Jonathan Whitaker, U0752100
CS 3700-001
Lab 3

4-bit Ripple-Carry Adder

The synthesized 4-bit ripple-carry adder is composed of 20 *Slice LUTs* (see the synthesis report below), so we can consider the area $Area_{4\text{-}bit\ ripple}=20$. We would expect the critical path delay to be from either *Cin* to *Cout* or from X_0 *to Cout.* If we denote the delay of a single full-adder as Δt, then a 4-bit ripple carry adder has a critical path delay of $4\Delta t$, because it has to pass through 4 full-adders. The synthesis report for the 4-bit ripple carry adder is shown below.

```
-------------------------------------------------------------------
Slice Logic Utilization:
 Number of Slice LUTs:                   20  out of  9112     0%
    Number used as Logic:                20  out of  9112     0%
-------------------------------------------------------------------
Delay:               11.826ns (Levels of Logic = 20)
  Source:            X<0> (PAD)
  Destination:       Cout (PAD)
-------------------------------------------------------------------
```

From the timing synthesis report, you can see exactly what was expected. The critical path delay was from X_0 to C_{out} , and the total delay was *11.826ns*. This means that each full-adder had a total delay of $\Delta t = \frac{11.826ns}{4} = 2.9565ns$. Likewise, there were 20 logic levels that were traversed to go from X_0 and *Cout*, and this makes sense if you account for each of the buffers on the input lines. There should be 3 buffers total for the input signals per full-adder, and 2 levels of logic within the full-adder itself. This accounts for a total of 5 levels of logic per full-adder. We have 4 full-adders rippled together, so we should expect $4(5) = 20$ levels of logic, which is exactly what the synthesis report conveyed.

4-bit Carry-Select Adder

In comparison with the 4-bit ripple-carry adder, the 4-bit carry-select adder should be close to 2 times the total area of the ripple carry adder, namely $Area_{4-bit\ carry} = 2Area_{4-bit\ ripple} = 40$. Likewise, the critical path delay should increase when compared to a 4-bit ripple carry adder. This is due to the added multiplexors that the carry-select adder is implemented with. After synthesizing my implementation, I received the following synthesis report:

```
---------------------------------------------------------------------
Slice Logic Utilization:
 Number of Slice LUTs:                   41  out of  9112     0%
    Number used as Logic:                41  out of  9112     0%
---------------------------------------------------------------------
Delay:               11.957ns (Levels of Logic = 21)
  Source:            X<0> (PAD)
  Destination:       Cout (PAD)
---------------------------------------------------------------------
```

Again, the synthesis report shows almost exactly what I expected. There were $2Area_{4-bit\ carry} + 1 = 41$ *Slice LUTs* utilized, which is extremely close to the expected area of $2Area_{4-bit\ ripple} = 40$. I imagine the added LUT was due to some optimization that occurred, but I am not certain of this. Also, the timing report showed that the critical path was again from X_0 to C_{out} , but the critical path delay was *11.957ns*. The critical path delay increased slightly when compared to the 4-bit ripple-carry adder, and the added delay is due to the multiplexors. If $t_{4-bit\ ripple}$ and $t_{4-bit\ carry}$ denote the critical path delay of the 4-bit ripple-carry and 4-bit carry-select adders, and if Δm denotes the delay of the multiplexors, then $t_{4-bit\ carry} - t_{4-bit\ ripple} = \Delta m = 11.957ns - 11.826ns = 0.131ns$. So the multiplexors add an additional

0.131ns to the total delay. Δm will help with the comparison between the 16-bit ripple carry and the 16-bit carry-select adder analysis.

16-bit Ripple-Carry & Carry-Select Adders

Both the 16-bit ripple-carry and carry-select adders should show extremely similar characteristics to their 4-bit counterpart. For example, we know the area and timing analysis involved with a 4-bit ripple-carry adder, so the 16-bit ripple-carry implementation should have nearly 4 times the area as its counterpart, and nearly 4 times the critical path delay. The same argument holds for the 16-bit carry-select implementation. Below is the synthesis report for both of these adders.

16-bit Ripple-Carry

```
-------------------------------------------------------------------
Slice Logic Utilization:
 Number of Slice LUTs:                   80  out of  9112     0%
    Number used as Logic:                80  out of  9112     0%
-------------------------------------------------------------------
Delay:               31.112ns (Levels of Logic = 72)
  Source:            X<0> (PAD)
  Destination:       Cout (PAD)
-------------------------------------------------------------------
```

You can see that the utilization of the *Slice LUTs* is 4 times that of the 4-bit ripple-carry adder design, namely $Area_{16-bit\ ripple} = 4Area_{4-bit\ ripple} = 80$. This is exactly as expected, because I simply connected 4x4-bit ripple-carry adders together. The timing analysis is unique though, because I expected 4 times the critical path delay of the 4-bit ripple-carry adder, but the total

delay was actually less than the expected delay of $4(11.286ns) = 47.304ns$. Again, I imagine that this difference in timing is largely due to the synthesis tool optimizing the circuit.

16-bit Carry-Select

```
---------------------------------------------------------------
Slice Logic Utilization:
 Number of Slice LUTs:                  164  out of  9112
1%
    Number used as Logic:               164  out of  9112
1%
---------------------------------------------------------------
Delay:               15.401ns (Levels of Logic = 31)
  Source:            X<0> (PAD)
  Destination:       Cout (PAD)

Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------  ------------
 IBUF:I->O             4   1.222   0.788  X_0_IBUF (X_0_IBUF)
 OBUF:I->O                 2.571          Cout_OBUF (Cout)
---------------------------------------------------------------
```

The 16-bit carry-select adder is defined hierarchically in term of its 4-bit counterpart. As I expected, the *Slice LUTs* utilization is 4 times that of its 4-bit carry-select counterpart, namely $Area_{16-bit\ carry} = 4Area_{4-bit\ carry} = 164$. Likewise, I expected the critical path delay to be given by $t_{4-bit\ carry} + 3(\Delta m) = 11.957 + 3(0.131) = 12.088ns$. This expected critical path delay is due to the delay of the first 4-bit carry-select adder, and the 3 multiplexors following it. The timing synthesis shows results similar to this expected value, but the added delay is a result of the input and output buffers that the synthesis tool adds. I have included the delays associated with these in the synthesis report shown above. As you can see, the input buffer delay and output buffer delay result in a total delay of $1.222ns + 2.571ns = 3.793ns$, so the critical path delay should be the sum between what I expected ($12.088ns$) and this added delay for the buffers

($3.783ns$), namely the total delay should roughly be $12.088ns + 3.793ns = 15.881ns$ which is almost spot on.

Conclusion

Like most things in digital design, there's always a trade-off, and the best circuit choice is dependent upon the given application. Undoubtedly, you wouldn't want to implement a 4-bit adder with a 4-bit carry-select adder, because it experiences a performance hit as well as an increase in area. Though, as you can see in the two synthesis reports for the 16-bit ripple-carry and carry-select adders, the 16-bit ripple-carry adder uses nearly half as much area, but its performance is nearly half as fast as the 16-bit carry-select adder. If area is more critical than performance, then the 16-bit ripple-carry adder is the best choice. In contrast, if performance is more critical than area, then the 16-bit carry-select adder is the best candidate.