Reduced Restoring Divider

Akshay Tukaram Rao

August 16, 2024

Contents

1 Reduced Restoring Divider

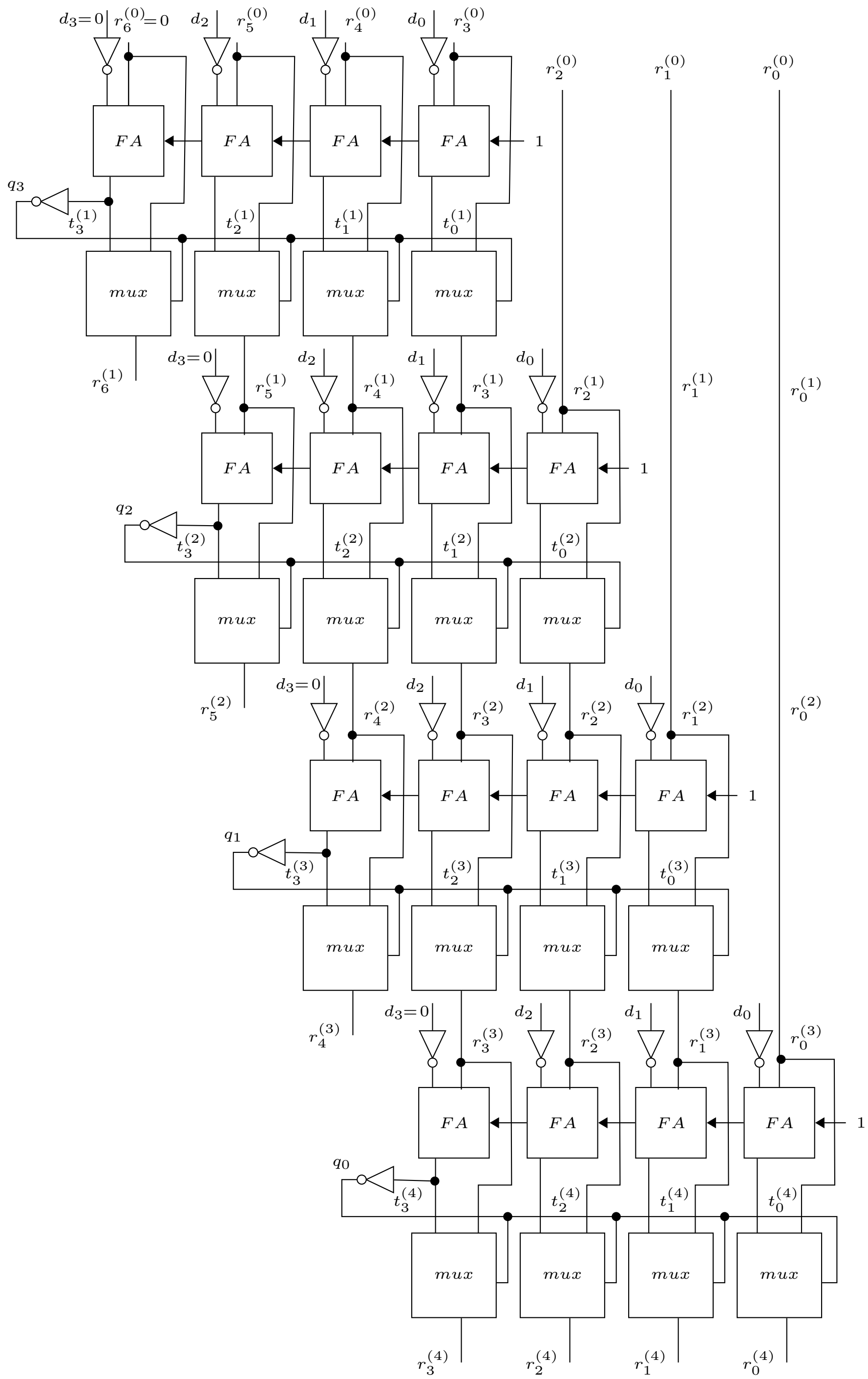

Figure 1: Reduced restoring divider with multiplexers, n = 4.

2 Proof for Don't Care Assignment

2.1 To prove that $(t_{n-1}^{(n)}, r_{n-1}^{(n-1)}) = (1, 1)$ is a Don't Care

Proof 1: To prove that $(t_{n-1}^{(n)}, r_{n-1}^{(n-1)}) = (1, 1)$ is a Don't Care

For all restoring dividers with the input constraint $0 \leq R^{(0)} < D.2^{n-1}$, the following holds [1]:

$$0 \leq R^{(j)} < D.2^{n-j} \tag{1}$$

The above constraint is extended to the final reminder using j = n.

$$0 \leq R^{(n)} < D. \tag{2}$$

From Equations 1 and 2, the partial reminder at any stage should be a positive integer. $R^{(n)}$ is expressed as $\sum_{i=0}^{n-2} r_i^{(n)} 2^{(i)} - 2^{(n-1)} r_{n-1}^{(n)}$ (The current proof concerns the most significant multiplexer of the stage n. The output of this mux is $r_{n-1}^{(n)}$, where $r_{n-1}^{(n)}$ is a signed bit in 2's complement representation. Therefore, it is sufficient to check if $R^{(n)} \geq 0$, as $r_{n-1}^{(n)} = 1$ would make $R^{(n)}$ negative, which then violates Equation 2.

if $(t_{n-1}^{(n)}, r_{n-1}^{(n-1)}) = (1, 1)$, then $r_{n-1}^{(n)} = 1$ (from figure 1) regardless of $q_0 = 0$ or 1 . This makes $R^{(n)} < 0$, which contradicts equation 2. Hence $(t_{n-1}^{(n)}, r_{n-1}^{(n-1)}) = (1, 1)$ is a true don't care assignment.

2.2 Proof 2 : To prove that $(r_{n-1}^{(n-1)}, c_{n-2}^{(n)}) = (1,1)$ is don't Care

Proof 2: To prove that $(r_{n-1}^{(n-1)}, c_{n-2}^{(n)}) = (1,1)$ is a don't Care at the most significant adder of stage n.

To prove $(r_{n-1}^{(n-1)}, c_{n-2}^{(n)}) = (1,1)$ is truly a don't care of full adder (FA_{n-1}), assume that for all j=1 : n-1 , the following holds:

$$0 \leq R^{(j)} < D.2^{n-j} \tag{3}$$

$$-D.2^{n-j} \leq T^{(j)} < D.2^{n-j} \tag{4}$$

Since $\overline{d}_{n-1}$ is always equal to 1, proving $(t_{n-1}^{(n)}, c_{n-1}^{(n)}) \neq (1,1)$ implies $(r_{n-1}^{(n-1)}, c_{n-2}^{(n)}) = (1,1)$ is a true don't care.

$(t_{n-1}^{(n)}, c_{n-1}^{(n)})$ = (1,1) implies $R^{n-1} \geq D + 2^{n-1}$.This can be proven by considering the following equation derived from the carry ripple adder, which performs the subtraction between $R^{(n-1)}$ and D.

$$2^n c_{n-1}^{(n)} + \sum_{i=0}^{n-1} t_i^{(n)} 2^i = \sum_{i=0}^{n-1} r_i^{(n-1)} 2^i + \sum_{i=0}^{n-1} \overline{d}_i 2^i + c_{-1} \tag{5}$$

In above Equation, substituting $(t_{n-1}^{(n)}, c_{n-1}^{(n)}) = (1,1)$ and ignoring the values of terms i = 0 : n-2 in $\sum_{i=0}^{n-1} t_i^{(n)} 2^i$ result in :

$$2^n + 2^{n-1} \leq \sum_{i=0}^{n-1} r_i^{(n-1)} 2^i + \sum_{i=0}^{n-1} \overline{d}_i 2^i + c_{-1} \tag{6}$$

substituting $c_{-1} = 1$, $R^{(n-1)} = \sum_{i=0}^{n-1} r_i^{(n-1)} 2^i$(since $r_n^{(n-1)} = 0$), $\overline{d}_i = 1 - d_i$ and $D = \sum_{i=0}^{n-1} d_i 2^i$ in the above equation:

$$2^n + 2^{n-1} \leq R^{(n-1)} - D + 2^n \tag{7}$$

Simplifying the above equation proves $R^{(n-1)} \geq D + 2^{n-1}$ if $(t_{n-1}^{(n)}, c_{n-1}^{(n)}) = $ (1,1).

From the above Equation, it is clear that $R^{(n-1)} < 2.D$. if $R^{n-1} \geq D + 2^{n-1}$, then $D + 2^{n-1} < 2.D$ which makes $D > 2^{n-1}$. This contradicts the constraint: $1 \leq D \leq 2^{n-1} - 1$.Therefore, as long as R^{n-1} obeys $0 \leq R^{n-1} < 2.D$, $(t_{n-1}^{(n)}, c_{n-1}^{(n)}) \neq (1,1)$. $(r_{n-1}^{(n-1)}, c_{n-2}^{(n)}) = (1,1)$ is the only combination that gives rise to $(t_{n-1}^{(n)}, c_{n-1}^{(n)}) = (1,1)$. Hence $(r_{n-1}^{(n-1)}, c_{n-2}^{(n)}) = (1,1)$ is a true don't care.

2.3 Proof 3: To prove $(\overline{d}_{n-2}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)}) = (1,1,1)$ is a don't Care

Proof 3: To prove $(\overline{d}_{n-2}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)}) = (1,1,1)$ is a don't care of the second most significant adder (FA_{n-2}).

lemma 1 : $\overline{d}_{n-2} = 1/d_{n-2} = 0$ implies $r_{n-1}^{(n-1)} = 0$.

This is true due to the constraint: $R^{n-1} < 2.D$. R^{n-1} is expressed as $\sum_{i=0}^{n-1} r_i^{(n-1)} 2^i - 2^n r_n^{(n-1)}$ and D is expressed as $\sum_{i=0}^{n-1} d_i 2^i$. Now, assume that $r_{n-1}^{(n-1)} = 1$ with all other terms in $R^{(n-1)}$ equal to zero and similarly, all the terms: i = 0 to n-3 is equal to one in D. Since d_{n-1} and d_{n-2} are equal to zero and from the assumption it is clear that $R^{n-1} > 2.D$ $(2^{n-1} r_{n-1}^{(n-1)} > \sum_{i=0}^{n-3} d_i 2^i)$. Hence, to satisfy the constraint $R^{n-1} < 2.D$, $r_{n-1}^{(n-1)} = 0$ when $d_{n-2} = 0$

For $\overline{d}_{n-2} = 1/d_{n-2} = 0$, the following condition must hold:

$$1 \leq D \leq 2^{n-2} - 1 \tag{8}$$

R^{n-1} should hold the following constraint:

$$0 \leq R^{n-1} < 2^{n-1} - 2 \tag{9}$$

If and when equations (8) and (9) are true, the last stage of an n-bit restoring divider with 0 to n-1 full adders can be reduced to 0 to n-2 full adders ($\because$ *lemma 1* and $d_{n-1} = 0$). The same argument as in Proof 2 is applicable here as well.

$(t_{n-2}^{(n)}, c_{n-2}^{(n)}) = (1, 1)$ implies $R^{(n-1)} \geq D + 2^{n-2}$. This can be proven using the following equation derived from the reduced carry ripple adder that subtracts D from R^{n-1}.

$$2^{n-1}c_{n-2}^{(n)} + \sum_{i=0}^{n-2} t_i^{(n)} 2^i = \sum_{i=0}^{n-2} r_i^{(n-1)} 2^i + \sum_{i=0}^{n-2} \overline{d}_i 2^i + c_{-1} \tag{10}$$

Using $(t_{n-2}^{(n)}, c_{n-2}^{(n)}) = (1,1)$, $c_{-1} = 1$, $R^{(n-1)} = \sum_{i=0}^{n-2} r_i^{(n-1)} 2^i$(since $r_n^{(n-1)}, r_{n-1}^{(n-1)} = 0$), $\overline{d}_i = 1 - d_i$, $D = \sum_{i=0}^{n-2} d_i 2^i$ and ignoring the values of all terms i = 0: n-3 in $\sum_{i=0}^{n-2} t_i^{(n)} 2^i$ in the above equation, this results in the following.

$$2^{n-1} + 2^{n-2} \leq R^{(n-1)} - D + 2^{n-1} \tag{11}$$

The above equation simplifies to $D + 2^{n-2} \leq R^{(n-1)}$.From Equation 1, it is clear that $R^{(n-1)} < 2.D$. if $R^{n-1} \geq D + 2^{n-2}$, then $D + 2^{n-2} < 2.D$ which makes $D > 2^{n-2}$. This contradicts Equation (8). Therefore, as long as R^{n-1} obeys $0 \leq R^{n-1} < 2.D$, $(t_{n-2}^{(n)}, c_{n-2}^{(n)}) \neq (1, 1)$. $(\overline{d}_{n-2}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)}) = (1, 1, 1)$ is the only combination that gives rise to $(t_{n-2}^{(n)}, c_{n-2}^{(n)}) = (1, 1)$.Hence it can be concluded that $(\overline{d}_{n-2}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)}) = (1, 1, 1)$ is a true don't care.

2.4 Proof 4 : To prove that $(t_{n-2}^{(n)}, q_0, r_{n-2}^{(n-1)})$=(0,0,1) is a Don't Care

Proof 4: To prove that $(t_{n-2}^{(n)}, q_0, r_{n-2}^{(n-1)})$=(0,0,1) is a Don't Care of the second most significant multiplexer (Mux_{n-2})

There are four possible combinations at the input of FA_{n-2} that would result in $t_{n-2}^{(n)} = 0$. They are $(\overline{d_{n-2}}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)}) = (0,0,0), (0,1,1), (1,0,1), (1,1,0)$. In these four combinations, only two combinations are of interest as the proof requires $r_{n-2}^{(n-1)} = 1$. Hence, there are two cases:

Case 1 : $(\overline{d_{n-2}}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)}) = (1,1,0)$

Case 2 : $(\overline{d_{n-2}}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)}) = (0,1,1)$

In both the cases it clear that $c_{n-2}^{(n)} = 1$, which makes $r_{n-1}^{(n-1)} = 0$ (Due to Proof 2.2) and this results in $t_{n-1}^{(n)} = 0$ and $q_0 = 1$ (Since $q_0 = 1 - t_{n-1}^{(n)}$). Hence it is clearly seen that $q_0 \neq 0$ when $t_{n-2}^{(n)} = 0$ and $r_{n-2}^{(n-1)} = 1$. Therefore $(t_{n-2}^{(n)}, q_0, r_{n-2}^{(n-1)})$=(0,0,1) is a Don't Care.

2.5 Proof 5: proof of Care Assignments

Proof 5.1: To prove that $(\overline{d}_{n-2}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)})$ = (0,0,0) , $(r_{n-1}^{(n-1)}, c_{n-2}^{(n)})$ = (1,0) and $(t_{n-1}^{(n)}, r_{n-1}^{(n-1)}) = (0,1)$ are care assignments.

$D = 2^{n-1} - 1$ and $R^{n-1} = 2^{n-1}$ will give rise to the combinations mentioned in 5.1. This condition also satisfies the constraints $1 \leq D < 2^{n-1}$ and $0 \leq R^{n-1} < 2.D$ ($\because 2^{n-1} < 2^n - 2$). From figure 1, the following equation holds:

$$R^{n-1} - D = \sum_{i=0}^{n-1} r_i^{(n-1)} 2^i + \sum_{i=0}^{n-1} \overline{d}_i 2^i + c_{-1} \tag{12}$$

In the above equation $\sum_{i=0}^{n-1} r_i^{(n-1)} 2^i$, $\sum_{i=0}^{n-1} \overline{d}_i 2^i$ are denoted as $R^{(n-1)}$ and $\overline{D}$ ($\overline{D}$ is the negated representation of D) respectively. For $R^{n-1} = 2^{n-1}$, all the bits in $R^{(n-1)}$ are equal to zero except $r_{n-1}^{(n-1)} = 1$. Similarly for $D = 2^{n-1} - 1$, all the bits in $\overline{D}$ are zero except $\overline{d}_{n-1} = 1$. As $r_0^{(n-1)}, \overline{d}_0 = 0$ and $c_{-1} = 1$, c_0 will be equal to 0. $c_0 = 0$ and all bits upto n-1 equal zero in $R^{(n-1)}$, $\overline{D}$ respectively results in $c_{n-3}^{(n)} = 0$. Hence $(\overline{d}_{n-2}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)}) = (0,0,0)$ is a possible combination and a care assignment.

The combination of $(\overline{d}_{n-2}, r^{(n-1)}_{n-2}, c^{(n)}_{n-3}) = (0, 0, 0)$ also results in $c^{(n)}_{n-2} = 0$. Since $r^{(n-1)}_{n-1} = 1$(From the above discussion) and $c^{(n)}_{n-2} = 0$, $(r^{(n-1)}_{n-1}, c^{(n)}_{n-2}) = (1, 0)$ is possible combination and a care assignment.$(r^{(n-1)}_{n-1}, c^{(n)}_{n-2}) = (0, 0)$ extends to $(t^{(n)}_{n-1}, r^{(n-1)}_{n-1}) = (0, 1)$ (See figure 1).

Proof 5.2: To prove that $(\overline{d}_{n-2}, r^{(n-1)}_{n-2}, c^{(n)}_{n-3})$ = (0,0,1) is a care assignment.

$D = 2^{n-1} - 1$ and $R^{n-1} = 2^{n-2} - 1$ will give rise to the combinations mentioned in 5.2. This condition also satisfies the constraints $1 \leq D < 2^{n-1}$ and $0 \leq R^{n-1} < 2.D$ (since $2^{n-2} - 1 < 2^n - 2$). For $R^{n-1} = 2^{n-2} - 1$, all the bits in $R^{(n-1)}$ are 1 except $r^{(n-1)}_{n-1} = r^{(n-1)}_{n-2} = 0$. Similarly for $D = 2^{n-1} - 1$, all the bits in $\overline{D}$ is zero except $\overline{d}_{n-1} = 1$. As $c_{-1} = 1$ and $r^{(n-1)}_0 = 1$,c_0 is equal to 1. $c_0 = 1$ and all bits up to n-2 are equal to 1 in $R^{(n-1)}$ leads to $c^{(n)}_{n-3} = 1$ (See figure 1). Therefore $(\overline{d}_{n-2}, r^{(n-1)}_{n-2}, c^{(n)}_{n-3}) = (0, 0, 1)$ is a possible combination and a care assignment.

Proof 5.3: To prove that $(\overline{d}_{n-2}, r^{(n-1)}_{n-2}, c^{(n)}_{n-3})$ = (0,1,0) , $(r^{(n-1)}_{n-1}, c^{(n)}_{n-2})$ = (0,0) and $(t^{(n)}_{n-1}, r^{(n-1)}_{n-1}) = (1, 0)$ are care assignments.

$D = 2^{n-1} - 1$ and $R^{n-1} = 2^{n-2}$ will give rise to the combinations mentioned in 5.3. This condition also satisfies the constraints $1 \leq D < 2^{n-1}$ and $0 \leq R^{(n-1)} < 2D$ (since $2^{n-2} < 2^n - 2$).For $R^{n-1} = 2^{n-2}$, all the bits in $R^{(n-1)}$ are equal to zero except $r^{(n-1)}_{n-2} = 1$. Similarly for $D = 2^{n-1} - 1$, all the bits in $\overline{D}$ is zero except $\overline{d}_{n-1} = 1$. As $r^{(n-1)}_0 = \overline{d}_0 = 0$ and $c_{-1} = 1$, c_0 is equal to 0.With $c_0 = 0$ and all bits upto n-2 , n-1 being zero in $R^{(n-1)}$, $\overline{D}$ respectively , results in $c^{(n)}_{n-3} = 0$. Hence $(\overline{d}_{n-2}, r^{(n-1)}_{n-2}, c^{(n)}_{n-3}) = (0, 1, 0)$ is a possible combination and a care assignment.

The combination of $(\overline{d}_{n-2}, r^{(n-1)}_{n-2}, c^{(n)}_{n-3}) = (0, 1, 0)$ also results in $c^{(n)}_{n-2} = 0$. Since $r^{(n-1)}_{n-1} = 0$(From the above discussion) and $c^{(n)}_{n-2} = 0$, $(r^{(n-1)}_{n-1}, c^{(n)}_{n-2}) = (0, 0)$ is possible combination and a care assignment.$(r^{(n-1)}_{n-1}, c^{(n)}_{n-2}) = (0, 0)$ extends to $(t^{(n)}_{n-1}, r^{(n-1)}_{n-1}) = (1, 0)$ (See figure 1).

Proof 5.4: To prove that $(\overline{d}_{n-2}, r^{(n-1)}_{n-2}, c^{(n)}_{n-3})$ = (0,1,1) , $(r^{(n-1)}_{n-1}, c^{(n)}_{n-2})$ = (0,1) and $(t^{(n)}_{n-1}, r^{(n-1)}_{n-1}) = (0, 0)$ are care assignments.

$R^{(n-1)} = D = 2^{n-1} - 1$ can give rise to the combinations mentioned in 5.4. This combination holds the constraint $1 \leq D < 2^{n-1}$ and $0 \leq R^{n-1} < 2.D$ (Since $R^{n-1} = D$). For $R^{n-1} = 2^{n-1} - 1$, all the bits in $R^{(n-1)}$ are one except $r^{(n-1)}_{n-1} = 0$. Similarly for $D = 2^{n-1} - 1$, all the bits in $\overline{D}$ is zero except $\overline{d}_{n-1} = 1$. As $c_{-1} = 1$ and $r^{(n-1)}_0 = 1$,c_0 is equal to 1. $c_0 = 1$ and all bits upto n-1 equal to 1 in $R^{(n-1)}$ leads to $c^{(n)}_{n-3} = 1$ (See figure 1). Therefore $(\overline{d}_{n-2}, r^{(n-1)}_{n-2}, c^{(n)}_{n-3}) = (0, 1, 1)$ is a possible combination and a care assignment.

The combination of $(\overline{d}_{n-2}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)}) = (0,1,1)$ also results in $c_{n-2}^{(n)} = 1$. Since $r_{n-1}^{(n-1)} = 0$(From the above discussion) and $c_{n-2}^{(n)} = 1$, $(r_{n-1}^{(n-1)}, c_{n-2}^{(n)}) = (0,1)$ is possible combination and a care assignment.$(r_{n-1}^{(n-1)}, c_{n-2}^{(n)}) = (0,1)$ extends to $(t_{n-1}^{(n)}, r_{n-1}^{(n-1)}) = (0,0)$ (See figure 1).

Proof 5.5: To prove that $(\overline{d}_{n-2}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)})$ = (1,0,0) is a care assignments.

$R^{(n-1)} = 0$, $D = 1$ can give rise to the combinations mentioned in 5.5. This combination holds the constraint $1 \leq D < 2^{n-1}$ and $0 \leq R^{n-1} < 2.D$ (Since $0 < 2$). For $R^{n-1} = 0$, all the bits in $R^{(n-1)}$ are zero. Similarly for $D = 1$, all the bits in $\overline{D}$ are 1 except $\overline{d}_0 = 0$.As $r_0^{(n-1)} = \overline{d}_0 = 0$ and $c_{-1} = 1$, c_0 is equal to 0. $c_0 = 0$ and all bits in $R^{(n-1)} = 0$ results in $c_{n-3}^{(n)} = 0$. Hence $(\overline{d}_{n-2}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)}) = (1,0,0)$ is a possible combination and a care assignment.

Proof 5.6: To prove that $(\overline{d}_{n-2}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)})$ = (1,0,1) is a care assignments.

$R^{(n-1)} = D = 1$ can give rise to the combinations mentioned in 5.6. This combination holds the constraint $1 \leq D < 2^{n-1}$ and $0 \leq R^{n-1} < 2.D$ (Since$R^{(n-1)} = D$). For $R^{n-1} = 1$, all the bits in $R^{(n-1)}$ are zero except $r_0^{(n-1)} = 1$. Similarly for $D = 1$, all the bits in $\overline{D}$ are 1 except $\overline{d}_0 = 0$.As $r_0^{(n-1)}, \overline{d}_0 = (1,0)$ and $c_{-1} = 1$, c_0 is equal to 1. $c_0 = 1$ and all bits from 1 : n-1 are equal to 1 in $\overline{D}$ results in $c_{n-3}^{(n)} = 1$. Hence $(\overline{d}_{n-2}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)}) = (1,0,1)$ is a possible combination and a care assignment.

Proof 5.7: To prove that $(\overline{d}_{n-2}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)})$ = (1,1,0) is a care assignments.

$R^{(n-1)} = 2^{n-2}$, $D = 2^{n-2} - 1$ can give rise to the combinations mentioned in 5.7. This combination holds the constraint $1 \leq D < 2^{n-1}$ and $0 \leq R^{n-1} < 2.D$ (Since $2^{n-2} < 2^{n-1} - 2$). For $R^{n-1} = 2^{n-2}$, all the bits in $R^{(n-1)}$ are zero except $r_{n-2}^{(n-1)} = 1$. Similarly for $D = 2^{n-2} - 1$, all the bits in $\overline{D}$ are 0 except $\overline{d}_{n-1} = \overline{d}_{n-2} = 1$. As $r_0^{(n-1)} = \overline{d}_0 = 0$ and $c_{-1} = 1$, c_0 is equal to $0.c_0 = 0$ and all bits i = 0: n-3 are zero in $R^{(n-1)}$ and $\overline{D}$ results in $c_{n-3}^{(n)} = 0$. Hence $(\overline{d}_{n-2}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)}) = (1,1,0)$ is a possible combination and a care assignment.

2.6 Proof for the exponential size of polynomial representing Restoring Divider

In the current proof, $r_{n-1}^{(n-1)}$ and $r_{n-2}^{(n-1)}$ are set to zero. Since $\overline{d}_{n-1} = 1$ and $r_{n-1}^{(n-1)} = 0$, $t_{n-1}^{(n)} = \overline{c_{n-2}^{(n)}}$. There are two don't care assignments :$(r_{n-1}^{(n-1)}, c_{n-2}^{(n)}) = (1,1)$,$(\overline{d}_{n-2}, r_{n-2}^{(n-1)}, c_{n-3}^{(n)}) = (1,1,1)$ at FA_{n-1} and FA_{n-2} respectively. Using these don't-care assignments would not be effective at this stage as they would be mapped zero because $r_{n-1}^{(n-1)}$ and $r_{n-2}^{(n-1)}$ is set to zero. The same argument holds for the splits of the don't care cubes since they will also have the terms $r_{n-1}^{(n-1)}$ and $r_{n-2}^{(n-1)}$ that will reduce to zero.

The proof starts with the polynomial obtained after rewriting the MUXs equations:

$$\sum_{i=1}^{n-1} q_i 2^{(i)}.D + [q_0.D + q_0.\sum_{i=0}^{n-2} t_i^{(n)} 2^{(i)} - 2^{(n-1)} t_{n-1}^{(n)} - q_0.R^{(n-1)}] + R^{(n-1)} - R^{(0)} \tag{13}$$

where $T^{(n)} = \sum_{i=0}^{n-2} t_i^{(n)} 2^{(i)} - 2^{(n-1)} t_{n-1}^{(n)}$.For now the backward rewriting is only performed on $T^{(n)}$ i.e. q_0 is not replaced when $T^{(n)}$ is being replaced. Substituting $t_{n-1}^{(n)} = 1 - c_{n-2}^{(n)}$ in the expression for $T^{(n)}$ results to $\sum_{i=0}^{n-2} t_i^{(n)} 2^{(i)} + 2^{(n-1)} c_{n-2}^{(n)} - 2^{(n-1)}$.

$c_{n-2}^{(n)}$ can be written as :

$$\begin{aligned} c_{n-2}^{(n)} = [&r_{n-2}^{(n-1)} - r_{n-2}^{(n-1)} d_{n-2} + c_{n-3}^{(n)} \\ &- c_{n-3}^{(n)}.r_{n-2}^{(n-1)} - c_{n-3}^{(n)} d_{n-2} + 2c_{n-3}^{(n)} r_{n-2}^{(n-1)} d_{n-2}] \end{aligned} \tag{14}$$

As per our assumption $r_{n-2}^{(n-1)} = 0$, therefore the above expression for $c_{n-2}^{(n)}$ simplifies to:

$$c_{n-2}^{(n)} = c_{n-3}^{(n-1)} - c_{n-3}^{(n)} d_{n-2} \tag{15}$$

The term $t^{(n)}_{n-2} = (1 - d_{n-1}) \oplus r^{(n-1)}_{n-1} \oplus c^{(n)}_{n-2}$ in $T^{(n)}$ can be expressed as :

$$\begin{aligned} t^{(n)}_{n-2} = [&1 - d_{n-2} - r^{(n-1)}_{n-2} - c^{(n)}_{n-3} + 2r^{(n-1)}_{n-2}c^{(n)}_{n-3} \\ &- 2d_{n-2}(2r^{(n-1)}_{n-2}c^{(n)}_{n-3} - r^{(n-1)}_{n-2} - c^{(n)}_{n-3})] \end{aligned} \tag{16}$$

The above equation can further be simplified using the assumption $r^{(n-1)}_{n-2} = 0$.This will result to :

$$t^{(n)}_{n-2} = [1 - d_{n-2} - c^{(n)}_{n-3} + 2d_{n-2}c^{(n)}_{n-3})] \tag{17}$$

The expression : $2^{(n-2)}t^{(n)}_{n-2} + 2^{(n-1)}c^{(n)}_{n-2} - 2^{(n-1)}$ can be simplified further using equations (15), (17), and results in $2^{(n-2)}c^{(n)}_{n-3} - 2^{(n-2)}d_{n-2} - 2^{(n-2)}$. Hence $T^{(n)}$ simplifies to $\sum_{i=0}^{n-3} t^{(n)}_i 2^{(i)} + 2^{(n-2)}c^{(n)}_{n-3} - 2^{(n-2)}d_{n-2} - 2^{(n-2)}$. Now $2^{n-3}[t^{(n)}_{n-3} + 2c^{(n)}_{n-3}\ ]$ in $T^{(n)}$ can be replaced by the full adder equation : $2^{n-3}[1 - d_{n-3} + r^{(n-1)}_{n-3} + c^{(n)}_{n-4}]$. Therefore $T^{(n)}$ simplifies to

$$T^{(n)} = \sum_{i=0}^{n-4} t^{(n)}_i 2^{(i)} + 2^{(n-3)}c^{(n)}_{n-4} + 2^{(n-3)}r^{(n-1)}_{n-3} - \sum_{i=n-3}^{n-2} d_i 2^{(i)} - 2^{(n-3)} \tag{18}$$

After replacing the remaining full adders, the polynomial for $T^{(n)}$ simplifies to
:

$$T^{(n)} = t^{(n)}_0 + 2c^{(n)}_0 + \sum_{i=1}^{n-3} r^{(n-1)}_i 2^{(i)} - \sum_{i=1}^{n-2} d_i 2^{(i)} - 2. \tag{19}$$

Using equation (19) in equation (13) :

$$\sum_{i=1}^{n-1} q_i 2^{(i)}.D + q_0.[D + t_0^{(n)} + 2c_0^{(n)} + \sum_{i=1}^{n-3} r_i^{(n-1)} 2^{(i)} - \sum_{i=1}^{n-2} d_i 2^{(i)} - 2 - R^{(n-1)}]$$
$$+ R^{(n-1)} - R^{(0)} \tag{20}$$

The above equation can be further simplified by using $D = \sum_{i=0}^{n-2} d_i 2^{(i)}$ and $R^{(n-1)} = \sum_{i=0}^{n-3} r_i^{(n-1)} 2^{(i)}$ ($\because r_{n-1}^{(n-1)} = r_{n-2}^{(n-1)} = 0$).

$$\sum_{i=1}^{n-1} q_i 2^{(i)}.D + q_0.[t_0^{(n)} + 2c_0^{(n)} + d_0 - r_0^{(n-1)} - 2] + R^{(n-1)} - R^{(0)} \tag{21}$$

$[t_0^{(n)} + 2c_0^{(n)} + d_0 - r_0^{(n-1)} - 2]$ in the above equation will reduce to zero if $t_0 + 2c_0$ is rewritten by using the full adder equation: $1 - d_0 + r_0^{(n-1)} + 1$. however, if q_0 in equation is expressed in terms of $< d_{n-1} \dots d_0 >, < r_{n-1}^{(n-1)} \dots r_0 >, c_0^{(n)}$, then its is possible to prove that the resulting polynomial for Eqn. (21) is exponential. With $r_{n-1}^{(n-1)} = 0$ and $\overline{d}_{n-1} = 1$, $q0 = c_{n-2}^{(n)}$. Thus q_0 is a carry bit of an adder and an exponential representation for q_0 can be derived with arguments similar to those in the proofs of lemma 1 and 2 of [1].

3 Delay rewriting of the last stage reduced restoring divider

The current section illustrates a method to arrive at the required polynomial at the end of backward rewriting of the last stage reduced restoring divider. The following analysis is more in line with the implementation and also simple to prove that delaying the rewriting of q_0 results in the intended polynomial.

The specification polynomial for a reduced restoring divider is given as :

$$\sum_{i=1}^{n-1} q_i 2^{(i)}.D + q_0.D + \sum_{i=0}^{n-2} r_i^{(n)} 2^{(i)} - 2^{(n-1)} r_{n-1}^{(n)} - R^{(0)} \tag{22}$$

Where $R^{(n)} = \sum_{i=0}^{n-2} r_i^{(n)} 2^{(i)} - 2^{(n-1)} r_{n-1}^{(n)}$ and is the final reminder after multiplexer of the last stage ($MUX^{(n)}$).The polynomial between $MUX^{(n)}$ and $SUB^{(n)}$ is obtained by replacing every bit of $R^{(n)}$ by Mux equation : $q_0.t_i^{(n)} + (1 - q_0) r_i^{(n-1)}$. In the following analysis, the term $R^{(0)}$ is ignored because it is not considered at this stage for backward rewriting.

$$\sum_{i=1}^{n-1} q_i 2^{(i)}.D + q_0.D + q_0 (\sum_{i=0}^{n-2} t_i^{(n)} 2^{(i)} - 2^{(n-1)} t_{n-1}^{(n)}) + (1 - q_0)(\sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} - 2^{(n-1)} r_{n-1}^{(n-1)}) \tag{23}$$

The above equation is simplified by grouping terms that are associated with q_0:

$$\begin{aligned} &\sum_{i=1}^{n-1} q_i 2^{(i)}.D + (\sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} - 2^{(n-1)} r_{n-1}^{(n-1)}) + q_0 (\sum_{i=0}^{n-2} t_i^{(n)} 2^{(i)} - 2^{(n-1)} t_{n-1}^{(n)} \\ &- \sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} + 2^{(n-1)} r_{n-1}^{(n-1)} + D) \end{aligned} \tag{24}$$

In the above equation, $\sum_{i=0}^{n-2} t_i^{(n)} 2^{(i)} - 2^{(n-1)} t_{n-1}^{(n)}$ can be replaced as $R^{(n-1)} - D$. Firstly, $t_{n-1}^{(n)} = \overline{d}_{n-1} \oplus r_{n-1}^{(n-1)} \oplus c_{n-2}^{(n)}$, as $d_{n-1} = 0$, the backward rewriting replaces $t_{n-1}^{(n)}$ as $1 - r_{n-1}^{(n-1)} - c_{n-2}^{(n)} + 2 r_{n-1}^{(n-1)} c_{n-2}^{(n)}$.

$$\begin{aligned} &\sum_{i=1}^{n-1} q_i 2^{(i)}.D + (\sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} - 2^{(n-1)} r_{n-1}^{(n-1)}) + q_0 (\sum_{i=0}^{n-2} t_i^{(n)} 2^{(i)} + 2^{(n-1)} c_{n-2}^{(n)} - 2^{(n)} r_{n-1}^{(n-1)} c_{n-2}^{(n)} \\ &- 2^{(n-1)} - \sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} + 2^{(n)} r_{n-1}^{(n-1)} + D) \end{aligned} \tag{25}$$

At this stage of analysis, there is a don't care assignment $(r_{n-1}^{(n-1)} c_{n-2}^{(n)})$ that can be used to optimise the above equation. However, in the above equation, the monomial $(r_{n-1}^{(n-1)} c_{n-2}^{(n)})$ is also associated with the term q_0; therefore, to optimise the above equation accordingly, don't care splitting is used.

The available don't care assignment can be rewritten as $v1 * (q_0 r_{n-1}^{(n-1)} c_{n-2}^{(n)}) + v2 * (1 - q_0)(r_{n-1}^{(n-1)} c_{n-2}^{(n)})$, which simplifies further as $(v1 - v2) * (q_0 r_{n-1}^{(n-1)} c_{n-2}^{(n)}) + v2 * (r_{n-1}^{(n-1)} c_{n-2}^{(n)})$ and added to equation (25).

$$\begin{aligned}&\sum_{i=1}^{n-1} q_i 2^{(i)}.D + (\sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} - 2^{(n-1)} r_{n-1}^{(n-1)}) + q_0(\sum_{i=0}^{n-2} t_i^{(n)} 2^{(i)} + 2^{(n-1)} c_{n-2}^{(n)} + (v1 - v2 - 2^{(n)}) \\ &* (r_{n-1}^{(n-1)} c_{n-2}^{(n)}) + v2 * (r_{n-1}^{(n-1)} c_{n-2}^{(n)}) - 2^{(n-1)} - \sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} + 2^{(n)} r_{n-1}^{(n-1)} + D) \qquad (26)\end{aligned}$$

$v2 = 0$ and $v1 = 2^n$ is the optimal solution to optimise the above equation leading to:

$$\begin{aligned}&\sum_{i=1}^{n-1} q_i 2^{(i)}.D + (\sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} - 2^{(n-1)} r_{n-1}^{(n-1)}) + q_0(\sum_{i=0}^{n-2} t_i^{(n)} 2^{(i)} + 2^{(n-1)} c_{n-2}^{(n)} - 2^{(n-1)} \\ &- \sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} + 2^{(n)} r_{n-1}^{(n-1)} + D) \qquad (27)\end{aligned}$$

In the above equation, $2^{(n-2)} t_{n-2}^{(n)} + 2^{(n-1)} c_{n-2}^{(n)}$ can be replaced by full adder equation :$2^{(n-2)}[1 - d_{n-2} + r_{n-2}^{(n-1)} + c_{n-3}^{(n)}]$

$$\begin{aligned}&\sum_{i=1}^{n-1} q_i 2^{(i)}.D + (\sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} - 2^{(n-1)} r_{n-1}^{(n-1)}) + q_0(\sum_{i=0}^{n-3} t_i^{(n)} 2^{(i)} + 2^{(n-2)} c_{n-3}^{(n)} + 2^{(n-2)} r_{n-2}^{(n-1)} \\ &- 2^{(n-2)} d_{n-2} - 2^{(n-2)} - \sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} + 2^{(n)} r_{n-1}^{(n-1)} + D) \qquad (28)\end{aligned}$$

Replacing the remaining full adders with their corresponding equations results in :

$$\begin{aligned}&\sum_{i=1}^{n-1} q_i 2^{(i)}.D + (\sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} - 2^{(n-1)} r_{n-1}^{(n-1)}) + q_0(\sum_{i=0}^{n-2} r_i^{(n-1)} - \sum_{i=0}^{n-2} d_i - \sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} \\ &+ 2^{(n)} r_{n-1}^{(n-1)} + D) \qquad (29)\end{aligned}$$

In the above equation, $D - \sum_{i=0}^{n-2} d_i = 0$ and $\sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} - \sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} = 0$. Thus, the equation simplifies to

$$\sum_{i=1}^{n-1} q_i 2^{(i)}.D + (\sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} - 2^{(n-1)} r_{n-1}^{(n-1)}) + q_0(2^{(n)} r_{n-1}^{(n-1)}) \tag{30}$$

Replace $q_0 = 1 - t_{n-1}^{(n)}$, where $t_{n-1}^{(n)} = 1 - r_{n-1}^{(n-1)} - c_{n-2}^{(n)} + 2r_{n-1}^{(n-1)} c_{n-2}^{(n)}$ and therefore $q_0 = r_{n-1}^{(n-1)} + c_{n-2}^{(n)} - 2r_{n-1}^{(n-1)} c_{n-2}^{(n)}$.

$$\sum_{i=1}^{n-1} q_i 2^{(i)}.D + (\sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} - 2^{(n-1)} r_{n-1}^{(n-1)}) + (r_{n-1}^{(n-1)} + c_{n-2}^{(n)} - 2r_{n-1}^{(n-1)} c_{n-2}^{(n)})(2^{(n)} r_{n-1}^{(n-1)}) \tag{31}$$

The above equation further simplifies to

$$\sum_{i=1}^{n-1} q_i 2^{(i)}.D + (\sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} - 2^{(n-1)} r_{n-1}^{(n-1)}) + 2^{(n)} r_{n-1}^{(n-1)} - 2^{(n)} r_{n-1}^{(n-1)} c_{n-2}^{(n)} \tag{32}$$

As already mentioned previously $(r_{n-1}^{(n-1)} c_{n-2}^{(n)})$ is a don't care assignment that can be used to optimise the above equation, which leads to

$$\sum_{i=1}^{n-1} q_i 2^{(i)}.D + (\sum_{i=0}^{n-2} r_i^{(n-1)} 2^{(i)} + 2^{(n-1)} r_{n-1}^{(n-1)}) \tag{33}$$

From the above analysis, we can infer that delaying the signal that occurs more often and replacing it when it no longer reduces helps to maintain quadratic polynomial size and, in the best case, can also give rise to the intended polynomial. However, delaying q_0 works accurately in the $stage_n$ of a restoring divider due to the presence of the don't care $(r_{n-1}^{(n-1)} c_{n-2}^{(n)}) = (1, 1)$, which helps to replace $\sum_{i=0}^{n-2} t_i^{(n)} 2^{(i)} - 2^{(n-1)} t_{n-1}^{(n)}$ as $R^{(n-1)} - D$. Hence, yielding the correct polynomial between $Stage_n^{mux}$ and $stage_n^{sub}$. This might not always be the case, as seen in the next section.

3.1 Analysis of $n-1^{th}$ stage of reduced restoring divider

The analysis of $stage_{n-1}$ continues with the polynomial seen in equation (33).The polynomial between $stage^{mux}_{(n-1)}$ and $stage^{sub}_{(n-1)}$ is obtained by replacing every bit of $R^{(n-1)}$ by Mux equation : $q_1.t_i^{(n-1)} + (1-q_1)r_i^{(n-2)}$. The term $R^{(0)}$ in the following illustration is ignored as it is not considered for backward rewriting.

$$\sum_{i=2}^{n-1} q_i 2^{(i)}.D + 2q_1.D + q_1 \sum_{i=0}^{n-2} t_i^{(n-1)} 2^{(i+1)} + (1-q_1)(\sum_{i=1}^{n-1} r_i^{(n-2)} 2^{(i)}) + r_0^{(n-2)} \tag{34}$$

The above equation is simplified by grouping terms that are associated with q_1:

$$\sum_{i=2}^{n-1} q_i 2^{(i)}.D + q_1(2.D + \sum_{i=0}^{n-2} t_i^{(n-1)} 2^{(i+1)} - \sum_{i=1}^{n-1} r_i^{(n-2)} 2^{(i)}) + \sum_{i=0}^{n-1} r_i^{(n-2)} 2^{(i)} \tag{35}$$

Similar to the previous section, the replacement of signal q_1 is delayed. In the above equation, $t_{n-2}^{(n-1)} = \overline{d}_{n-2} \oplus r_{n-1}^{(n-2)} \oplus c_{n-3}^{(n-1)}$ and is equivalent to

$$\begin{aligned} t_{n-2}^{(n-1)} &= \overline{d}_{n-2} + r_{n-1}^{(n-2)} + c_{n-3}^{(n-1)} - 2r_{n-1}^{(n-2)} c_{n-3}^{(n-1)} - 2\overline{d}_{n-2}.r_{n-1}^{(n-2)} \\ &\quad - 2\overline{d}_{n-2} c_{n-3}^{(n-1)} + 4\overline{d}_{n-2} r_{n-1}^{(n-2)} c_{n-3}^{(n-1)} \end{aligned} \tag{36}$$

Substituting equation (36) in (35):

$$\begin{aligned} &\sum_{i=2}^{n-1} q_i 2^{(i)}.D + q_1(2.D + \sum_{i=0}^{n-3} t_i^{(n-1)} 2^{(i+1)} + 2^{n-1}(\overline{d}_{n-2} + r_{n-1}^{(n-2)} + c_{n-3}^{(n-1)} - 2r_{n-1}^{(n-2)} c_{n-3}^{(n-1)} \\ &- 2\overline{d}_{n-2}.r_{n-1}^{(n-2)} - 2\overline{d}_{n-2} c_{n-3}^{(n-1)} + 4\overline{d}_{n-2} r_{n-1}^{(n-2)} c_{n-3}^{(n-1)}) - \sum_{i=1}^{n-1} r_i^{(n-2)} 2^{(i)}) + \sum_{i=0}^{n-1} r_i^{(n-2)} 2^{(i)} \end{aligned} \tag{37}$$

The above equation can be optimised using the don't care splitting of the don't care assignment $(\overline{d}_{n-2}, r_{n-1}^{(n-2)}, c_{n-3}^{(n-1)}) = (1,1,1)$ in the similar way as shown in the previous section, and the above equation simplifies to

$$\begin{aligned} &\sum_{i=2}^{n-1} q_i 2^{(i)}.D + q_1(2.D + \sum_{i=0}^{n-3} t_i^{(n-1)} 2^{(i+1)} + 2^{n-1}(\overline{d}_{n-2} + r_{n-1}^{(n-2)} + c_{n-3}^{(n-1)} - 2r_{n-1}^{(n-2)} c_{n-3}^{(n-1)} \\ &- 2\overline{d}_{n-2}.r_{n-1}^{(n-2)} - 2\overline{d}_{n-2} c_{n-3}^{(n-1)}) - \sum_{i=1}^{n-1} r_i^{(n-2)} 2^{(i)}) + \sum_{i=0}^{n-1} r_i^{(n-2)} 2^{(i)} \end{aligned} \tag{38}$$

Although in the above equation, $2^{(n-2)}t_{n-3}^{(n-1)} + 2^{(n-1)}c_{n-3}^{(n-1)}$ can be replaced by full adder equation:$2^{(n-2)}[\overline{d}_{n-3} + r_{n-2}^{(n-2)} + c_{n-4}^{(n-1)}]$, c_{n-3}^{n-1} is also associated with other signals such as $\overline{d}_{n-2}$, $r_{n-1}^{(n-2)}$, and replacing c_{n-3} could increase the polynomial size significantly. The expression obtained after replacing c_{n-3}^{n-1} is denoted as k.

$$\sum_{i=2}^{n-1} q_i 2^{(i)}.D + q_1(2.D + \sum_{i=0}^{n-4} t_i^{(n-1)} 2^{(i+1)} + 2^{(n-2)} c_{n-4}^{(n-1)} + \sum_{i=n-3}^{n-2} \overline{d}_i^{(n-1)} 2^{(i+1)} + 2^{n-1}(k)$$
$$- \sum_{i=1}^{n-3} r_i^{(n-2)} 2^{(i)}) + \sum_{i=0}^{n-1} r_i^{(n-2)} 2^{(i)} \tag{39}$$

Similarly, $2^{(n-3)}t_{n-4}^{(n-1)} + 2^{(n-2)}c_{n-4}^{(n-1)}$ can be replaced by full adder equation:$2^{(n-3)}[\overline{d}_{n-4} + r_{n-3}^{(n-2)} + c_{n-5}^{(n-1)}]$ and by replacing the remaining full adders with their equations, we arrive at the following equation:

$$\sum_{i=2}^{n-1} q_i 2^{(i)}.D + q_1(2.D + \sum_{i=0}^{n-2} \overline{d}_i 2^{(i+1)} + 2c_{-1}^{(n-1)} + 2^{n-1}(k)) + \sum_{i=0}^{n-1} r_i^{(n-2)} 2^{(i)} \tag{40}$$

The above equation can further be simplified since $\overline{d}_i = 1 - d_i$

$$\sum_{i=2}^{n-1} q_i 2^{(i)}.D + q_1(2.D - 2\sum_{i=0}^{n-2} d_i + \sum_{i=0}^{n-2} 2^{(i+1)} + 2c_{-1}^{(n-1)} + 2^{n-1}(k)) + \sum_{i=0}^{n-1} r_i^{(n-2)} 2^{(i)} \tag{41}$$

simplifying $2.D - 2\sum_{i=0}^{n-2} d_i = 0, \sum_{i=0}^{n-2} 2^{(i+1)} + 2c_{-1}^{(n-1)} = 2^n (\text{as } c_{-1}^{n-1} = 1)$ in the above expressions, gives rise to:

$$\sum_{i=2}^{n-1} q_i 2^{(i)}.D + q_1(2^n + 2^{n-1}(k)) + \sum_{i=0}^{n-1} r_i^{(n-2)} 2^{(i)} \tag{42}$$

Signal q_1 is replaced as $1 - t_n^{(n-1)}$, which is equivalent to $q_1 = r_n^{(n-2)} + c_{n-2}^{(n-1)} - 2r_n^{(n-2)}c_{n-2}^{(n-1)}$. Substituting the expression for signal q_1 in the above equation results in

$$\sum_{i=2}^{n-1} q_i 2^{(i)}.D + (r_n^{(n-2)} + c_{n-2}^{(n-1)} - 2r_n^{(n-2)} c_{n-2}^{(n-1)})(2^n + 2^{n-1}(k)) + \sum_{i=0}^{n-1} r_i^{(n-2)} 2^{(i)} \tag{43}$$

It is still possible to remove all the terms associated with $(r_n^{(n-2)} c_{n-2}^{(n-1)})$ as $(r_n^{(n-2)}, c_{n-2}^{(n-1)}) = (1,1)$ is a don't care. However, the size of k is significantly large and is function of $< r_{n-1}^{n-2} \dots r_1^{n-2} >$ and $< d_{n-2} \dots d_0 >$. Additionally, we still have the signal c_{n-2}^{n-1} that has to be replaced, which can increase the polynomial size significantly. Though delaying the replacement of high occurrence signals will give better results than replacing them in topological order, without the correct don't care assignment, delaying signals will not avoid exponential blow-up of the polynomial size. In $stage_n$, though we do not have a real subtractor stage, the presence of don't care assignment $(r_n^{(n-2)}, c_{n-2}^{(n-1)}) = (1,1)$ and constant propagation of $d_{n-1} = 0$ enable the backward rewriting to replace $\sum_{i=0}^{n-2} t_i^{(n)} 2^{(i)} - 2^{(n-1)} t_{n-1}^{(n)}$ as $R^{(n-1)} - D$. However, in $stage_{n-1}$, we start the backward rewriting with polynomial obtained after rewriting of $stage_n$ (see equation (33)), which prunes the sign bit $r_n^{(n-1)}$ of the partial remainder R^{n-1}. Hence, the incomplete subtractor of $stage_{n-2}$ cannot be replaced as $R^{n-2} - D$ as there is only one don't care assignment $(\overline{d}_{n-2}, r_{n-1}^{(n-2)}, c_{n-3}^{(n-1)}) = (1,1,1)$ available. Due to the residual terms associated with signal $c_{n-3}^{(n-1)}$, there is still a significant increase in the polynomial size. The same discussion holds for all the stages from n-1 to 1. One of the possible solution is to add the sign extensions after rewriting completes each stage. For example, Equation (33) can be altered as $\sum_{i=1}^{n-1} q_i 2^{(i)}.D + (\sum_{i=0}^{n-1} r_i^{(n-1)} 2^{(i)} - 2^{(n)} r_n^{(n-1)})$, we could add the bit $2^{(n)} r_n^{(n-1)}$, as it has to be zero for a correct restoring divider. But, this solution could still be challenging to implement in practice.

Another possible solution at our disposal is to use the don't care monomials to extend a polynomial. In Equation (34), $q_1 \cdot \sum_{i=0}^{n-2} t_i^{(n-1)} 2^{(i+1)}$ can be extended by adding a don't care $v1 \cdot q_1 t_{n-1}^{(n-1)}$. Instead of using ILP to optimize the polynomial, the variable v_1 can be assigned the value of -2^n. This value for v_1 is chosen based on the weight associated with the term $t_{n-2}^{(n-1)} 2^{(n-1)}$. Now the polynomial in Eqution (34) transforms to the following.

$$\sum_{i=2}^{n-1} q_i 2^{(i)}.D + q_1(2.D + \sum_{i=0}^{n-2} t_i^{(n-1)} 2^{(i+1)} - 2^n t_{n-1}^{(n-1)} - \sum_{i=1}^{n-1} r_i^{(n-2)} 2^{(i)}) + \sum_{i=0}^{n-1} r_i^{(n-2)} 2^{(i)} \tag{44}$$

Equation (44) exactly resembles Equation (24). Therefore, similar backward rewriting illustrated in the rewriting of $Stage_n^{sub}$ can also be followed in replacing $Stage_{n-1}^{sub}$ along with delayed rewriting for signal q_1. Doing so will yield $\sum_{i=2}^{n-1} q_i 2^{(i)}.D + \sum_{i=0}^{n-1} r_i^{(n-2)} 2^{(i)}$ as the final polynomial without any exponential blow-up of intermediate polynomial size. Though this is a theoretically viable solution, implementing this solution in a practical automatic verification tool needs a few changes. Firstly, the computation or extracting don't care candidates on signals that are not directly part of an atomic block. In our case $q_1 t_{n-1}^{(n-1)}$ are antivalent signals that can also be regarded as a don't care. The second step involves the identification of incomplete polynomials For example, $(q_1 \cdot \sum_{i=0}^{n-2} t_i^{(n-1)} 2^{(i+1)})$. The next step will also require to extract the weight of the most significant bit of the incomplete monomial, then scale that weight by 2 and negated it. This value will be assigned to the integer variable associated with the don't care monomial. The final step includes the continuation of backward rewriting. Overall this approach requires more analysis on how it has to be implemented such that it generalises to all divider circuits.

3.2 A Short Note on Implementation of Delayed Rewriting

The implementation combines delayed rewriting and backtracking with don't care. A single stack handles the backtracking of don't care and signals with high occurrences.

1. The initial step involves recording the signals' occurrences at regular intervals. In the current implementation, depending on the size of the divider, the signal occurrence is recorded after rewriting certain atomic blocks. For example, for a 4-bit divider, the signal occurrence is recorded periodically after rewriting 3 atomic blocks. Similarly, for an 8-bit divider, the signals are periodically recorded after rewriting seven atomic blocks. However, this frequency of recording the occurrence of the signal can be changed accordingly.

2. From Step 1, we select the signals that appear more frequently according to a certain threshold. In the current implementation, signals with a count greater than 25 percent of the actual polynomial size are selected.

3. Whenever backward rewriting is about to replace the most occurring signals that have been extracted from Step 2, a backward point is set, and the rewriting continues. Meanwhile, along with setting the backtrack point on high-occurrence signals, a backtrack point is also set on the don't care candidates. Both backtrack points are pushed on a single stack. Additional information on what type of backtrack point is also included.

4. When the size of the polynomial increases beyond a threshold, the tool backtracks to the recently stored backtrack point. Based on the type of backtrack point, the tool either skips the signal or enables don't care optimization.

5. When and if a signal is skipped, we keep track of it to replace it later. When the occurrence of the skipped signal no longer reduces, we replace it.

By using a single stack to store the delayed backtrack points and Don't Care backtrack points, the implementation remains simple and also allows for exploring all kinds of combinations to reduce or optimize the polynomial size.

References

[1] C. Scholl, “Some Notes on Arithmetic Verification,” Institute of Computer Science, Albert-Ludwigs-University of Freiburg, Freiburg im Breisgau, Germany 2024.