
    d = |nL - mL| + |nm - mm| + |nM - mM| + |nH-mH|
U SAD A 8   Rd,     Rn,     Rm,     Ra
    d = |nL - mL| + |nm - mm| + |nM - mM| + |nH - mH| + a
```

3 MAC 指令集（7+35=42 条）

3.1 Cortex-M3 的基本 MAC 指令(7 条)

```
MUL             Rd,     Rn,     Rm
MLA             Rd,     Rn,     Rm,     Ra
MLS
S MUL L         RdLo,   RdHi,   Rn,     Rm
U MUL L
S MLA L
U MLA L
```

3.2 计算单次 16 位积的 MAC 指令（8 条）

```
S MUL B/T B/T   Rd,     Rn,     Rm
    d = n.H? * m.H?
S MLA B/T B/T   Rd,     Rn,     Rm,     Ra
    d = n.H? * m.H? + a
```

3.3 计算两次 16 位积之和差的 MAC 指令（8 条）

```
S MU AD         Rd,     Rn,     Rm
    d = nb*mb + nt*mt
S MU AD X
    d = nb*mt + nt*mb
S MU SD
S MU SD X
S MLA D         Rd,     Rn,     Rm,     Ra
    d = nb*mb + nt*mt + a
S MLA D X
S MLS D         Rd,     Rn,     Rm,     Ra
    d = nb*mb - nt*mt + a
S MLS D X
```

3.4 64 位和差的 MAC 指令（8 条）

```
S MLA L D       RdLo,RdHi,      Rn,     Rm
```

```
    d64 = nb*mb + nt*mt + d64
S MLS L D
S MLA L D X
S MLS L D X
S MLA L B/T B/T  RdLo,RdHi,    Rn,     Rm
    n.H? * m.H? + d64 = d64
```

3.5 用于加减某数带权值的 MAC 指令（10 条）

```
S MUL W B/T       Rd,     Rn,     Rm
    d = ((n * m.H?)>>16)
S MLA W B/T       Rd,     Rn,     Rm,     Ra
    d = ((n * m.H?)>>16) + a
S M MUL           Rd,     Rn,     Rm
    d = (n*m)>>32
```

用途：求出 Rn 的带权值，权范围： 0 至$(2^{32}-1)/2^{32}$

```
S M MUL R         SMMUL 的带四舍五入版本
    d = ((n*m+2^31)>>32
S M MLA           Rd,     Rn,     Rm,     Ra
    d = ((n*m)>>32) + a
S M MLA R         SMMLA 的带四舍五入版本
S M MLS
    d = -((n*m)>>32) + a
S M MLS R
```

3.6 其它 MAC 指令（1 条）

```
U MAA L           RdHi,  RdLo,  Rn,     Rm
    d64 = n * m + dHi + dLo
```