Formal Analysis and Proof of the `NMOD_RED2` Reduction Algorithm

1 Introduction and Setup

Let $W = b$ be the machine word size in bits, and let the radix be

$$B = 2^b.$$

We are given a two-word dividend

$$x = u_1 B + u_0, \qquad 0 \le u_1, u_0 < B,$$

and a pre-normalized divisor n satisfying

$$\frac{B}{2} \le n < B.$$

We write

$$n = \frac{B}{2} + k, \qquad 0 \le k < \frac{B}{2}.$$

The goal is to compute $x \bmod n$.

2 Precomputation of the Inverse

The algorithm uses the Möller–Granlund pseudo-inverse

$$v = \left\lfloor \frac{B^2 - 1}{n} \right\rfloor - B.$$

In the C implementation, this is computed by a two-limb-by-one-limb division:

```
udiv_qrnnd(ninv, r, ~n, ~UWORD(0), n);
```

The bitwise complements construct the dividend

$$D = (B - 1 - n)B + (B - 1) = B^2 - 1 - nB.$$

Hence the computed quotient is

$$v = \left\lfloor \frac{B^2 - 1 - nB}{n} \right\rfloor = \left\lfloor \frac{B^2 - 1}{n} \right\rfloor - B.$$

Lemma 1. *Assume $k \ge 1$ and $16k^2 \le B$. Then*

$$v = B - 4k.$$

Proof. Using $n = B/2 + k$, we compute

$$(B/2 + k)(2B - 4k) = B^2 - 4k^2,$$

so

$$B^2 - 1 = (B/2 + k)(2B - 4k) + (4k^2 - 1) = n(2B - 4k) + (4k^2 - 1).$$

Since $k \geq 1$, we have

$$4k^2 - 1 \geq 3 > 0.$$

Also, from $16k^2 \leq B$,

$$4k^2 \leq \frac{B}{4},$$

and therefore

$$4k^2 - 1 < \frac{B}{4} < \frac{B}{2} \leq n.$$

Thus $4k^2 - 1$ is a valid remainder on division by n, so

$$\left\lfloor \frac{B^2 - 1}{n} \right\rfloor = 2B - 4k.$$

Hence

$$v = \left\lfloor \frac{B^2 - 1}{n} \right\rfloor - B = B - 4k.$$

□

When $k = 0$, we have $n = B/2$, and directly

$$v = \left\lfloor \frac{B^2 - 1}{B/2} \right\rfloor - B = \left\lfloor 2B - \frac{2}{B} \right\rfloor - B = B - 1.$$

This is a trivial special case. In the main proof below we assume $k \geq 1$.

3 The `NMOD_RED2` Algorithm

The reduction algorithm is:

1. Compute
$$P = u_1 v + u_1 B + u_0.$$
2. Write
$$P = q_1 B + q_0, \qquad 0 \leq q_0 < B.$$
3. Compute the machine-word value
$$r_1 \equiv u_0 - (q_1 + 1)n \pmod{B}.$$
4. If $r_1 > q_0$, replace r_1 by
$$r_1 \equiv r_1 + n \pmod{B}.$$
5. Return r_1 if $r_1 < n$, otherwise return $r_1 - n$.

All congruences modulo B are to be understood as ordinary unsigned word arithmetic.

4 Theorem

Theorem 2. *Let*

$$n = \frac{B}{2} + k$$

with

$$0 \le k \le \left\lfloor \sqrt{\frac{B}{16}} \right\rfloor .$$

Equivalently, assume

$$16k^2 \le B.$$

Then for every

$$0 \le x < B^2,$$

the `NMOD_RED2` *algorithm returns* $x \bmod n$.

5 Proof

5.1 Word decomposition of the product

By Lemma 1, for $k \ge 1$ we have

$$v = B - 4k.$$

Hence

$$P = u_1(B - 4k) + u_1 B + u_0 = 2u_1 B - 4ku_1 + u_0.$$

Define

$$\gamma = \left\lceil \frac{4ku_1 - u_0}{B} \right\rceil .$$

Then

$$q_1 = 2u_1 - \gamma, \qquad q_0 = u_0 - 4ku_1 + \gamma B,$$

and by construction

$$P = q_1 B + q_0.$$

Moreover, from the definition of γ we have

$$0 \le q_0 < B.$$

We also need a bound on γ. Since $0 \le u_1, u_0 < B$,

$$\frac{-(B-1)}{B} \le \frac{4ku_1 - u_0}{B} \le \frac{4k(B-1)}{B}.$$

Therefore

$$0 = \left\lceil \frac{-(B-1)}{B} \right\rceil \le \gamma \le \left\lceil \frac{4k(B-1)}{B} \right\rceil \le 4k.$$

So

$$0 \le \gamma \le 4k.$$

5.2 The exact remainder attached to the quotient estimate

Let

$$r^* = x - q_1 n.$$

This is the exact mathematical remainder associated to the quotient estimate q_1.

Substituting $x = u_1 B + u_0$, $q_1 = 2u_1 - \gamma$, and $n = B/2 + k$ gives

$$\begin{aligned} r^* &= (u_1 B + u_0) - (2u_1 - \gamma)\left(\frac{B}{2} + k\right) \\ &= u_1 B + u_0 - u_1 B - 2ku_1 + \frac{\gamma B}{2} + \gamma k \\ &= u_0 - 2ku_1 + \frac{\gamma B}{2} + \gamma k. \end{aligned}$$

From the definition of q_0,

$$q_0 = u_0 - 4ku_1 + \gamma B,$$

so

$$2ku_1 = \frac{u_0 - q_0 + \gamma B}{2}.$$

Substituting into the previous expression yields

$$r^* = u_0 - \frac{u_0 - q_0 + \gamma B}{2} + \frac{\gamma B}{2} + \gamma k = \frac{u_0 + q_0}{2} + \gamma k.$$

Thus

$$r^* = \frac{u_0 + q_0}{2} + \gamma k. \tag{1}$$

Since $0 \le u_0, q_0 \le B - 1$ and $\gamma \le 4k$, we obtain

$$r^* \le \frac{(B-1) + (B-1)}{2} + 4k^2 = (B-1) + 4k^2.$$

Using $16k^2 \le B$, we get

$$r^* < B + 4k^2 \le B + \frac{B}{4} = \frac{5B}{4}.$$

Since $n \ge B/2$,

$$3n \ge \frac{3B}{2} > \frac{5B}{4},$$

so

$$r^* < 3n. \tag{2}$$

Therefore the only possibilities are:

$$0 \le r^* < n, \qquad n \le r^* < 2n, \qquad 2n \le r^* < 3n.$$

Equivalently, the quotient estimate q_1 can differ from the true quotient by at most 2.

5.3 The machine-word remainder used by the algorithm

The algorithm first computes

$$r_1 \equiv u_0 - (q_1 + 1)n \pmod{B}.$$

Since

$$x = u_1 B + u_0 \equiv u_0 \pmod{B},$$

we have

$$r_1 \equiv x - (q_1 + 1)n = (x - q_1 n) - n = r^* - n \pmod{B}.$$

So the machine-word value stored in r_1 is the reduction modulo B of $r^* - n$.

We now analyze the three possible ranges for r^*.

Case 0: $0 \le r^* < n$

In this case q_1 is the true quotient. Then

$$-n \le r^* - n < 0.$$

Since $n < B$, the machine-word value is

$$r_1 = B + r^* - n.$$

We claim that the condition $r_1 > q_0$ is always true.

Using (1),

$$\begin{aligned} r_1 > q_0 &\iff B + r^* - n > q_0 \\ &\iff B + \frac{u_0 + q_0}{2} + \gamma k - \left(\frac{B}{2} + k\right) > q_0 \\ &\iff \frac{B}{2} + \frac{u_0 - q_0}{2} + k(\gamma - 1) > 0. \end{aligned}$$

If $\gamma \ge 1$, then $k(\gamma - 1) \ge 0$, and since

$$u_0 - q_0 > -(B - 1),$$

we have

$$\frac{B}{2} + \frac{u_0 - q_0}{2} > \frac{1}{2} > 0.$$

Hence $r_1 > q_0$.

If $\gamma = 0$, then by definition

$$4ku_1 - u_0 \le 0,$$

so $4ku_1 \le u_0$. Since $\gamma = 0$,

$$q_0 = u_0 - 4ku_1,$$

and therefore

$$u_0 - q_0 = 4ku_1 \ge 0.$$

Thus

$$\frac{B}{2} + \frac{u_0 - q_0}{2} + k(\gamma - 1) = \frac{B}{2} + 2ku_1 - k \ge \frac{B}{2} - k.$$

Because $k \geq 1$ and $16k^2 \leq B$, we have

$$\frac{B}{2} \geq 8k^2 > k,$$

so this quantity is strictly positive. Hence again $r_1 > q_0$.

Therefore the first correction is always applied in Case 0. After that correction, the stored machine-word value becomes

$$r_1 \equiv (r^* - n) + n \equiv r^* \pmod{B}.$$

Since $0 \leq r^* < n < B$, the stored value is in fact exactly

$$r_1 = r^*.$$

The final test $r_1 < n$ is true, so the algorithm returns $r^* = x \bmod n$.

Case 1: $n \leq r^* < 2n$

In this case q_1 is smaller than the true quotient by 1. Then

$$0 \leq r^* - n < n < B,$$

so the initial machine-word value is exactly

$$r_1 = r^* - n.$$

There are two subcases.

If $r_1 \leq q_0$, the first correction is skipped. Since $0 \leq r_1 < n$, the final test returns

$$r_1 = r^* - n = x \bmod n.$$

If $r_1 > q_0$, the first correction is applied, and the new machine-word value is

$$r_1 = r^*.$$

Since here $r^* \geq n$, the final branch returns

$$r_1 - n = r^* - n = x \bmod n.$$

So the algorithm is correct throughout Case 1.

Case 2: $2n \leq r^* < 3n$

In this case q_1 is smaller than the true quotient by 2. Since $r^* < 3n$ and $n \geq B/2$,

$$r^* - n < 2n \leq 2B.$$

More precisely, using $r^* < 5B/4$,

$$r^* - n < \frac{5B}{4} - \frac{B}{2} = \frac{3B}{4} < B.$$

Also $r^* - n \geq n > 0$. Hence the initial machine-word value is exactly

$$r_1 = r^* - n, \qquad n \leq r_1 < B.$$

For the algorithm to be correct, the first correction must *not* be taken, because otherwise the stored value would become r^* and the final subtraction would return $r^* - n$ instead of the correct $r^* - 2n$. We therefore prove that

$$r_1 > q_0$$

is impossible in this case.

Assume for contradiction that $r_1 > q_0$. Since $r_1 = r^* - n$, this means

$$r^* - n > q_0.$$

Using (1),

$$\frac{u_0 + q_0}{2} + \gamma k - \left(\frac{B}{2} + k\right) > q_0 \iff u_0 - q_0 + 2\gamma k > B + 2k.$$

So we have

$$u_0 - q_0 + 2\gamma k > B + 2k. \tag{3}$$

On the other hand, the assumption $r^* \geq 2n$ gives

$$\frac{u_0 + q_0}{2} + \gamma k \geq B + 2k,$$

hence

$$u_0 + q_0 + 2\gamma k \geq 2B + 4k. \tag{4}$$

Adding (3) and (4) cancels q_0:

$$2u_0 + 4\gamma k > 3B + 6k.$$

Since $u_0 \leq B - 1$, we have $2u_0 < 2B$, so this implies

$$2B + 4\gamma k > 3B + 6k,$$

that is,

$$4\gamma k > B + 6k.$$

But $\gamma \leq 4k$, so

$$4\gamma k \leq 16k^2 \leq B,$$

which contradicts

$$4\gamma k > B + 6k > B.$$

This contradiction shows that $r_1 > q_0$ is impossible. Therefore the first correction is never applied in Case 2.

So the stored value remains

$$r_1 = r^* - n.$$

Since $r_1 \geq n$, the final branch returns

$$r_1 - n = r^* - 2n = x \bmod n.$$

5.4 Conclusion

All three possible ranges for r^* have been analyzed, and in each case the algorithm returns $x \bmod n$. This proves the theorem.

□