The ARM processor (Thumb-2), part 7: Bitwise operations

devblogs.microsoft.com/oldnewthing/20210608-00

June 8, 2021

Raymond Chen

The ARM processor offers the following bitwise operations:

```
; bitwise and
and     Rd, Rn, op2         ; Rd = Rn & op2

; bitwise or
orr     Rd, Rn, op2         ; Rd = Rn | op2

; bitwise exclusive or
eor     Rd, Rn, op2         ; Rd = Rn ^ op2

; bitwise not
mvn     Rd, op2             ; Rd = ~op2

; bitwise and not ("bit clear")
bic     Rd, Rn, op2         ; Rd = Rn & ~op2

; bitwise or not
orn     Rd, Rn, op2         ; Rd = Rn | ~op2

; all support the S suffix
```

For bit-testing purposes, there are also discarding versions:

```
; test for equivalence
teq     Rn, op2             ; set flags for Rn ^ op2

; test
tst     Rn, op2             ; set flags for Rn & op2
```

For bitwise operations that set flags, the negative (N) and zero (Z) flags reflect the result, the carry (C) flag reflects any shifting that occurred during the calculation of `op2` (noting that calculating constants may also involve shifting, as noted earlier when we discussed constants), and the overflow (V) flag is unchanged.

I don't see much value in the `TEQ` instruction. It sets the Z flag the same way as the the `CMP` instruction. I guess you could use it to see if two registers have the same sign bit, since it sets N based on the exclusive-or of the two inputs. I guess that's handy when calculating the sign of emulated multiplication or division, but even in those cases, you aren't going to jump based on the sign; you're going to save the sign of the result for later application, so you would be better off with the `EOR` instruction anyway.

Okay, well, you can use the `LSL` shift on the second register argument in order to compare the high bit of one register with an arbitrary bit of another.

```
    teq     Rn, Rm LSL #n       ; compare Rn bit 31
                                ; and Rm bit 31 - n
    bmi     same                ; branch if different
```

Still not particularly compelling. Maybe there's some specialized workflow where this is useful, like cryptography?

Next time, we'll look at the bit shifting instructions.

Raymond Chen

Follow