Notes on Non-restoring Unsigned Division

Mainak Chaudhuri

This note discusses the details of the extra special iteration that may be needed at the end of the non-restoring unsigned division algorithm in the case when the remainder is negative. First, we would like to recall the equivalence of the non-restoring and restoring division algorithms during the regular iterations. Suppose the remainder is r and the divisor is d in both algorithms at the end of a certain number n of iterations.

Case I: Suppose $r - d < 0$, but $r - d/2 \geq 0$. So, in the $(n+1)^{th}$ iteration of restoring division, a restoring addition will take place and at the end of the $(n+2)^{th}$ iteration, the restoring division will have remainder equal to $r - d/2$. In the non-restoring division, the remainder at the end of the $(n+1)^{th}$ iteration would be $r - d$ and at the end of the $(n+2)^{th}$ iteration, the remainder would be $r - d + d/2$ or $r - d/2$. Thus, both the algorithms are now in sync.

Case II: Suppose $r-d, r-d/2, \ldots, r-d/2^k$ are all negative, but $r - d/2^{k+1} \geq 0$. So, in iterations $(n+1), (n+2), \ldots, (n+k+1)$, restoring additions will take place in the restoring division algorithm and the remainder will remain r at the end of the $(n+k+1)^{th}$ iteration. At the end of the $(n+k+2)^{th}$ iteration, the remainder will be $r - d/2^{k+1}$ in the restoring division algorithm. In the non-restoring division algorithm, the $(n+1)^{th}$ iteration produces remainder $r - d$, which is negative; the $(n+2)^{th}$ iteration produces remainder $r - d + d/2$ or $r - d/2$, which is negative; the $(n+3)^{th}$ iteration produces remainder $r - d/2 + d/4$ or $r - d/4$, which is negative. The additions continue and the $(n+k+1)^{th}$ iteration produces remainder $r - d/2^k$, which is still negative. At this point, the divisor is $d/2^{k+1}$. The $(n+k+2)^{th}$ iteration of non-restoring division produces remainder $r - d/2^k + d/2^{k+1}$ or $r - d/2^{k+1}$, which is non-negative. Thus, both the algorithms are now in sync. Case I is a special case of Case II where $k = 0$.

Now we discuss the case where the adequate number of iterations in the non-restoring division algorithm has finished, but the remainder is still negative. Consider the latest iteration in the non-restoring division algorithm when the remainder was non-negative. Suppose the remainder was $r \geq 0$ at that point and the divisor was d. From the next iteration onward, the remainder became negative i.e., $r-d, r-d/2, r-d/4, \ldots$ are all negative. Let us also suppose that when all iterations have completed, the remainder is $r - d/2^k$. Since this is the last iteration, the divisor must equal the original divisor at this point i.e., $d/2^k$ is the original divisor. We do not shift it to right any further because then we will start losing bits of the original divisor. Now, observe that in the restoring division algorithm, the last k iterations would have done the restoring additions because $r-d, r-d/2, \ldots, r-d/2^k$ are all negative. Therefore, the remainder in the restoring division algorithm would have stayed at r and this is the correct remainder. So, it is clear what needs to be done in the extra special iteration of the non-restoring division: take the negative remainder $r - d/2^k$ and add the original divisor i.e., $d/2^k$ to get the final remainder r.