Assembly Programming Overview (1A)

Young Won Lim
6/5/18

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice.

Based on

ARM System-on-Chip Architecture, 2nd ed, Steve Furber

Data Processing Instruction Rules

All 32-bit operands
- Come from registers
- Are specified as literals in the instruction itself

The 32-bit result, if any
- Is placed in a register

The 64-bit result from long multiply instructions

3-address format
- Each of the operand and the result register are independently specified

Arithmetic Operations

```
ADD    r0, r1, r2          ;    r0   :=   r1 + r2
ADC    r0, r1, r2          ;    r0   :=   r1 + r2 + C
SUB    r0, r1, r2          ;    r0   :=   r1 – r2
SBC    r0, r1, r2          ;    r0   :=   r1 – r2 + C – 1
RSB    r0, r1, r2          ;    r0   :=   r2 + r1
RSC    r0, r1, r2          ;    r0   :=   r2 + r1 + C – 1
```

Bit-wise Logical Operations

```
AND     r0, r1, r2          ;   r0   :=   r1 and r2
ORR     r0, r1, r2          ;   r0   :=   r1 or r2
EOR     r0, r1, r2          ;   r0   :=   r1 xor r2
BIC     r0, r1, r2          ;   r0   :=   r1 and not r2
```

Register Movement Operations

```
MOV     r0, r2          ;    r0   :=   r2
MVN     r0, r2          ;    r0   :=   not r2
```

Comparison Operations

```
CMP     r1, r2          ;   set cc on r1 – r2
CMN     r1, r2          ;   set cc on r1 + r2
TST     r1, r2          ;   set cc on r1 and r2
TEQ     r1, r2          ;   set cc on r1 xor r2
```

Immediate Operands

```
ADD     r3, r3, #1          ;    r3   := r3 + 1
AND     r8, r7, #&ff        ;    r8   := r7[7:0]
```

Most valid immediate value

Immediate = (0 → 255) x 2^{2n},   n : [0, 12]

Shifted Register Operands

```
ADD     r3, r2, r1 , LSL #3     ;     r3   := r2 + 8*r1
```

LSL (Logical Shift Left)
LSR (Logical Shift Right)
ASL (Arithmetic Shift Left)
ASR (Arithmetic Shift Right)
ROR (Rotate Right by 0)
ROX (Rotate Right Extended by 1)

Setting the Condition Codes

The comparison instructions only set cc
All other data processing instructions must make explicit request
S: Set condition codes

```
ADDS    r2, r2, r0              ;       32-bit carry out → C
ADC     r3, r3, r1              ;       and added into high word
```

Multiplies

```
MUL      r4, r3, r2                 ;      r4    := (r3 x r2)[31:0]
```

Immediate second operand are not supplied
The result register must not be the same as the first source register
If the S bit set
    the V flag is preserved and
    the C flag is rendered meaningless

References

[1] ftp://ftp.geoinfo.tuwien.ac.at/navratil/HaskellTutorial.pdf
[2] https://www.umiacs.umd.edu/~hal/docs/daume02yaht.pdf