ECE 465 - Digital System Design
Project1 - Full
Spring 2025

Michael Stanley
Dominik Kornak
Arthur Stanson
Ayse Unlu

(i) We get correct outputs for all the inputs in the given waveform.

(ii) Michael Stanley: Developed conceptual design with Dominik in part 1. Debugged and optimized div7 circuits in Quartus. Ran DV simulations. Compiled report in latex.

(ii) Dominik Kornak: Designed 16 bit div7 circuit in Quartus, schematic in part 1. Developed conceptual design with Michael in part 1.

(ii) Arthur Stanson: Designed 8 bit div7 circuit in Qaurtus. Made FA and RCA's in Quartus. Helped with conceptual design in part 1. Analyzed DV results.

(ii) Ayse Unlu: Helped with conceptual design in part 1. Analyzed timing simulations for the TA waveform. Analyzed DV results.

Design Problem

(a):

Let's begin with a mathematical analysis of the problem. Once we understand the mathematical breakup, we can apply our understanding to a D&C design. First, consider an n bit number, D, where $n = 2^k$ for some k. For what values of k does it make sense to break D up at all? That is, what should the bit width of our leaf node be?

Well, a 2 bit number cannot represent 7, so a breakup for k=1 does not make sense. Next, we have a 4 bit number, with k=2. A 4 bit number is the minimum bit width for some $n = 2^k$ that can represent 7. Clearly, it makes no sense to break a 4 bit number up more, so we have discovered the bit width of our leaf node. We will take an n bit input, with $k \geq 3$, and recursively break it up until we have reached the level at which we have 4 bit sub problems.

Now, consider an n bit number, D, with $k \geq 3$. We can split D in half as such:

$$D = 2^{n/2} D_H + D_L \tag{1}$$

Next, we can solve the following two sub problems to obtain:

$$D_H = 7 \cdot Q_H + R_H$$
$$D_L = 7 \cdot Q_L + R_L$$

Recombining terms with (1), we find:

$$D = 7(2^{n/2} Q_H + Q_L) + (2^{n/2} R_H + R_L) \tag{2}$$

From (2), we can derive equations for the quotient and remainder of D as such:

$$Q = 2^{n/2} Q_H + Q_L + \lfloor \frac{2^{n/2} R_H + R_L}{7} \rfloor \tag{3}$$

$$R = (2^{n/2} R_H + R_L) \ mod \ 7 \tag{4}$$

Equations (3) and (4) represent the solution to our problem from the perspective of a non-leaf node in our D&C break up. From the perspective of the leaf nodes, equations (3) and (4) can be generalized as such:

$$Q = 2^{n-4} Q_j + 2^{n-8} Q_{j-1} + ... + Q_1 + \lfloor \tfrac{2^{n-4} R_j}{7} \rfloor + \lfloor \tfrac{2^{n-8} R_{j-1}}{7} \rfloor + ... + \lfloor \tfrac{R_1}{7} \rfloor +$$
$$+ \lfloor \frac{(2^{n-4} R_j \% 7) + (2^{n-8} R_{j-1} \% 7) + ... + (R_1 \% 7)}{7} \rfloor \tag{5}$$

$$R = ((2^{n-4} R_j) + (2^{n-8} R_{j-1}) + ... + R_1) \ mod \ 7 \tag{6}$$

Where j corresponds to the quotient and remainder of the 4 MSB's, and 1 to the quotient and remainder of the 4 LSB's of the break up. Equation (5) shows us how we will stitch the quotient up. All partial Q terms of the form $2^x Q_y$ can be combined using or's instead of adders. The value of the "floor" terms of the form $\lfloor \frac{2^x R_y}{7} \rfloor$ will be determined at the leaf level, along with the terms of the form $2^x R_y \% 7$. At the top stitch up level, we will add the partial Q terms and floor terms using 16 bit adders with their carry in bit inputs set to the output of the final floor function seen in (5).

We will determine R by computing (6) at the final stitch up level. There is no reason to break (6) up, as the mod operator is not distributive.

Now that we understand the problem, we can perform the breakup. Beginning with the root, we break the problem up as so:

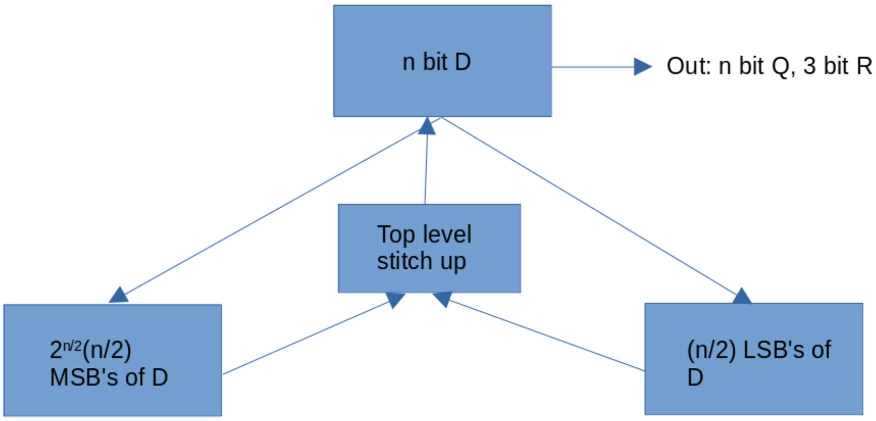


Figure 1: Breakup of root

Note, we output an n bit quotient for simplicity, and a 3 bit remainder as the maximum value the remainder can attain is 6. Once the root problem is broken up, the left and right hand sub problems are broken up recursively in the same way as seen in the figre above until each sub problem has a bit width of 4, and we have reached the leaf level.

(b): The top level stitch up function does the majority of the work in our design. Here, we calculate the remainder, and compute the final quotient sum. The output size of Q is n bits, and the output size of R is 3 bits.

To begin with, the top level stitch up receives the sum of partial Q's:

$$Q_p = 2^{n-4}Q_j + 2^{n-8}Q_{j-1} + ... + Q_1 \tag{7}$$

The additional Q terms output by the leaf nodes:

$$\lfloor \tfrac{2^{n-4}R_j}{7} \rfloor, \lfloor \tfrac{2^{n-8}R_{j-1}}{7} \rfloor, ..., \lfloor \tfrac{R_1}{7} \rfloor$$

Along with the terms of the following form output by the leaf nodes:

$$(2^{n-4}R_j\%7), (2^{n-8}R_{j-1}\%7), ..., (R_1\%7)$$

First, these terms are summed together using a cascaded sequence of RCA's (note that each term is 3 bits):

$$(2^{n-4}R_j\%7) + (2^{n-8}R_{j-1}\%7) + ... + (R_1\%7)$$

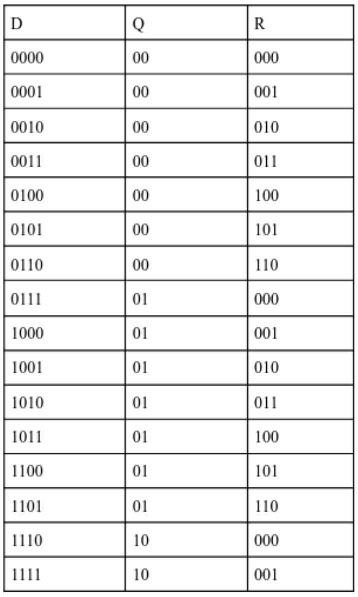
Next, we compute the following function using combinatorial logic:

$$Q_r = \lfloor \frac{(2^{n-4}R_j\%7) + (2^{n-8}R_{j-1}\%7) + ... + (R_1\%7)}{7} \rfloor \tag{8}$$

The function can be easily computed combinatorialially by noting that if the sum is greater than 6, Q_r must be 1 or greater. If the sum is greater than 13, Q_r must be 2 or greater, etc. Since we know the number of values Q_r can take on for an n bit number, the function can be derived and implemented for a particular n.

Next, we take Q_p and input it and one of the additional Q's into a 16 bit RCA. The output of this RCA is input into another 16 bit RCA to be summed with another additional Q. The process is cascaded, until Q_p has been summed together with all the additional Q's. The carry in bits of the RCA's are controlled by the value of Q_r. If Q_r=1, 1 carry in bit is high, if Q_r=2, 2 carry in bits are high etc.

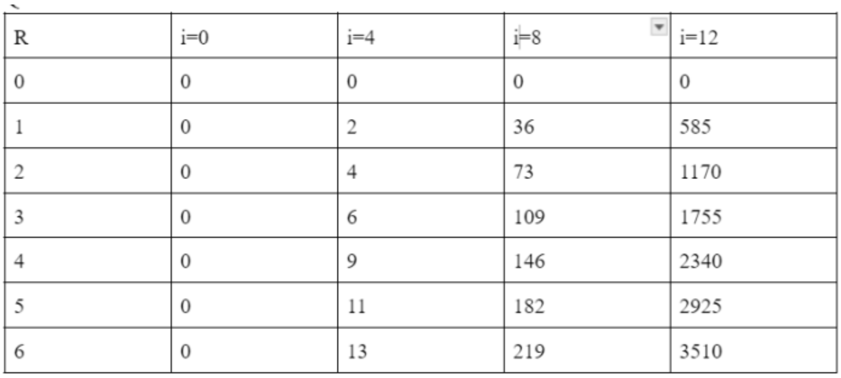
(c): Our leaf function if of 4 bits. It takes in a 4 bit number, and then applies it to a combinatorial circuit implementing the following truth table:



D	Q	R
0000	00	000
0001	00	001
0010	00	010
0011	00	011
0100	00	100
0101	00	101
0110	00	110
0111	01	000
1000	01	001
1001	01	010
1010	01	011
1011	01	100
1100	01	101
1101	01	110
1110	10	000
1111	10	001

Figure 2: 4 bit input leaf function

The Q output of the TT is << 4 if it is the left child node of its parent. Next, the remainder is used to index a hardcoded lookup table to produce the additional Q:



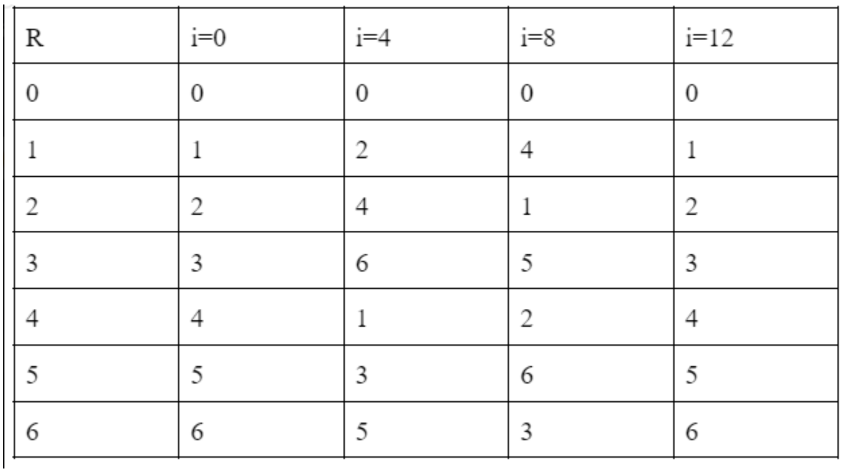
R	i=0	i=4	i=8	i=12
0	0	0	0	0
1	0	2	36	585
2	0	4	73	1170
3	0	6	109	1755
4	0	9	146	2340
5	0	11	182	2925
6	0	13	219	3510

Figure 3: Example of Qa table for some i

Where each leaf node has a 7 entry lookup table associated with it, as can be seen above. The entries are computed using the formula:

$$Q_a = \lfloor \frac{2^i R}{7} \rfloor \tag{9}$$

Finally, the terms of the form $2^iR\%7$ are indexed in another hardcoded lookup table using the output remainder. Some examples of the tables for particular i are seen below:

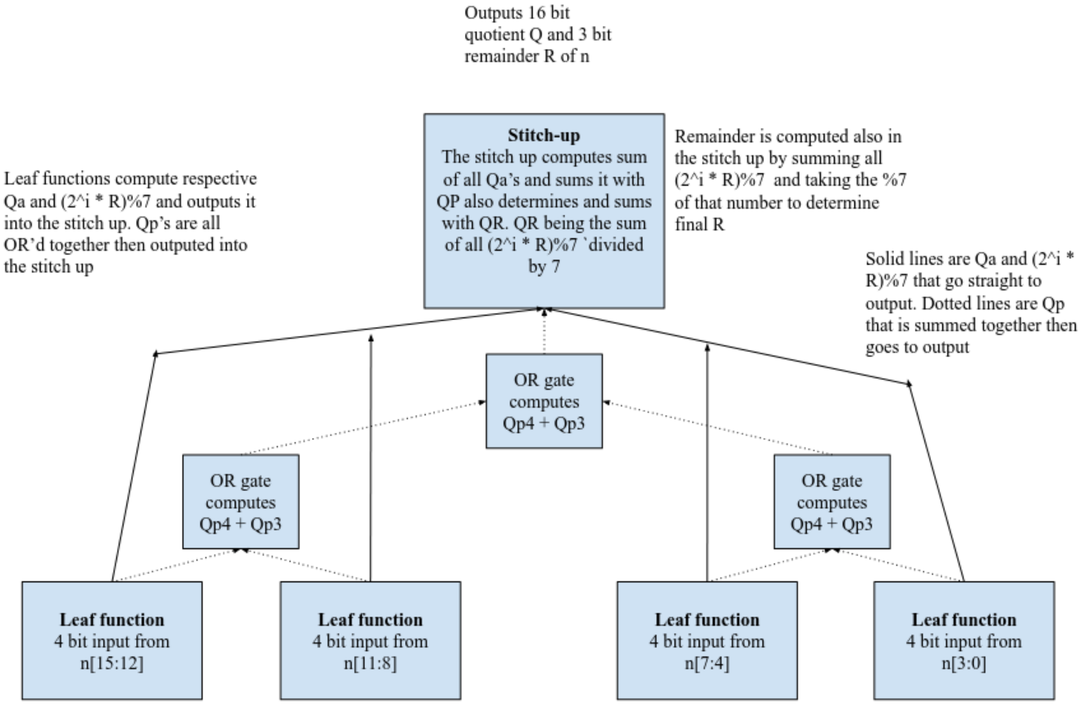


R	i=0	i=4	i=8	i=12
0	0	0	0	0
1	1	2	4	1
2	2	4	1	2
3	3	6	5	3
4	4	1	2	4
5	5	3	6	5
6	6	5	3	6

Figure 4: Final lookup table examples for some i

The outputs of both lookup tables in each leaf node are sent to the top stitch up function to compute the final values as described in part (b).

(d):


Stitch-up
The stitch up computes sum of all Qa's and sums it with QP also determines and sums with QR. QR being the sum of all (2^i * R)%7 `divided by 7
OR gate computes Qp4 + Qp3
OR gate computes Qp4 + Qp3
OR gate computes Qp4 + Qp3
Leaf function 4 bit input from n[15:12]
Leaf function 4 bit input from n[11:8]
Leaf function 4 bit input from n[7:4]
Leaf function 4 bit input from n[3:0]

Outputs 16 bit quotient Q and 3 bit remainder R of n

Leaf functions compute respective Qa and (2^i * R)%7 and outputs it into the stitch up. Qp's are all OR'd together then outputed into the stitch up

Remainder is computed also in the stitch up by summing all (2^i * R)%7 and taking the %7 of that number to determine final R

Solid lines are Qa and (2^i * R)%7 that go straight to output. Dotted lines are Qp that is summed together then goes to output

Figure 5: 16 bit break up

(e): We will take Full adders to have a delay of 1 unit. The longest delay path in our circuit is the path from the output of a leaf node TT, to the first FA in the widest RCA for the Q_r calculation, through that RCA, and then input as one of the carry in bits of the RCA for the final Q calculation.

To determine what the widest RCA for the Q_r calculation is, we note that each term of the form $2^iR\%7$ is 3 bits wide. Noting that we will have $n/4$ of these terms, we must have $n/4$ RCA's in this stage. Then, we assume the worst case, where each $2^iR\%7$ term is 6. The sum will then equal 6*$n/4$. So, we will need a minimum of $\lceil log_2(6 * {}^n/_4) \rceil$ bits in the final RCA.

Adding the delay of the adder, we get a delay function for the top level stitch up of:

$$\delta = \lceil log_2(6 * {}^{n}\!/_{4})\rceil + n \tag{10}$$

Assuming that the combinatorial circuit producing the output of the leaf TT has a delay of 1 unit, the total delay of our circuit is:

$$\delta = 1 + \lceil log_2(6 * {}^{n}\!/_{4})\rceil + n \tag{11}$$

(f):

Assuming that FAs contribute 1 unit cost, the mod 7 circuit contributes 1 unit cost, and the 4 bit TT in the leaf node contributes 1 unit cost, we derived the following. To find the cost in the top level stitch up, we add the cost of calculating Q_r and summing Q. The cost to calculate Q_r is ${}^{n}\!/_{4}$*$\lceil log_2(6 * {}^{n}\!/_{4})\rceil$. The cost of the hardware that calculates Q is (${}^{n}\!/_{4}$-1)*n. Adding all of these term together, we get a total cost at the top level stitch up of:

$$\delta = 1 + (\frac{n}{4} - 1) * n + \frac{n}{4}\lceil log_2(6 * {}^{n}\!/_{4})\rceil \tag{12}$$

Adding in the cost of the leaf nodes, we end with a total cost of:

$$\delta = 1 + \frac{n}{4} + (\frac{n}{4} - 1) * n + \frac{n}{4}\lceil log_2(6 * {}^{n}\!/_{4})\rceil \tag{13}$$

Additionaly, we used the following python program to prove that our method works for n=16 bits:

```
import math

for D in range(0,2**16):
    D = bin(D)[2:].zfill(16)

    # Break into leaf nodes
    L_L = int(D[0:4],2) # 4 MSB
    L_R = int(D[4:8],2)
    R_L = int(D[8:12],2)
    R_R = int(D[12:16],2) # 4 LSB

    # compute Q and R at each leaf node
    R_L_L = (2**12)*(L_L%7)
    Q_L_L = ((2**12) * math.floor(L_L/7)) + math.floor( R_L_L/7 )

    R_L_R = (2**8)*(L_R%7)
    Q_L_R = ((2**8) * math.floor(L_R/7)) + math.floor( R_L_R/7 )

    R_R_L = (2**4)*(R_L%7)
    Q_R_L = ((2**4) * math.floor(R_L/7)) + math.floor( R_R_L/7 )

    Q_R_R = math.floor(R_R/7)
    R_R_R = (R_R%7)

    # Add partial Q's at the first SU
    Q_L = Q_L_L + Q_L_R
    Q_R = Q_R_L + Q_R_R

    # Add parial Q's and correction term at second SU
    # compute remainder
    Q = Q_L + Q_R + math.floor( ((R_L_L%7)+(R_L_R%7)+(R_R_L%7)+(R_R_R%7))/7 )
    R = (R_L_L + R_L_R + R_R_L + R_R_R) % 7

    # check answer
    ans = (7*Q)+R
    if ans != int(D,2):
        print(f'D: {D}={int(D,2)}, ans: {ans}, Q: {Q}, R: {R}')
        break
```

Implementation and Simulation using Quartus II

Once we began implementing the design outlined in the **Design Problem** portion of the report, we realized many optimizations that could be made. First, no 16 bit RCA's needed to be used in the final design, since the maximum output quotient is 14 bits. Moreover, the RCA used to stitch up the outputs of the Qa lookup tables for i=4,8 could be 8 bits. This reduced the number of FA's used by 12.

Next, we realized that we could use the input of the Q_r logic function as the input to the R logic function as well. This reduced the number of FA's used in the final design by another 10.

Below is a hierarchical display of each level of our design. It begins with the top level design, the complete Div7 circuit for a 16 bit input. Then, we display each sub circuit of the top level design, going from left to right.

Quartus Schematics for 16-bit Div7

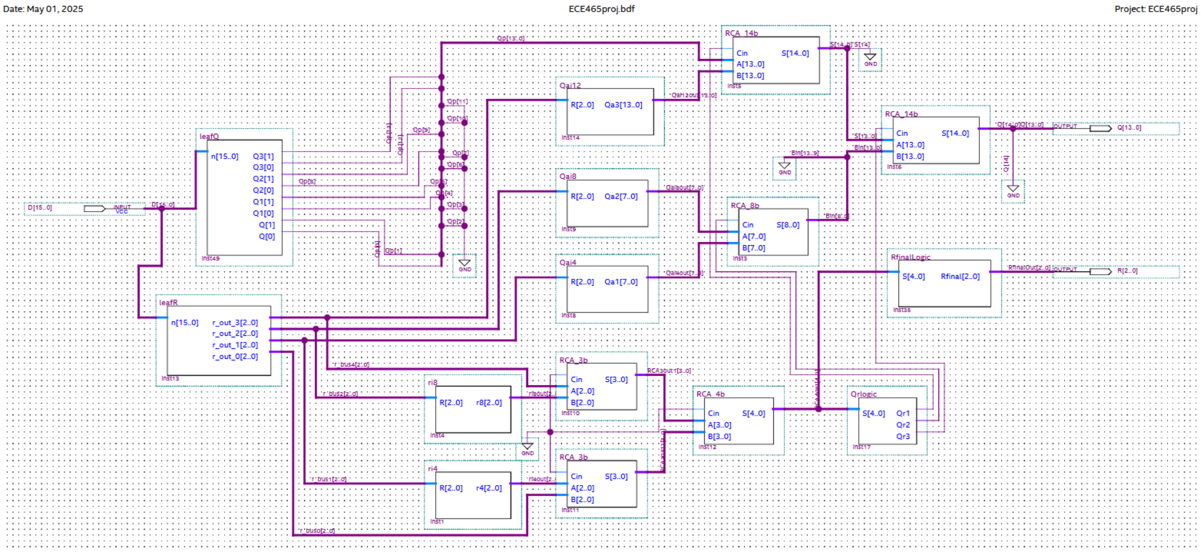


Figure 6: Top level 16-bit Div7 circuit

Figure 6 displays our complete circuit for the 16-bit Div7 problem.

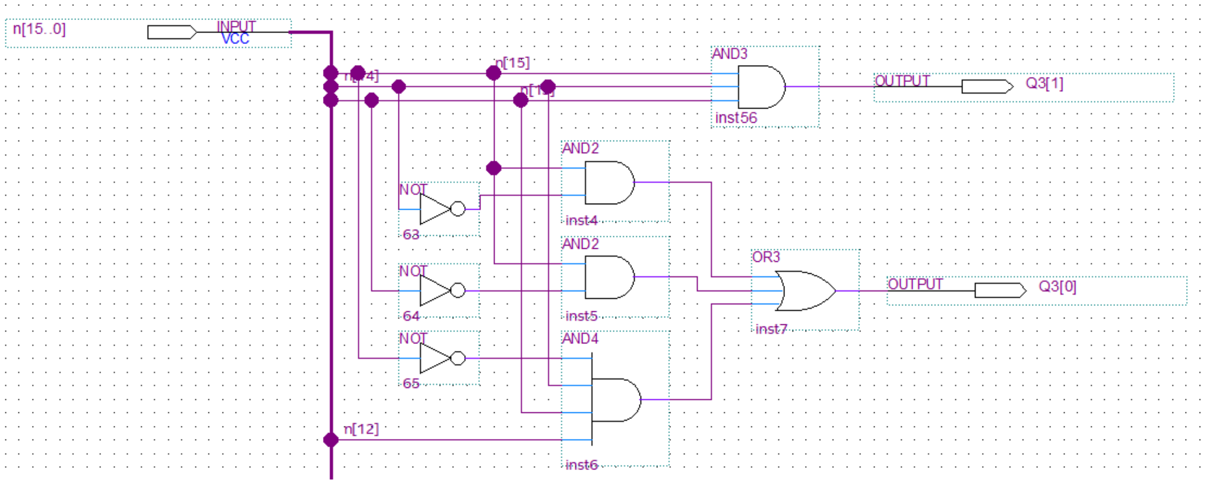


Figure 7: Quotient leaf function

Figure 7 displays our leaf function for the quotient. The design was derived by finding the minimized SOP expression for the Q output of the TT provided in Figure 2. The logic seen in figure 7 is reproduced four times, for each of the 4 bit D&C breakups.

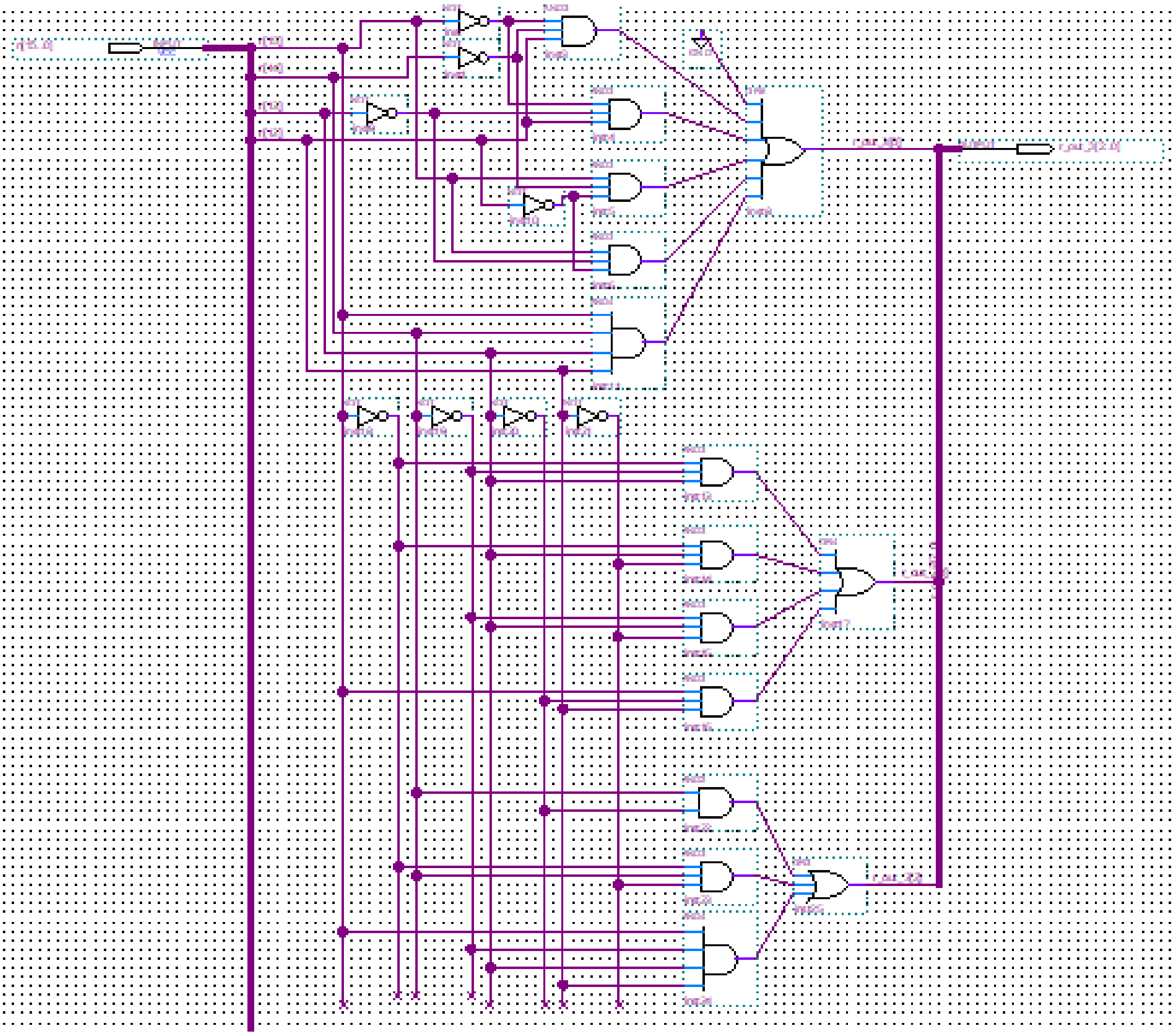


Figure 8: Remainder leaf function

Figure 8 displays our leaf function for the remainder. Similarly to the quotient leaf function in Figure 7, the design was derived by finding the minimized SOP expression for the R output of the TT provided in Figure 2. The logic seen in figure 8 is reproduced four times, for each of the 4 bit D&C breakups.

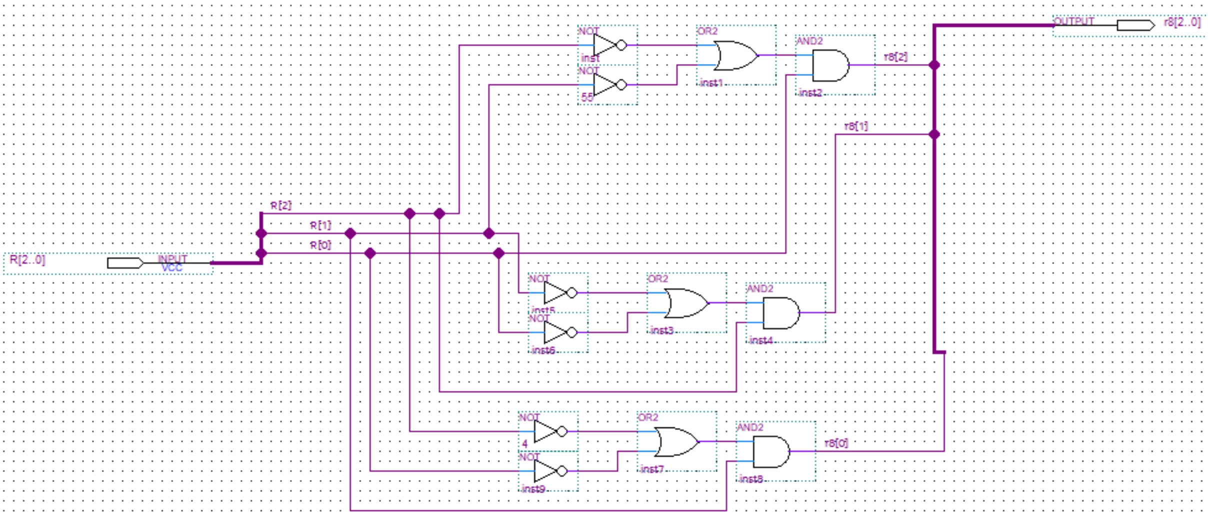


Figure 9: Combinatorial logic for i=8 terms seen in figure 4

Another simplification we made to the **Design Problem** portion of the project was to not hardcode the terms seen in Figure 4. Instead, we derived minimized SOP expressions to generate the terms seen in i=4,8 of that table. Figure 9 displays the logic for generating the i=8 terms.

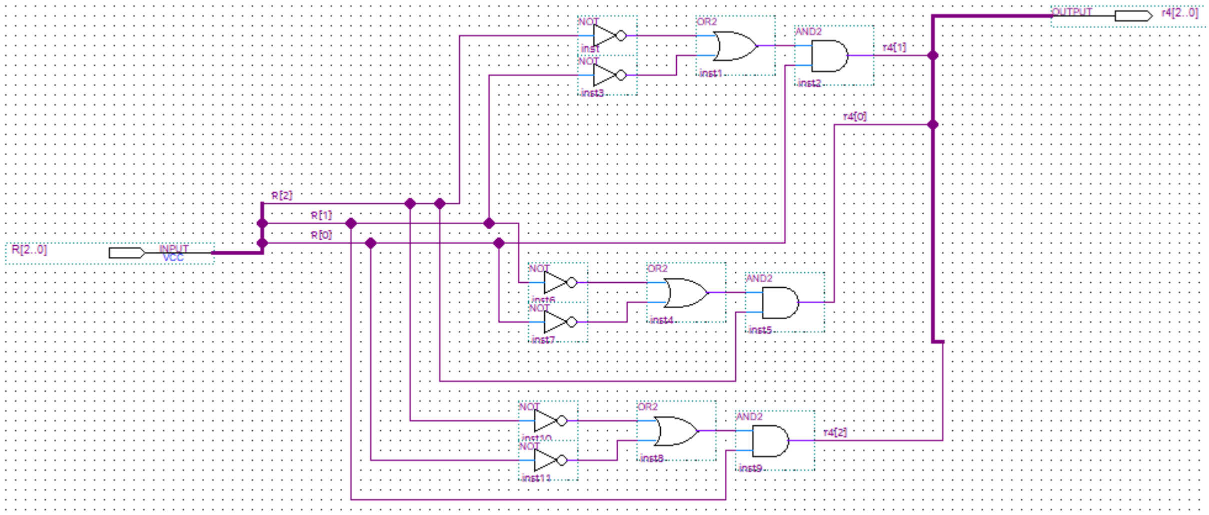


Figure 10: Combinatorial logic for i=4 terms seen in figure 4

Figure 10 displays the minimized SOP logic for generating the i=4 terms seen in Figure 4.

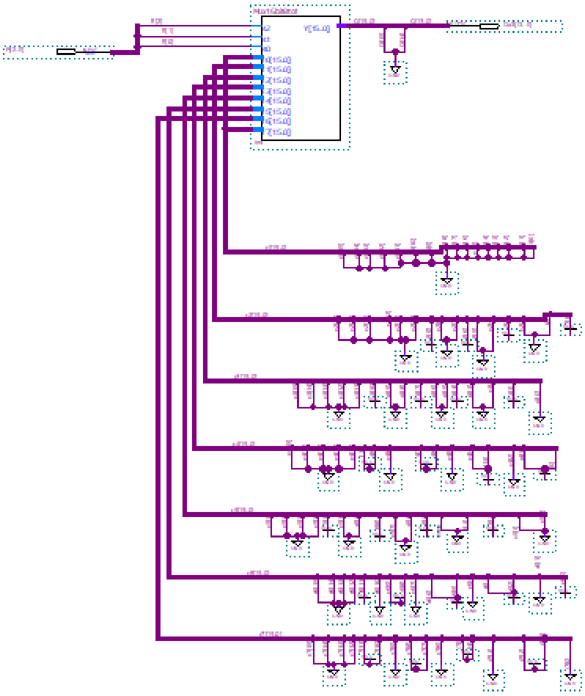


Figure 11: Look up table (LUT) for Q_a i=12

Figure 11 implements the hardcoded LUT seen in Figure 3 for i=12. The values are fed into a 16 bit mux, and selected using the remainder output of the 4 MSB's of the 16 bit input.

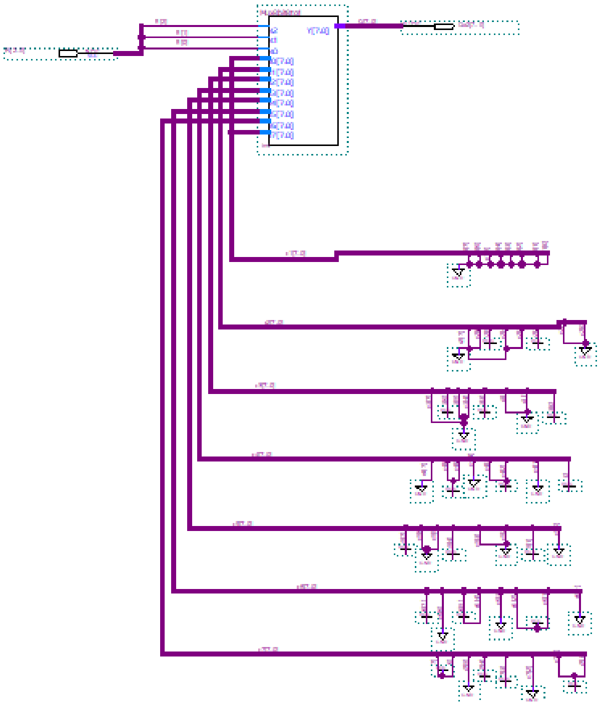


Figure 12: Look up table (LUT) for Q_a i=8

Figure 12 implements the hardcoded LUT seen in Figure 3 for i=8. The values are fed into an 8 bit mux, and selected using the remainder output of the 4 LSB's of the 8 MSB's of the 16 bit input.

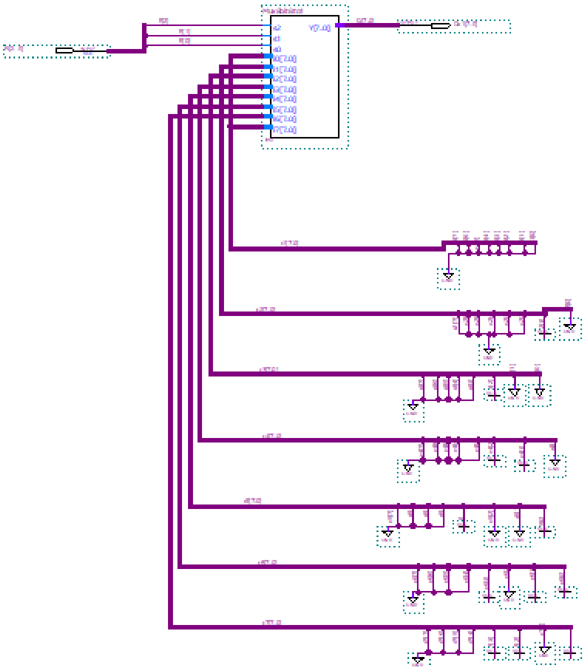


Figure 13: Look up table (LUT) for Q_a i=4

Figure 13 implements the hardcoded LUT seen in Figure 3 for i=4. The values are fed into an 8 bit mux, and selected using the remainder output of the 4 MSB's of the 8 LSB's of the 16 bit input.

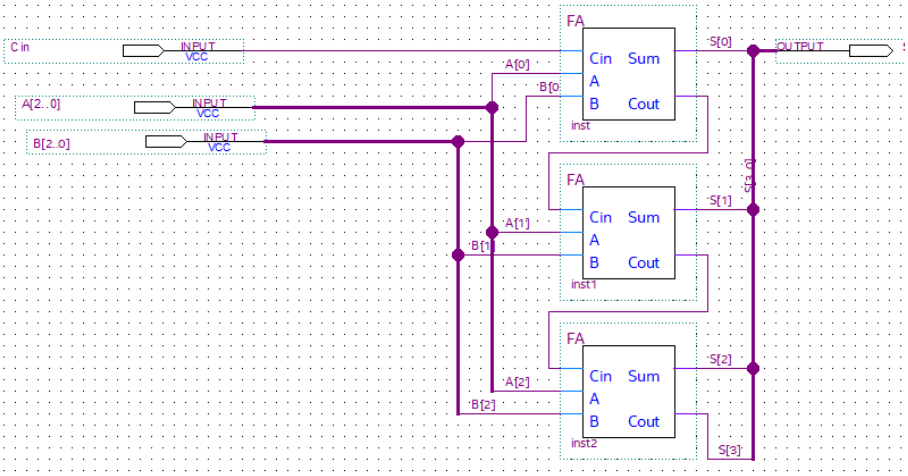


Figure 14: 3 bit RCA

Above is a 3 bit RCA, it is used in part of the remainder stitch up.

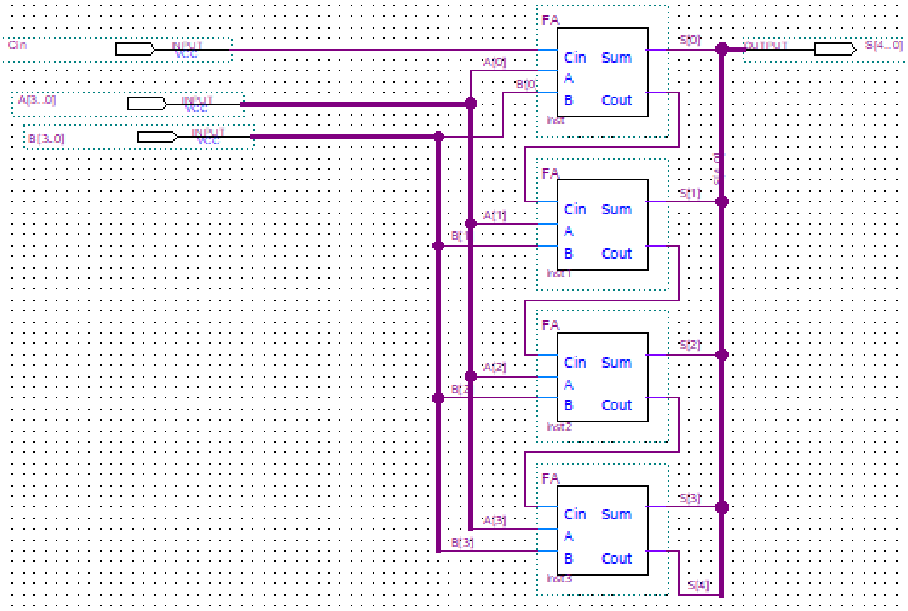


Figure 15: 4 bit RCA

Above is a 4 bit RCA, it is used in part of the remainder stitch up.

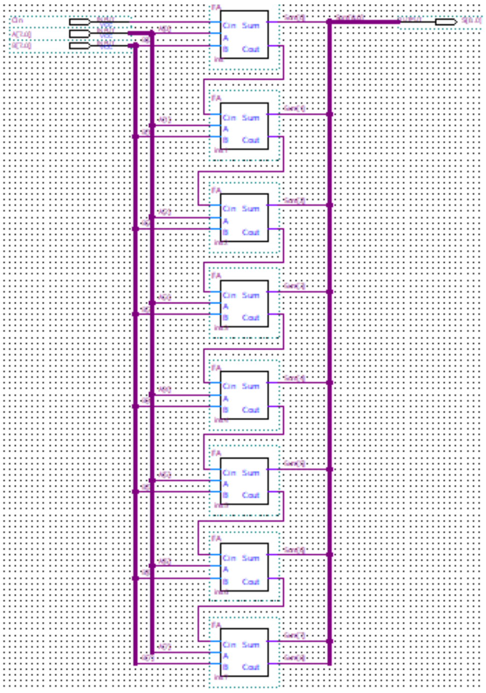


Figure 16: 8 bit RCA

Above is an 8 bit RCA, it is used to stitch up the outputs of the i=4,8 Q_a LUT's.

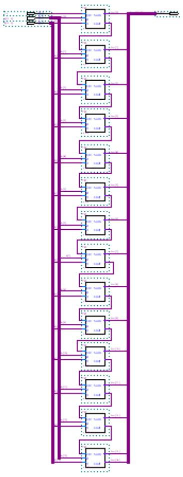


Figure 17: 14 bit RCA

Above is a 14 bit RCA, it is used to stitch up the output of the i=12 Q_a LUT with the outputs of the quotient leaf function and the output of the 8 bit RCA.

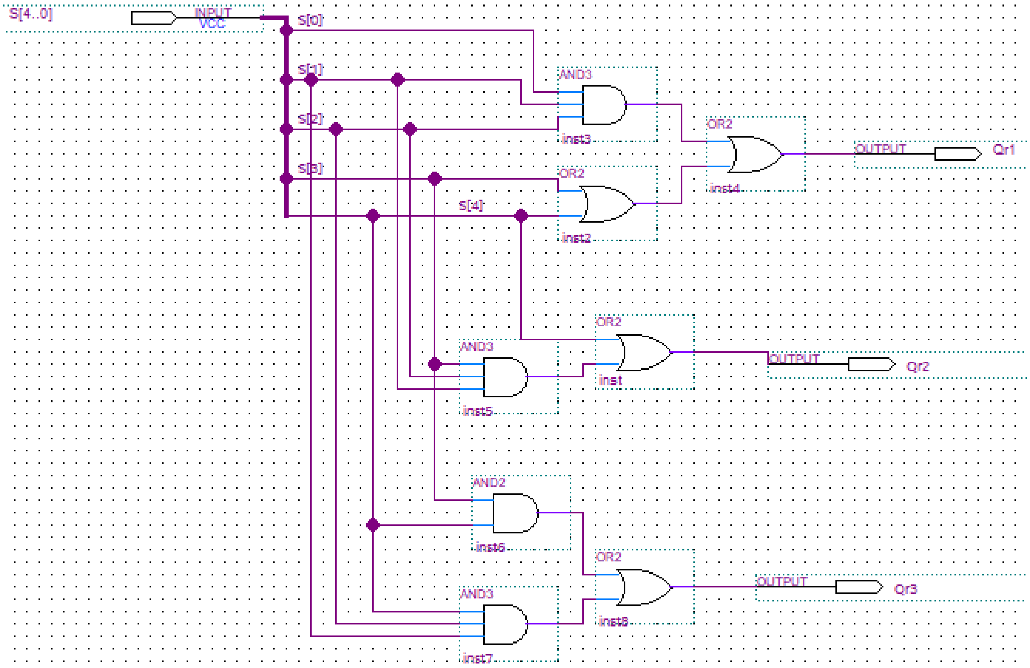


Figure 18: Logic for outputting Q_r

Figure 18 displays the logic used to compute the Q_r output described in the **Design Problem** portion of the project. Its input is the summed output of the values from Figure 4

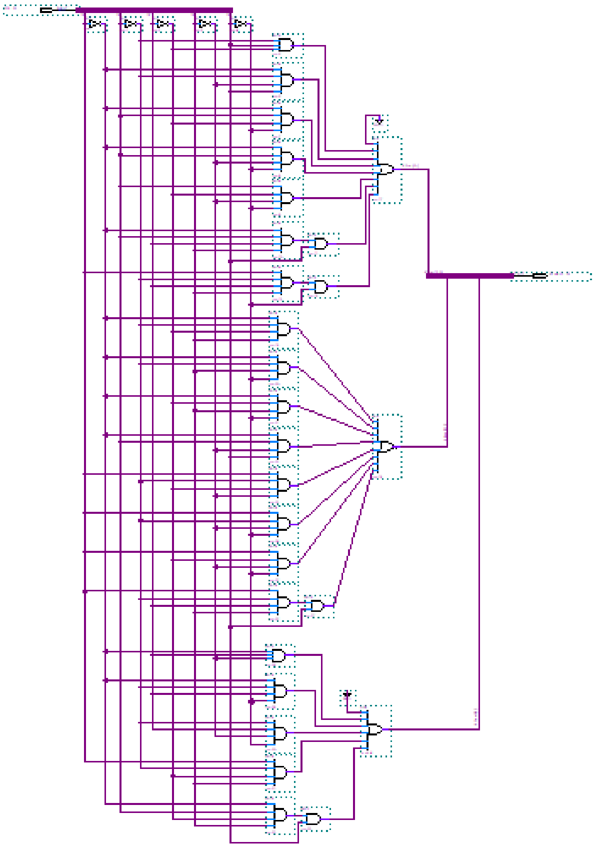


Figure 19: Final logic for computing R

Figure 19 shows the final logic for computing the remainder. It shared the same input as the Q_r logic block in Figure 18

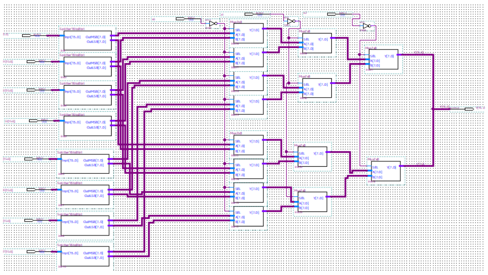


Figure 20: 16 bit 8 to 1 mux

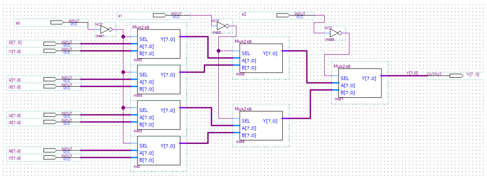


Figure 21: 8 bit 8 to 1 mux

Figure 20 and 21 are the multiplexers used to select the correct values in the Q_a LUT's.

16-bit Div7 Timing Simulation Results

Our design produces correct outputs for all the inputs in the provided waveform. Below are the complete output waveforms for the given inputs:

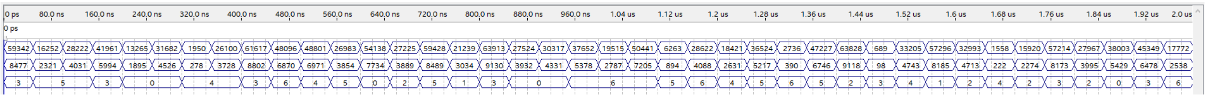


Figure 22: Functional outputs for TA waveform

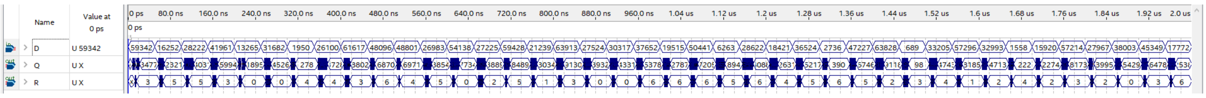


Figure 23: Timing outputs for TA waveform

Delays for 4 inputs including the maximum output delay one:

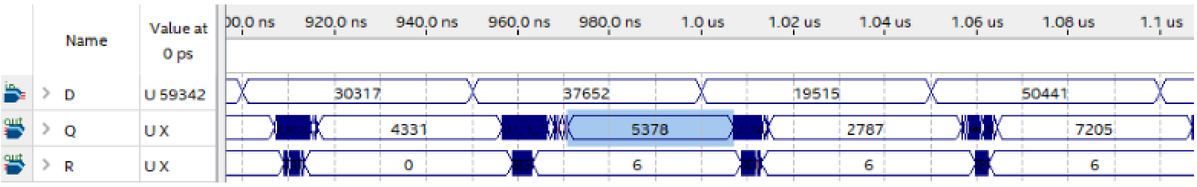

Name
Value at 0 ps
D
U 59342
Q
U X
R
U X
00.0 ns 920.0 ns 940.0 ns 960.0 ns 980.0 ns 1.0 us 1.02 us 1.04 us 1.06 us 1.08 us 1.1 us
30317 37652 19515 50441
4331 5378 2787 7205
0 6 6 6

Figure 24: 4 inputs values and their outputs

Max output delay (D=37652 Q=5378): 970.72 ns - 950.0 ns = 20.72 ns
Output delay 2 (D=30317 Q=4331): 916.27 ns - 900.0 ns= 16.27 ns
Output delay 3 (D=19515 Q=2787): 1000 ns - 1010 ns = 10 ns
Output delay 4 (D=50441 Q=7205): 1060 ns - 1050 ns = 10 ns

Conversion to VHDL and Area and Delay Reports from Synopsys's Design Vision

Here, we display each schematic for 16 bit Div7 after design compilation in Design Vision.

Quartus Schematics for 16-bit Div7

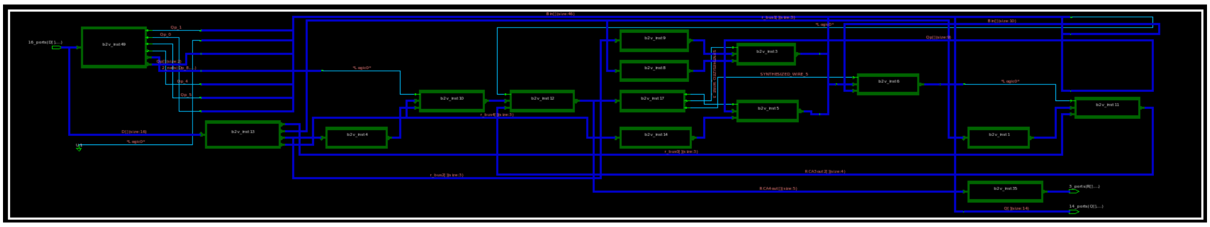


Figure 25: Top level 16-bit Div7 circuit

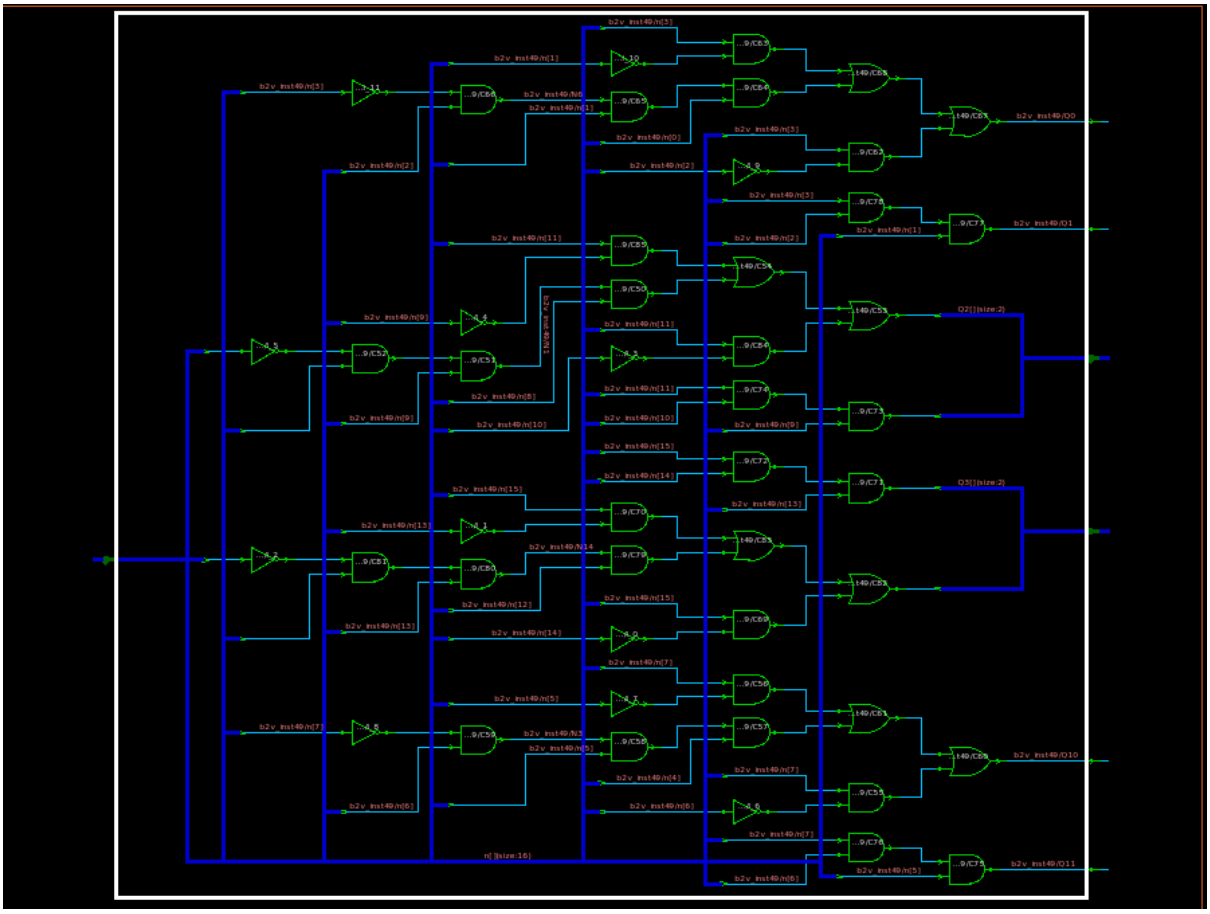


Figure 26: Quotient leaf function

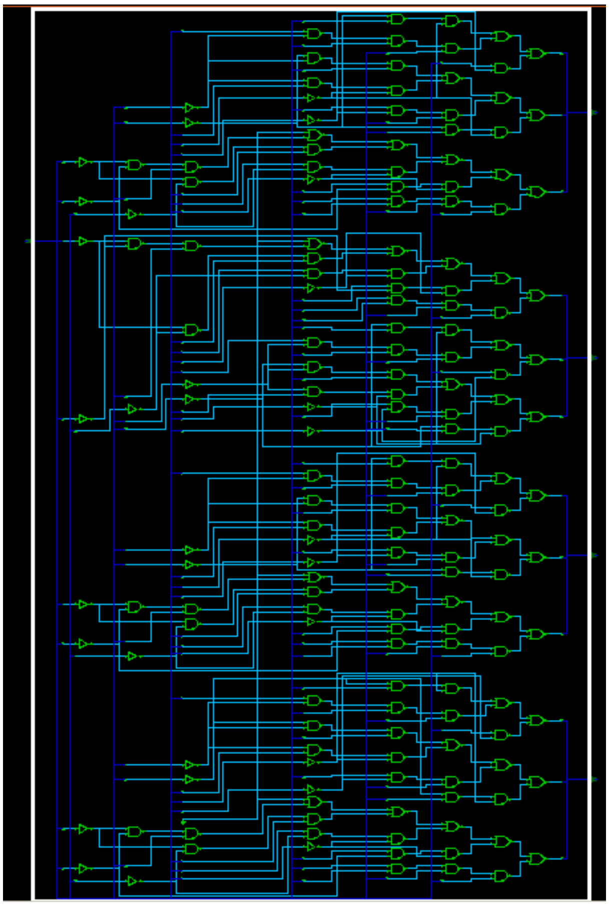


Figure 27: Remainder leaf function

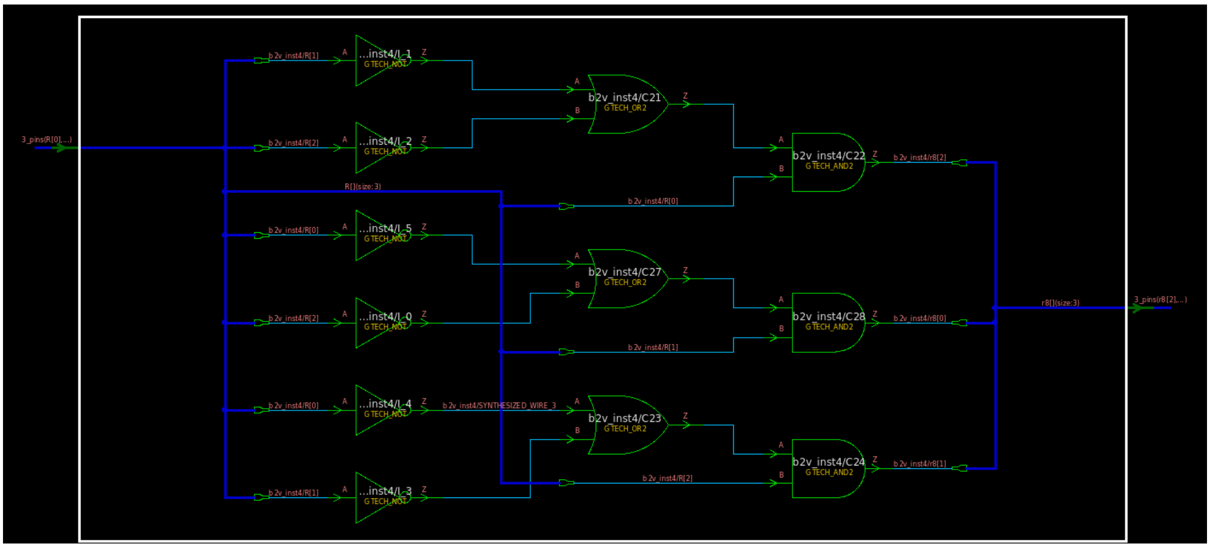


Figure 28: Combinatorial logic for i=8 terms seen in Figure 4

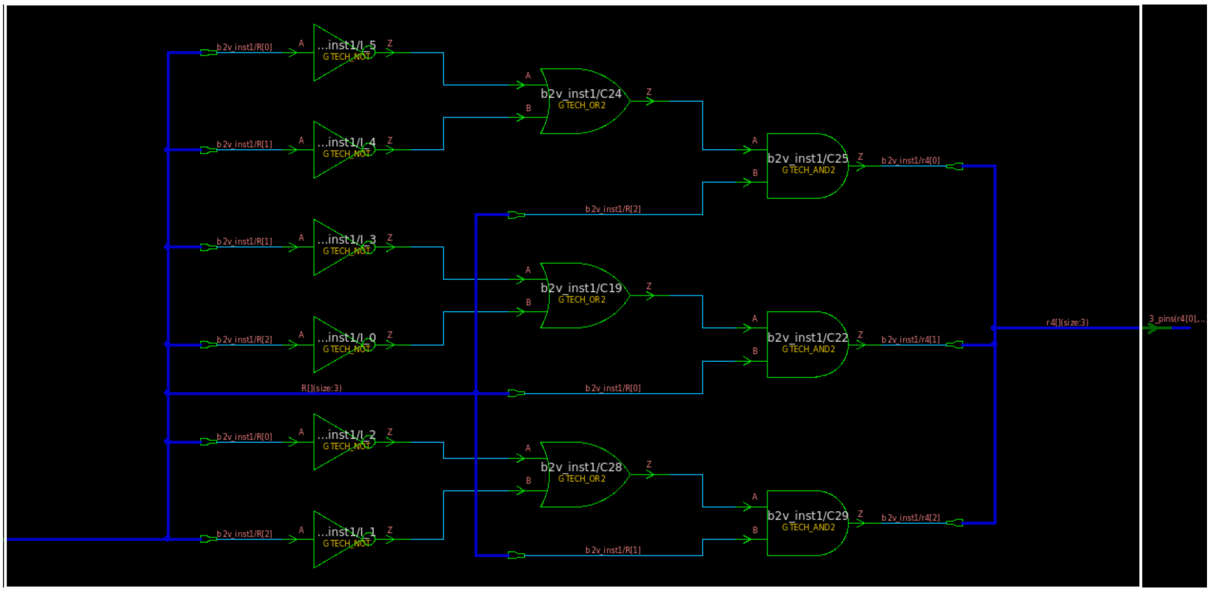


Figure 29: Combinatorial logic for i=4 terms seen in Figure 4

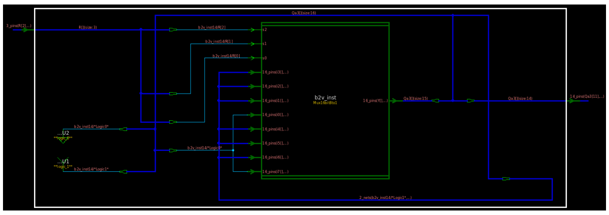


Figure 30: Look up table (LUT) for Q_a i=12

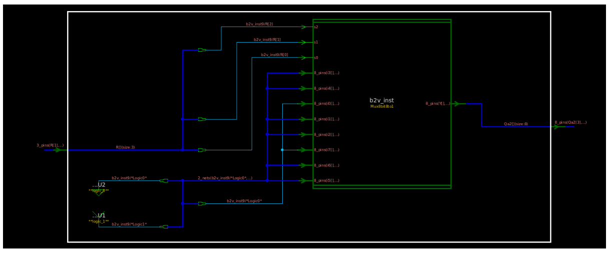


Figure 31: Look up table (LUT) for Q_a i=8

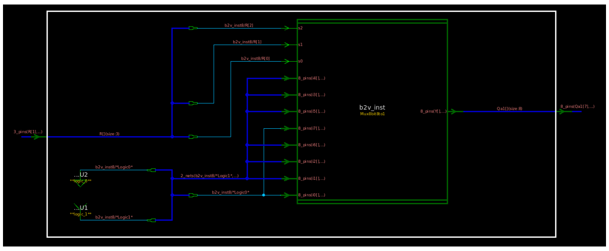


Figure 32: Look up table (LUT) for Q_a i=4

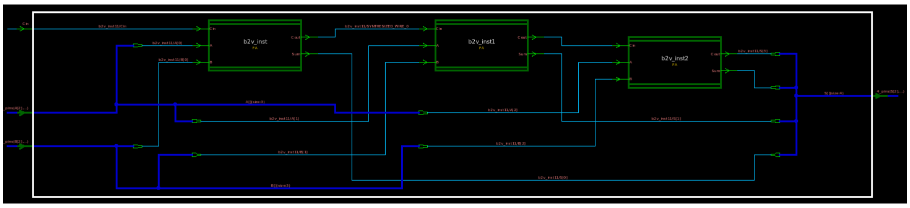


Figure 33: 3 bit RCA

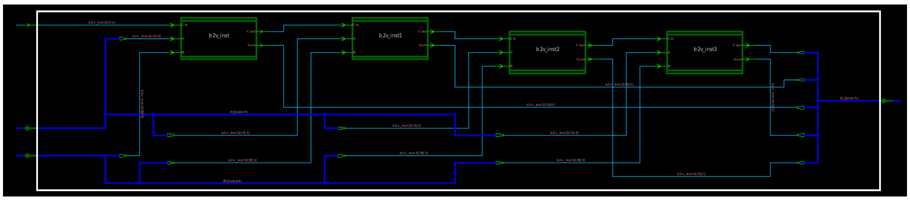


Figure 34: 4 bit RCA

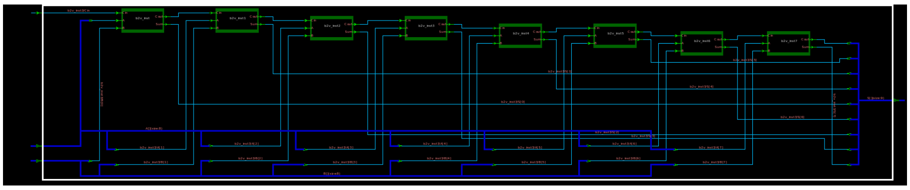


Figure 35: 8 bit RCA

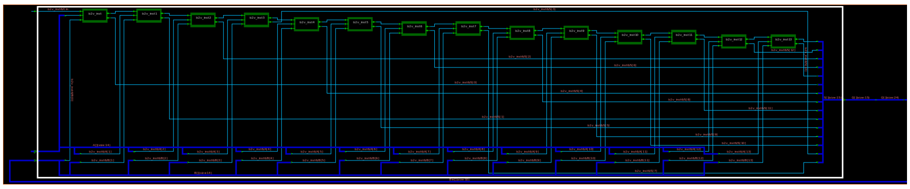


Figure 36: 14 bit RCA

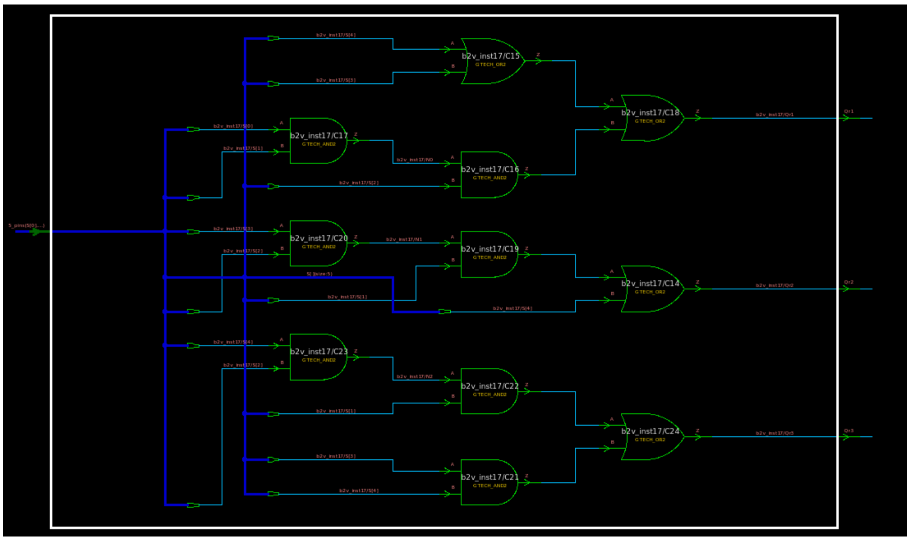


Figure 37: Logic for outputting Q_r

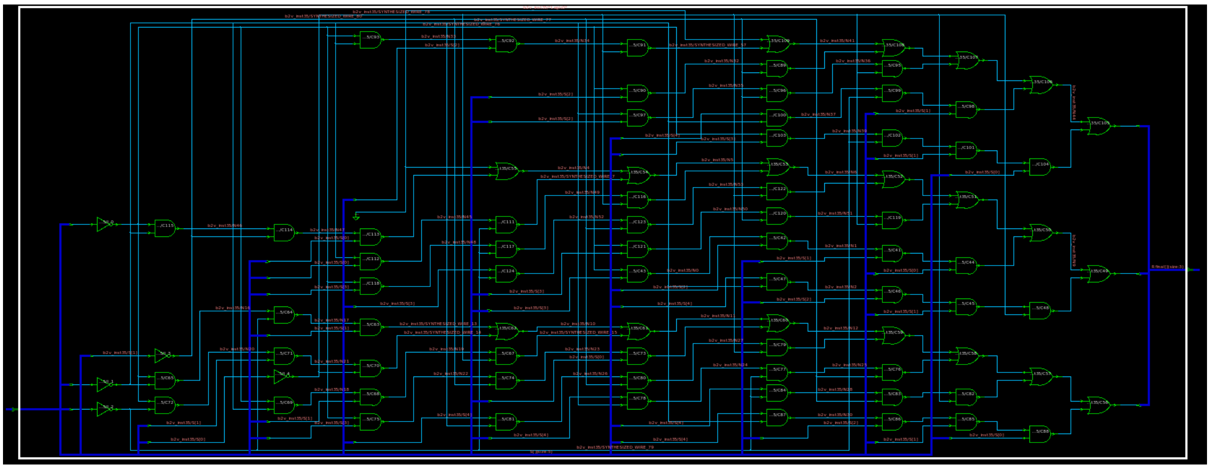


Figure 38: Final logic for computing R

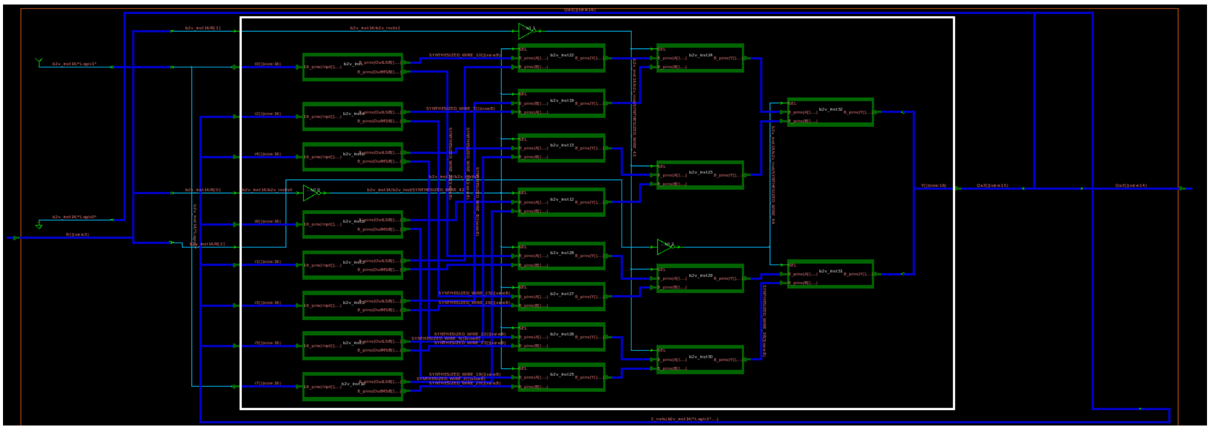


Figure 39: 16 bit 8 to 1 mux

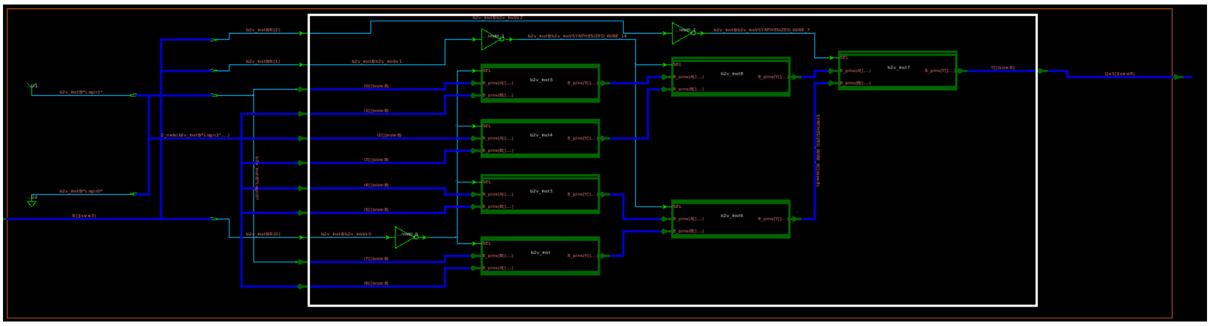


Figure 40: 8 bit 8 to 1 mux

Timing, Area, and Power Results for 16, 8, and 4 bit Div7

n	Delay [ns]	Dynamic Power [μW]	A_1	A_2
4	0.12	1.1572	38.629889	42.532300
8	0.75	29.5579	451.868039	513.793952
16	1.17	115.91	1518.510422	1862.181970

Table 1: Design Vision Results

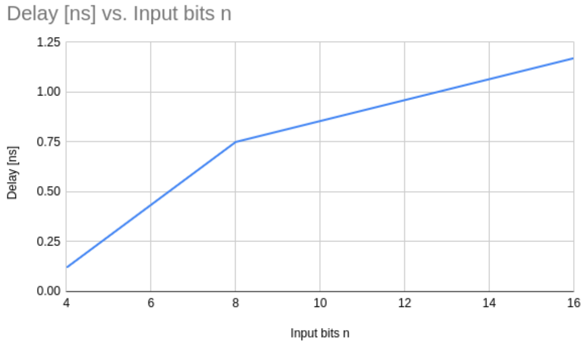


Figure 41: Delay vs n

The delay increases with input size n. However, the jump from n = 4 to n = 8 is greater than from n = 8 to n = 16 implying a nonlinear trend. This may reflect circuit optimizations that hinder performance more at lower bit-widths. Our analytical expression obtained in part 1, Equation 11, contains both a logarithmic term and a linear term. While the logarithmic term introduces slight nonlinearity, the linear term dominates growth for larger n, resulting in an approximately linear delay trend overall. This measured trend is consistent with what was obtained in the delay analysis.

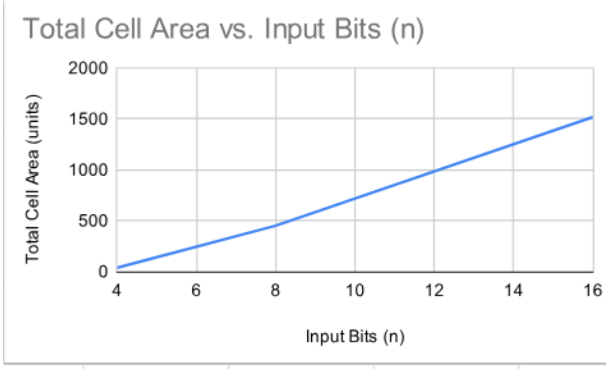


Figure 42: Total Cell Area vs n

From the plot, the total cell area increases sharply with input size n. The increase from n = 4 to n = 8 is over 10x, and from n = 8 to n = 16 is more than 3x suggesting superlinearity. This trend implies that if the number of components increases, but each components bit-width also increases with n, then the total area may grow quadratically. In part 1, our derived analytical expression for the area, Equation 13, was $O(n^2)$. Thus, the total theoretical cost grows quadratically with n and is consistent with what was observed experimentally. The expression includes a linear number of terms (n/4) each scaled by either a linear (n) or logarithmic (log2(n)) factor, resulting in overall superlinear (and roughly quadratic O(n^2)) behavior. This confirms that the derived cost function accurately models the scaling behavior of the DIV7 circuit.

Appendix

16-bit Div7 timing simulation verbatim results:

```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
        -sort_by group
Design : ECE465proj
Version: Q-2019.12-SP3
Date   : Thu May  1 15:55:47 2025
****************************************

Operating Conditions: dlvl_ff1p16v125c_i1p16v   Library: saed32rvt_dlvl_ff1p16v125c_i1p16v
Wire Load Model Mode: enclosed

    Startpoint: D[15] (input port)
    Endpoint: Q[13] (output port)
    Path Group: (none)
    Path Type: max

    Des/Clust/Port     Wire Load Model       Library
    ------------------------------------------------
    ECE465proj         8000                  saed32rvt_dlvl_ff1p16v125c_i1p16v
    leafR              ForQA                 saed32rvt_dlvl_ff1p16v125c_i1p16v
    Mux16bit8to1       8000                  saed32rvt_dlvl_ff1p16v125c_i1p16v
    FA_27              ForQA                 saed32rvt_dlvl_ff1p16v125c_i1p16v
    RCA_14b_0          ForQA                 saed32rvt_dlvl_ff1p16v125c_i1p16v
    FA_2               ForQA                 saed32rvt_dlvl_ff1p16v125c_i1p16v
    RCA_14b_1          ForQA                 saed32rvt_dlvl_ff1p16v125c_i1p16v

    Point                                                   Incr       Path
    --------------------------------------------------------------------------
    input external delay                                    0.00       0.00 f
    D[15] (in)                                              0.00       0.00 f
    b2v_inst13/n[15] (leafR)                                0.00       0.00 f
    b2v_inst13/U11/Y (XOR2X1_RVT)                           0.05       0.05 r
    b2v_inst13/U8/Y (MUX21X1_RVT)                           0.04       0.09 r
    b2v_inst13/r_out_3[0] (leafR)                           0.00       0.09 r
    b2v_inst14/R[0] (Qai12)                                 0.00       0.09 r
    b2v_inst14/b2v_inst/s0 (Mux16bit8to1)                   0.00       0.09 r
    b2v_inst14/b2v_inst/U1/Y (INVX0_RVT)                    0.34       0.42 f
    b2v_inst14/b2v_inst/b2v_inst12/SEL (Mux2x8_0)           0.00       0.42 f
    b2v_inst14/b2v_inst/b2v_inst12/U7/Y (MUX21X1_RVT)       0.09       0.52 f
    b2v_inst14/b2v_inst/b2v_inst12/Y[1] (Mux2x8_0)          0.00       0.52 f
    b2v_inst14/b2v_inst/b2v_inst23/B[1] (Mux2x8_24)         0.00       0.52 f
    b2v_inst14/b2v_inst/b2v_inst23/U7/Y (MUX21X1_RVT)       0.04       0.56 f
    b2v_inst14/b2v_inst/b2v_inst23/Y[1] (Mux2x8_24)         0.00       0.56 f
    b2v_inst14/b2v_inst/b2v_inst32/B[1] (Mux2x8_15)         0.00       0.56 f
    b2v_inst14/b2v_inst/b2v_inst32/U7/Y (MUX21X1_RVT)       0.04       0.60 f
    b2v_inst14/b2v_inst/b2v_inst32/Y[1] (Mux2x8_15)         0.00       0.60 f
    b2v_inst14/b2v_inst/Y[1] (Mux16bit8to1)                 0.00       0.60 f
    b2v_inst14/Qa3[1] (Qai12)                               0.00       0.60 f
    b2v_inst5/B[1] (RCA_14b_0)                              0.00       0.60 f
    b2v_inst5/b2v_inst1/B (FA_27)                           0.00       0.60 f
```

```
b2v_inst5/b2v_inst1/U3/Y (XOR2X1_RVT)                    0.05       0.65 r
b2v_inst5/b2v_inst1/U2/Y (AO22X1_RVT)                    0.03       0.68 r
b2v_inst5/b2v_inst1/Cout (FA_27)                         0.00       0.68 r
b2v_inst5/b2v_inst2/Cin (FA_22)                          0.00       0.68 r
b2v_inst5/b2v_inst2/U2/Y (AO22X2_RVT)                    0.03       0.71 r
b2v_inst5/b2v_inst2/Cout (FA_22)                         0.00       0.71 r
b2v_inst5/b2v_inst3/Cin (FA_21)                          0.00       0.71 r
b2v_inst5/b2v_inst3/U2/Y (AO22X2_RVT)                    0.03       0.75 r
b2v_inst5/b2v_inst3/Cout (FA_21)                         0.00       0.75 r
b2v_inst5/b2v_inst4/Cin (FA_20)                          0.00       0.75 r
b2v_inst5/b2v_inst4/U2/Y (AO22X1_RVT)                    0.03       0.78 r
b2v_inst5/b2v_inst4/Cout (FA_20)                         0.00       0.78 r
b2v_inst5/b2v_inst5/Cin (FA_19)                          0.00       0.78 r
b2v_inst5/b2v_inst5/U2/Y (AO22X1_RVT)                    0.03       0.81 r
b2v_inst5/b2v_inst5/Cout (FA_19)                         0.00       0.81 r
b2v_inst5/b2v_inst6/Cin (FA_18)                          0.00       0.81 r
b2v_inst5/b2v_inst6/U2/Y (AO22X2_RVT)                    0.03       0.84 r
b2v_inst5/b2v_inst6/Cout (FA_18)                         0.00       0.84 r
b2v_inst5/b2v_inst7/Cin (FA_17)                          0.00       0.84 r
b2v_inst5/b2v_inst7/U2/Y (AO22X2_RVT)                    0.03       0.88 r
b2v_inst5/b2v_inst7/Cout (FA_17)                         0.00       0.88 r
b2v_inst5/b2v_inst8/Cin (FA_16)                          0.00       0.88 r
b2v_inst5/b2v_inst8/U1/Y (XOR2X1_RVT)                    0.04       0.92 f
b2v_inst5/b2v_inst8/Sum (FA_16)                          0.00       0.92 f
b2v_inst5/S[8] (RCA_14b_0)                               0.00       0.92 f
b2v_inst6/A[8] (RCA_14b_1)                               0.00       0.92 f
b2v_inst6/b2v_inst8/A (FA_2)                             0.00       0.92 f
b2v_inst6/b2v_inst8/U3/Y (XOR2X1_RVT)                    0.05       0.97 r
b2v_inst6/b2v_inst8/U2/Y (AO22X1_RVT)                    0.03       0.99 r
b2v_inst6/b2v_inst8/Cout (FA_2)                          0.00       0.99 r
b2v_inst6/b2v_inst9/Cin (FA_1)                           0.00       0.99 r
b2v_inst6/b2v_inst9/U2/Y (AO22X2_RVT)                    0.03       1.03 r
b2v_inst6/b2v_inst9/Cout (FA_1)                          0.00       1.03 r
b2v_inst6/b2v_inst10/Cin (FA_12)                         0.00       1.03 r
b2v_inst6/b2v_inst10/U2/Y (AO22X2_RVT)                   0.03       1.06 r
b2v_inst6/b2v_inst10/Cout (FA_12)                        0.00       1.06 r
b2v_inst6/b2v_inst11/Cin (FA_11)                         0.00       1.06 r
b2v_inst6/b2v_inst11/U2/Y (AO22X2_RVT)                   0.03       1.10 r
b2v_inst6/b2v_inst11/Cout (FA_11)                        0.00       1.10 r
b2v_inst6/b2v_inst12/Cin (FA_10)                         0.00       1.10 r
b2v_inst6/b2v_inst12/U2/Y (AO22X2_RVT)                   0.03       1.13 r
b2v_inst6/b2v_inst12/Cout (FA_10)                        0.00       1.13 r
b2v_inst6/b2v_inst13/Cin (FA_9)                          0.00       1.13 r
b2v_inst6/b2v_inst13/U1/Y (XOR2X1_RVT)                   0.04       1.17 f
b2v_inst6/b2v_inst13/Sum (FA_9)                          0.00       1.17 f
b2v_inst6/S[13] (RCA_14b_1)                              0.00       1.17 f
Q[13] (out)                                              0.00       1.17 f
data arrival time                                                   1.17
---------------------------------------------------------------------------
(Path is unconstrained)
```

16-bit Div7 power simulation verbatim results:

```
Information: Propagating switching activity (low effort zero delay simulation). (PWR-6)
Warning: There is no defined clock in the design. (PWR-80)
Warning: Design has unannotated primary inputs. (PWR-414)

****************************************
Report : power
        -analysis_effort low
Design : ECE465proj
Version: Q-2019.12-SP3
Date   : Thu May  1 16:02:45 2025
****************************************


Library(s) Used:

    saed32rvt_ff1p16v125c (File: /usr/local/libraries/SAED32_EDK/lib/stdcell_rvt/db_ccs/saed32rvt_ff


Operating Conditions: dlvl_ff1p16v125c_i1p16v   Library: saed32rvt_dlvl_ff1p16v125c_i1p16v
Wire Load Model Mode: enclosed

Design        Wire Load Model            Library
------------------------------------------------
ECE465proj             8000              saed32rvt_dlvl_ff1p16v125c_i1p16v
ri4                    ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
RCA_3b_0               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
RCA_4b                 ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
leafR                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Qai12                  8000              saed32rvt_dlvl_ff1p16v125c_i1p16v
Qrlogic                ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
RCA_8b                 ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
RfinalLogic            ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
ri8                    ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
leafQ                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
RCA_14b_0              ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Qai4                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Qai8                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_0                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux16bit8to1           8000              saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux8bit8to1_0          ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Splitter16to8bit_0     ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_0               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
RCA_3b_1               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
RCA_14b_1              ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_1                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_2                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_3                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_4                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_5                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_6                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_7                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_8                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_9                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_10                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_11                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_12                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_13                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
```

```
FA_14                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_15                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_16                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_17                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_18                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_19                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_20                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_21                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_22                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_23                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_24                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_25                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_26                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_27                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_28                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_29                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_30                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_31                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_32                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_33                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_34                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_35                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_36                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_37                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_38                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_39                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_40                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_41                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_42                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_43                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_44                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_45                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux8bit8to1_1          ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Splitter16to8bit_1     ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Splitter16to8bit_2     ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Splitter16to8bit_3     ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Splitter16to8bit_4     ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Splitter16to8bit_5     ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Splitter16to8bit_6     ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Splitter16to8bit_7     ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_1               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_2               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_3               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_4               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_5               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_6               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_7               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_8               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_9               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_10              ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_11              ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_12              ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_13              ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_14              ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_15              ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_16              ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_17              ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_18              ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
```

```
Mux2x8_19              ForQA              saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_20              ForQA              saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_21              ForQA              saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_22              ForQA              saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_23              ForQA              saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_24              ForQA              saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_25              ForQA              saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_26              ForQA              saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_27              ForQA              saed32rvt_dlvl_ff1p16v125c_i1p16v


Global Operating Voltage = 1.16
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000ff
    Time Units = 1ns
    Dynamic Power Units = 1uW    (derived from V,C,T units)
    Leakage Power Units = 1pW


    Cell Internal Power  =   0.0000 uW    (0%)
    Net Switching Power  = 115.9059 uW  (100%)
                           ---------
Total Dynamic Power    = 115.9059 uW  (100%)

Cell Leakage Power     =  37.2272 uW

Information: report_power power group summary does not include estimated clock tree power. (PWR-789)

                    Internal         Switching           Leakage            Total
Power Group         Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad              0.0000            0.0000            0.0000            0.0000  (   0.00%)
memory              0.0000            0.0000            0.0000            0.0000  (   0.00%)
black_box           0.0000            0.0000            0.0000            0.0000  (   0.00%)
clock_network       0.0000            0.0000            0.0000            0.0000  (   0.00%)
register            0.0000            0.0000            0.0000            0.0000  (   0.00%)
sequential          0.0000            0.0000            0.0000            0.0000  (   0.00%)
combinational       0.0000          115.9060        3.7227e+07          153.1332  ( 100.00%)
--------------------------------------------------------------------------------------------------
Total               0.0000 uW       115.9060 uW     3.7227e+07 pW       153.1332 uW
```

16-bit Div7 area simulation verbatim results:

```
****************************************
Report : area
Design : ECE465proj
Version: Q-2019.12-SP3
Date   : Thu May  1 16:05:24 2025
****************************************

Library(s) Used:

    saed32rvt_ff1p16v125c (File: /usr/local/libraries/SAED32_EDK/lib/stdcell_rvt/db_ccs/saed32rvt_ff

Number of ports:                         1785
Number of nets:                          2247
Number of cells:                          583
Number of combinational cells:            476
Number of sequential cells:                 0
Number of macros/black boxes:               0
Number of buf/inv:                         32
Number of references:                      15

Combinational area:               1518.510422
Buf/Inv area:                       40.663040
Noncombinational area:               0.000000
Macro/Black Box area:                0.000000
Net Interconnect area:             343.671548

Total cell area:                  1518.510422
Total area:                       1862.181970
```

8-bit Div7 area simulation verbatim results:

```
****************************************
Report : area
Design : div78bit
Version: Q-2019.12-SP3
Date   : Thu May  1 22:09:09 2025
****************************************

Library(s) Used:

    saed32rvt_ff1p16v125c (File: /usr/local/libraries/SAED32_EDK/lib/stdcell_rvt/db_ccs/saed32rvt_ff

Number of ports:                          411
Number of nets:                           551
Number of cells:                          179
Number of combinational cells:            148
Number of sequential cells:                 0
Number of macros/black boxes:               0
Number of buf/inv:                         12
Number of references:                       7

Combinational area:                451.868039
Buf/Inv area:                       15.248640
Noncombinational area:               0.000000
Macro/Black Box area:                0.000000
Net Interconnect area:              61.925913

Total cell area:                   451.868039
Total area:                        513.793952
```

8-bit Div7 power simulation verbatim results:

```
Information: Propagating switching activity (low effort zero delay simulation). (PWR-6)
Warning: There is no defined clock in the design. (PWR-80)
Warning: Design has unannotated primary inputs. (PWR-414)

****************************************
Report : power
        -analysis_effort low
Design : div78bit
Version: Q-2019.12-SP3
Date   : Thu May  1 22:08:57 2025
****************************************


Library(s) Used:

    saed32rvt_ff1p16v125c (File: /usr/local/libraries/SAED32_EDK/lib/stdcell_rvt/db_ccs/saed32rvt_ff


Operating Conditions: dlvl_ff1p16v125c_i1p16v   Library: saed32rvt_dlvl_ff1p16v125c_i1p16v
Wire Load Model Mode: enclosed

Design        Wire Load Model            Library
------------------------------------------------
div78bit               8000              saed32rvt_dlvl_ff1p16v125c_i1p16v
leafr                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
rca_3b                 ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
leafq                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
qrlogic                ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
rca_8b                 ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
rfinallogic            ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
qai4                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_0                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
mux8bit8to1            ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_0               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_1                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_2                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_3                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_4                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_5                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_6                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_7                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_8                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_9                   ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
FA_10                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_1               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_2               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_3               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_4               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_5               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
Mux2x8_6               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v


Global Operating Voltage = 1.16
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000ff
    Time Units = 1ns
    Dynamic Power Units = 1uW    (derived from V,C,T units)
    Leakage Power Units = 1pW


  Cell Internal Power  =   0.0000 uW    (0%)
  Net Switching Power  =  29.5579 uW  (100%)
                         ---------
Total Dynamic Power    =  29.5579 uW  (100%)

Cell Leakage Power     =  11.2917 uW

Information: report_power power group summary does not include estimated clock tree power. (PWR-789)

                 Internal         Switching           Leakage            Total
Power Group      Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad             0.0000           0.0000             0.0000             0.0000  (   0.00%)
memory             0.0000           0.0000             0.0000             0.0000  (   0.00%)
black_box          0.0000           0.0000             0.0000             0.0000  (   0.00%)
clock_network      0.0000           0.0000             0.0000             0.0000  (   0.00%)
register           0.0000           0.0000             0.0000             0.0000  (   0.00%)
sequential         0.0000           0.0000             0.0000             0.0000  (   0.00%)
combinational      0.0000          29.5579          1.1292e+07           40.8497  ( 100.00%)
--------------------------------------------------------------------------------------------------
Total              0.0000 uW       29.5579 uW     1.1292e+07 pW        40.8497 uW
```

8-bit Div7 timing simulation verbatim results:

```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
        -sort_by group
Design : div78bit
Version: Q-2019.12-SP3
Date   : Thu May  1 22:08:00 2025
****************************************

Operating Conditions: dlvl_ff1p16v125c_i1p16v   Library: saed32rvt_dlvl_ff1p16v125c_i1p16v
Wire Load Model Mode: enclosed

  Startpoint: D[7] (input port)
  Endpoint: Q[5] (output port)
  Path Group: (none)
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  div78bit           8000                  saed32rvt_dlvl_ff1p16v125c_i1p16v
  leafr              ForQA                 saed32rvt_dlvl_ff1p16v125c_i1p16v
  mux8bit8to1        ForQA                 saed32rvt_dlvl_ff1p16v125c_i1p16v
  FA_8               ForQA                 saed32rvt_dlvl_ff1p16v125c_i1p16v
  rca_8b             ForQA                 saed32rvt_dlvl_ff1p16v125c_i1p16v

  Point                                                   Incr       Path
  --------------------------------------------------------------------------
  input external delay                                    0.00       0.00 f
  D[7] (in)                                               0.00       0.00 f
  b2v_inst/n_in[7] (leafr)                                0.00       0.00 f
  b2v_inst/U11/Y (XOR2X1_RVT)                             0.05       0.05 r
  b2v_inst/U8/Y (MUX21X1_RVT)                             0.04       0.09 r
  b2v_inst/r_out_1[0] (leafr)                             0.00       0.09 r
  b2v_inst8/R[0] (qai4)                                   0.00       0.09 r
  b2v_inst8/b2v_inst/s0 (mux8bit8to1)                     0.00       0.09 r
  b2v_inst8/b2v_inst/U1/Y (INVX0_RVT)                     0.26       0.35 f
  b2v_inst8/b2v_inst/b2v_inst/SEL (Mux2x8_0)              0.00       0.35 f
  b2v_inst8/b2v_inst/b2v_inst/U8/Y (MUX21X1_RVT)          0.08       0.43 f
  b2v_inst8/b2v_inst/b2v_inst/Y[0] (Mux2x8_0)             0.00       0.43 f
  b2v_inst8/b2v_inst/b2v_inst6/B[0] (Mux2x8_3)            0.00       0.43 f
  b2v_inst8/b2v_inst/b2v_inst6/U8/Y (MUX21X1_RVT)         0.04       0.47 f
  b2v_inst8/b2v_inst/b2v_inst6/Y[0] (Mux2x8_3)            0.00       0.47 f
  b2v_inst8/b2v_inst/b2v_inst7/B[0] (Mux2x8_2)            0.00       0.47 f
  b2v_inst8/b2v_inst/b2v_inst7/U8/Y (MUX21X1_RVT)         0.04       0.51 f
  b2v_inst8/b2v_inst/b2v_inst7/Y[0] (Mux2x8_2)            0.00       0.51 f
  b2v_inst8/b2v_inst/Y[0] (mux8bit8to1)                   0.00       0.51 f
  b2v_inst8/Qa1[0] (qai4)                                 0.00       0.51 f
  b2v_inst3/B[0] (rca_8b)                                 0.00       0.51 f
  b2v_inst3/b2v_inst/B (FA_8)                             0.00       0.51 f
  b2v_inst3/b2v_inst/U3/Y (XOR2X1_RVT)                    0.05       0.56 r
  b2v_inst3/b2v_inst/U2/Y (AO22X1_RVT)                    0.03       0.59 r
  b2v_inst3/b2v_inst/Cout (FA_8)                          0.00       0.59 r
  b2v_inst3/b2v_inst1/Cin (FA_7)                          0.00       0.59 r
```

```
b2v_inst3/b2v_inst1/U2/Y (AO22X1_RVT)                  0.03       0.62 r
b2v_inst3/b2v_inst1/Cout (FA_7)                        0.00       0.62 r
b2v_inst3/b2v_inst2/Cin (FA_6)                         0.00       0.62 r
b2v_inst3/b2v_inst2/U2/Y (AO22X2_RVT)                  0.03       0.65 r
b2v_inst3/b2v_inst2/Cout (FA_6)                        0.00       0.65 r
b2v_inst3/b2v_inst3/Cin (FA_5)                         0.00       0.65 r
b2v_inst3/b2v_inst3/U2/Y (AO22X2_RVT)                  0.03       0.68 r
b2v_inst3/b2v_inst3/Cout (FA_5)                        0.00       0.68 r
b2v_inst3/b2v_inst4/Cin (FA_4)                         0.00       0.68 r
b2v_inst3/b2v_inst4/U2/Y (AO22X1_RVT)                  0.03       0.71 r
b2v_inst3/b2v_inst4/Cout (FA_4)                        0.00       0.71 r
b2v_inst3/b2v_inst5/Cin (FA_3)                         0.00       0.71 r
b2v_inst3/b2v_inst5/U1/Y (XOR2X1_RVT)                  0.04       0.75 f
b2v_inst3/b2v_inst5/Sum (FA_3)                         0.00       0.75 f
b2v_inst3/S[5] (rca_8b)                                0.00       0.75 f
Q[5] (out)                                             0.00       0.75 f
data arrival time                                                 0.75
---------------------------------------------------------------------------
(Path is unconstrained)
```

4-bit Div7 area simulation verbatim results:

```
****************************************
Report : area
Design : div74bit
Version: Q-2019.12-SP3
Date   : Thu May  1 20:57:24 2025
****************************************

Library(s) Used:

    saed32rvt_ff1p16v125c (File: /usr/local/libraries/SAED32_EDK/lib/stdcell_rvt/db_ccs/saed32rvt_ff

Number of ports:                           22
Number of nets:                            32
Number of cells:                           17
Number of combinational cells:             15
Number of sequential cells:                 0
Number of macros/black boxes:               0
Number of buf/inv:                          2
Number of references:                       2

Combinational area:                 38.629889
Buf/Inv area:                        2.541440
Noncombinational area:               0.000000
Macro/Black Box area:                0.000000
Net Interconnect area:               3.902411

Total cell area:                    38.629889
Total area:                         42.532300
```

4-bit Div7 power simulation verbatim results:

```
Information: Propagating switching activity (low effort zero delay simulation). (PWR-6)
Warning: There is no defined clock in the design. (PWR-80)
Warning: Design has unannotated primary inputs. (PWR-414)

****************************************
Report : power
        -analysis_effort low
Design : div74bit
Version: Q-2019.12-SP3
Date   : Thu May  1 20:57:33 2025
****************************************


Library(s) Used:

    saed32rvt_ff1p16v125c (File: /usr/local/libraries/SAED32_EDK/lib/stdcell_rvt/db_ccs/saed32rvt_ff


Operating Conditions: dlvl_ff1p16v125c_i1p16v   Library: saed32rvt_dlvl_ff1p16v125c_i1p16v
Wire Load Model Mode: enclosed

Design        Wire Load Model            Library
------------------------------------------------
div74bit               ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
leafq                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v
leafr                  ForQA             saed32rvt_dlvl_ff1p16v125c_i1p16v


Global Operating Voltage = 1.16
Power-specific unit information :
    Voltage Units = 1V
    Capacitance Units = 1.000000ff
    Time Units = 1ns
    Dynamic Power Units = 1uW    (derived from V,C,T units)
    Leakage Power Units = 1pW


    Cell Internal Power  =   0.0000 uW    (0%)
    Net Switching Power  =   1.1572 uW  (100%)
                           ---------
Total Dynamic Power    =   1.1572 uW  (100%)

Cell Leakage Power     =   1.0500 uW

Information: report_power power group summary does not include estimated clock tree power. (PWR-789)

                    Internal         Switching           Leakage            Total
Power Group         Power            Power               Power              Power   (   %    )  Attrs
--------------------------------------------------------------------------------------------------
io_pad              0.0000           0.0000            0.0000             0.0000  (   0.00%)
memory              0.0000           0.0000            0.0000             0.0000  (   0.00%)
black_box           0.0000           0.0000            0.0000             0.0000  (   0.00%)
clock_network       0.0000           0.0000            0.0000             0.0000  (   0.00%)
register            0.0000           0.0000            0.0000             0.0000  (   0.00%)
sequential          0.0000           0.0000            0.0000             0.0000  (   0.00%)
combinational       0.0000           1.1572        1.0500e+06             2.2072  ( 100.00%)
--------------------------------------------------------------------------------------------------
```

```
Total              0.0000 uW         1.1572 uW     1.0500e+06 pW         2.2072 uW
```

4-bit Div7 timing simulation verbatim results:

```
Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
        -sort_by group
Design : div74bit
Version: Q-2019.12-SP3
Date   : Thu May  1 20:57:10 2025
****************************************

Operating Conditions: dlvl_ff1p16v125c_i1p16v   Library: saed32rvt_dlvl_ff1p16v125c_i1p16v
Wire Load Model Mode: enclosed

  Startpoint: D[3] (input port)
  Endpoint: R[2] (output port)
  Path Group: (none)
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  div74bit           ForQA                 saed32rvt_dlvl_ff1p16v125c_i1p16v
  leafr              ForQA                 saed32rvt_dlvl_ff1p16v125c_i1p16v

  Point                                    Incr       Path
  -------------------------------------------------------------
  input external delay                     0.00       0.00 r
  D[3] (in)                                0.00       0.00 r
  b2v_inst3/n_in[3] (leafr)                0.00       0.00 r
  b2v_inst3/U11/Y (XOR2X1_RVT)             0.05       0.05 f
  b2v_inst3/U3/Y (OAI21X1_RVT)             0.04       0.09 r
  b2v_inst3/U2/Y (MUX21X1_RVT)             0.03       0.12 r
  b2v_inst3/r_out_0[2] (leafr)             0.00       0.12 r
  R[2] (out)                               0.00       0.12 r
  data arrival time                                   0.12
  -------------------------------------------------------------
  (Path is unconstrained)
```