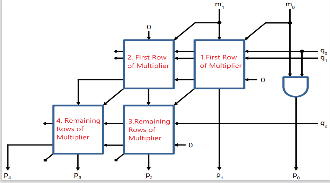
# 2-bit by 3-bit Multiplier Assessment

The following files are provided:

|  |  |
| --- | --- |
| File | Purpose |
| Multiplier2x3.v | Main Module for 2bit x 3bit Multiplier |
| MultiplerFirstRow.v | Submodule which comprises first row of multiplier |
| MultiplerRemainingRow.v | Submodule which comprises remaining rows of multiplier |
| Multiplier2x3.sdc | Generic timing constraints file |
| Structure/\* | Diagrams of the multiplier structure |
| simulation/Multiplier2x3\_tb.v | Test bench of the 2bit x 3bit Multiplier |
| Adder1Bit.v | Submodule which comprises 1-bit Adder |

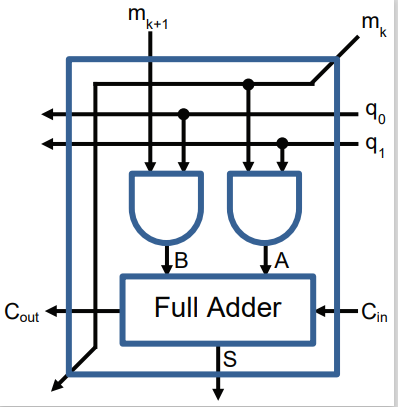
#### Implementation principle

* + Before claim the theory of this multiplier, we need to figure out how much variable we need:
    - m : 2 bits multiplier, its range is from 0 to 3 .
    - q : 3 bits multiplier, its range is from 0 to 7.
    - p : 5 bits result, its range is from 0 to 31.
  + After knowing the variables of multiplier, we need to know the structure of the 2-Bit by 3-Bit Multiplier. As the figure shown above, we can see there have 4 parts need us to achieve. Thus, we need to figure out what input and output of these multiplier in submodule.

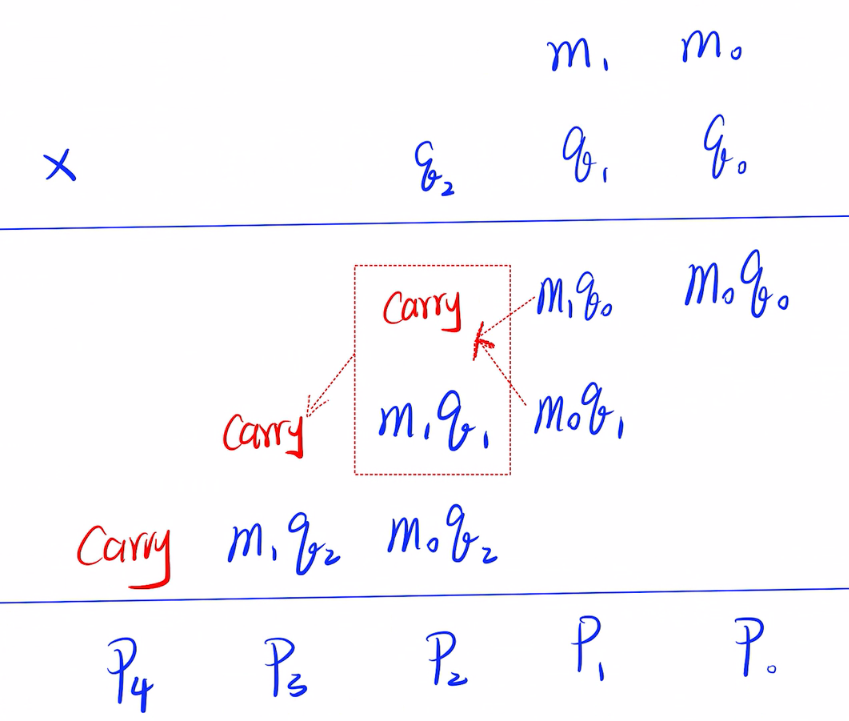
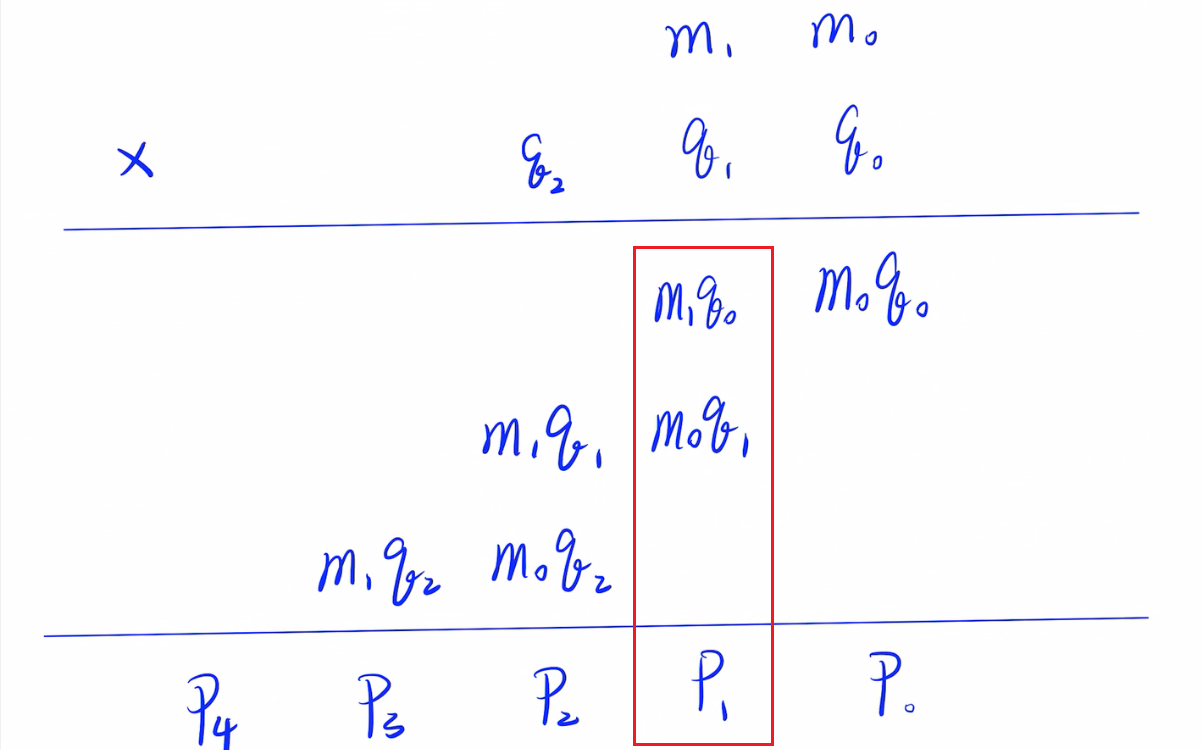
##### To calculate the p0, the p0 is easy to calculate since it is equal to m0 x q0.

* + To calculate the p1, according to the structure of first row of multiplier, we can see the Function of this struct is to calculate S and Cout through above table.

|  |  |
| --- | --- |
| Variable | How to calculate |
| A | q1 x mk |
| B | q0 x mk+1 |
| S | A + B = q1 x mk + q0 x mk+1 |
| Cout | Carry of S |

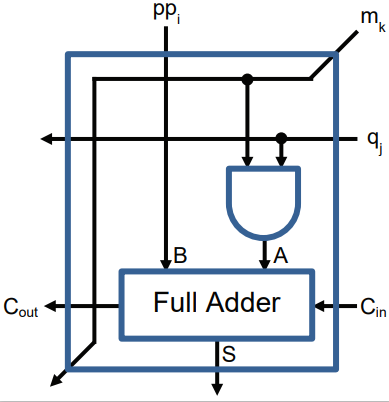


* + - According to the principle of above figure, the picture shown below illustrate the theory of the 2 bits by 3 bits multiplier. The red frame shows below is just like the function of 1th.First Row of Multiplier (Also it generate the carry to 2th.First Row of Multiplier) .



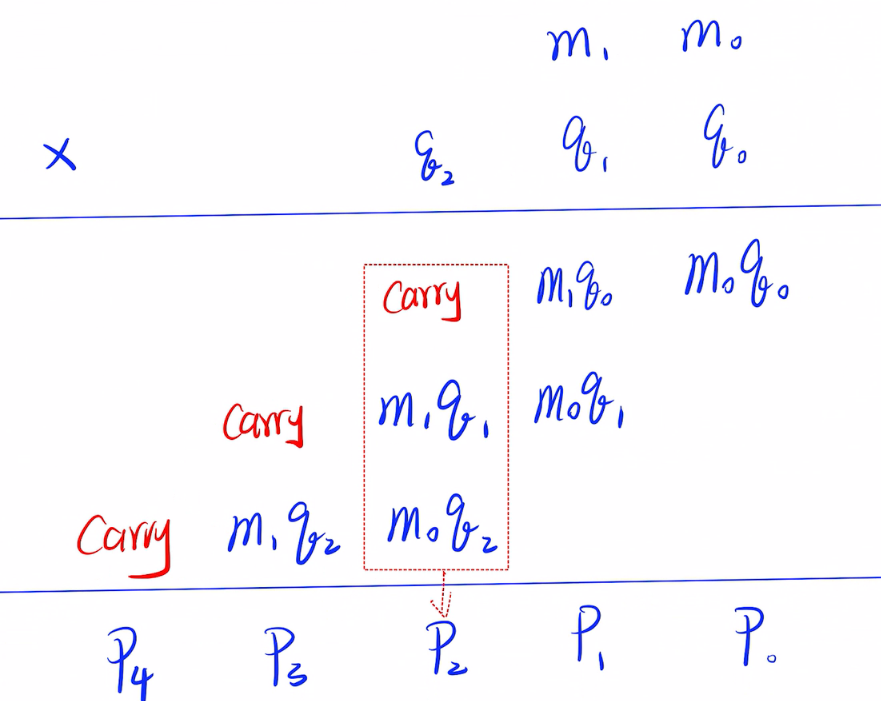
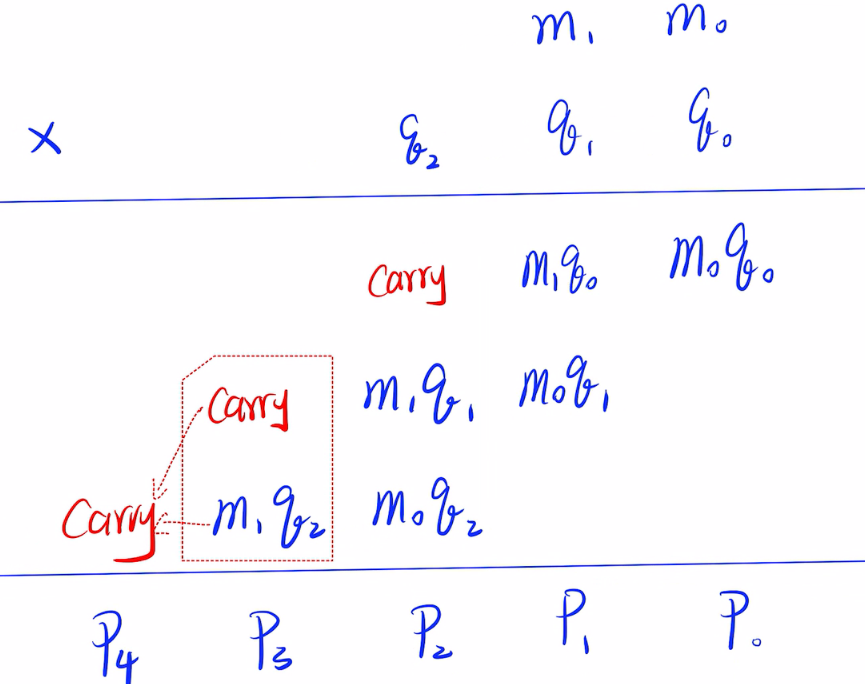
* + The main function of 2.First Row of Multiplier is to add the carry of last part (calculate P1) and m1 x q1 to generate the S and carry to 3th.Remaining Rows of Multiplier to calculated (As the figure shown above left).
  + To calculate the p3 and p4, we need to see the structure of Remaining Rows of Multiplier:

|  |  |
| --- | --- |
| Variable | How to calculate |
| A | qj x mk |
| B | ppi = S in the previous line |
| S | A+B=(qj x mk) + ppi |
| Cout | carry of the S |



* + We can see the Function of this struct is to calculate S and Cout through above table.

##### According to the principle of above figure, the picture shown below left illustrate the theory of the 2 bits by 3 bits multiplier. The red frame shows below is to show the main function of 3rd.Remaining Rows of Multiplier is to calculate p2 = carry + m1 x q1 + m0 x q2.

##### As for p3 and p4, it is be calculated in 4th. Remaining Rows of Multiplier. The 4th. Remaining Rows of Multiplier function is mainly using the carry calculated from 3rd. Remaining Rows of Multiplier and m1 x q2 to calculate p3 and generate the carry to act as p4 (as figure shown above right).

#### Output and Simulation

* + The mainly function of the test bench of the code is to simulation the process of 2 bits by 3 bits multiplier. The figure below is the output of simulation using the code of test bench, from which we can see when i = 2 and j = 5, the m would be equal to 2 , q would be equal to 5 and it can be calculated that both of the p and expected\_value are 10, which shows that the algorithm of multiplier2x3 is correct.

