APPENDIX A
THE C GATE AND THE VIRTEX-5 FPGA

The Muller C gate is a sequential element which switches only when all its inputs have the same value. When the inputs disagree, the gate output retains its previous value. Any multi-input C gate can be expressed as a cascade of 2-input C gates (henceforth MULLER). This appendix describes the MULLER gate and its realization on a Xilinx Virtex-5 FPGA.

Boolean expressions of sequential elements need special notation to distinguish the current value and next value of signals. We denote both the signal and its current value by the signal name. Hence, a denotes the value of signal 'a' at the present instant. Let a' denote its value at the next instant and $\overline{a}$ its logical negation. Also, let $\oplus$, $\cdot$ and $+$ represent logical XOR, AND and OR respectively.

The MULLER gate can be modelled as a latch which is transparent only when both its inputs agree:

$$\texttt{MULLER}(a,b)' = \overline{(a \oplus b)} \cdot a \ + \ (a \oplus b) \cdot \texttt{MULLER}(a,b) \quad (1)$$

The Xilinx Virtex-5 FPGA contains 4 storage elements in each SLICE. Each storage element can be configured as a level-sensitive latch with input driven by a LUT in the same SLICE. The latch is transparent when the control signal clock `CLK` is LOW. Any latch element can either be used directly as a *Transparent Data Latch with Asynchronous Clear and Present and Gate Enable* (LDCPE) primitive or inferred by the Xilinx Synthesis Tool (XST).

A straightforward way of synthesizing MULLER on Virtex-5 would be to program the LUT for the XOR signal $(a \oplus b)$ and connect its logical inverse to `CLK`.

A.1 C Gate using RS and SR Latches

Although MULLER can be implemented by a single latch and LUT, it is prone to spurious transitions in inputs. Using two latches reduces the risk of an output transition due to glitching. A well known implementation of MULLER using RS and SR latches due to Murphy [TODO:citation] can be used for this purpose.

Both SR and RS latches have two inputs `set` (`S`) and `reset` (`R`). SR latch is equivalent to the RS latch with `R` and `S` inputs interchanged and the output inverted.

$$\texttt{RSLatch}(S,R)' = \overline{R} \cdot (S + \texttt{RSLatch}(S,R)) \quad (2)$$

$$\texttt{SRLatch}(S,R)' = S + \overline{R} \cdot \texttt{SRLatch}(S,R) \quad (3)$$

For brevity $\texttt{MULLER}(a,b)$ is denoted by c. One can easily simplify eqn. 1 (using identity $x + \overline{x} \cdot y = x + y$) to obtain:

$$\begin{aligned}
\texttt{MULLER}(a,b)' &= c' \\
&= (a \cdot b + \overline{a} \cdot \overline{b}) \cdot a + (a \cdot \overline{b} + \overline{a} \cdot b) \cdot c \\
&= a \cdot (b + \overline{b} \cdot c) + \overline{a} \cdot b \cdot c \qquad (4) \\
&= a \cdot c + b \cdot (a + \overline{a} \cdot c) \\
&= a \cdot b + a \cdot c + b \cdot c
\end{aligned}$$

To express MULLER in terms of SR and RS latches, observe that eqn. 4 contains only true values of inputs but `R` appears in both SR and RS latches (eqs. 3 and 2) as $\overline{R}$. Hence, `R` of the latches will be $\overline{a}$ or $\overline{b}$. Only the expression for RS latch (eqn. 2) has a minterm containing both `R` and `S`, which will produce the minterm $a \cdot b$ in eqn. 4.

Therefore, the RS latch appears in the first stage producing the minterms $a \cdot b$ and $a \cdot c$ or $b \cdot c$ (depending on `R` being $\overline{a}$ or $\overline{b}$ respectively). The SR latch forms the second stage. Since `S` appears alone in its expression, we connect the RS output to `S` and the input other than `R` of previous stage to the `R` of the second stage.

Hence, representing the RS output by p and the subsequent SR output by q, we have:

$$\begin{aligned}
\texttt{RSLatch}(S = a, R = \overline{b}) &= p' = a \cdot b + b \cdot p \\
\texttt{SRLatch}(S = p, R = \overline{a}) &= q' = p' + a \cdot q \qquad (5) \\
q' &= a \cdot b + b \cdot p + a \cdot q
\end{aligned}$$