**Layout Design**

Our approach to the layout, excluding the inverter, was to separate the work-load evenly between the three of us as one circuit each. This worked by creating each design as a separate circuit or block and incorporating that block into the top-level design. This system for working on the layout worked well with each person being able to develop their circuit individually and being able to incorporate them into the top-level design quickly and easily. However, when it came to incorporating the last of the circuits into the design, we found that space was becoming tight and some nets had to be moved to allow the *Asynchronous Serial Interface* circuit to added without causing design rule or hierarchical errors. The nets to and from each circuit were routed predominantly around the edge of the chip, although some took a more direct route were practical.

**\*\*Write something about circuit 2 here\*\***

We approached the design of circuits three and four in much the same way. The first level of next state logic is implemented in the same row and feeds the next row which contains the subsequent levels of logic that takes its inputs either predominantly or entirely from the first level.

**\*\*Write something specific about Circuit 3\*\***

In circuit three, the sequential logic is positioned directly above the final level of next state logic so that the outputs can be fed into this with as short nets as possible. Due to the output combinational logic being limited to a single NOR gate, it was positioned on the same row as the sequential logic for proximity, however, this meant the output was situated at the top of the block and a net had to be routed over a longer distance to reach the output pins. To keep things tidy and clearer to understand, the sequential outputs that feed the next state logic are routed down the side of the logic in order from right to left so that one only needs to count Q0, Q0n, Q1, &c. from the end net to find a specific net. Like the output, the input data comes from the top of the block but is required at the bottom so must be routed on a circuitous route.

**\*\*I have written something here, but you may want to delete it and start from scratch\*\***

For circuit four, the register was positioned as near to the output pins as possible to reduce the length of the connections, however, it was only discovered too late, that the register was in the opposite order to the output pins, which means the connections are longer than desired and in some places, very densely packed. The channel of nets running through the middle of the circuit design allow all aspects of the circuit to easily access whichever net is required. Each different block of logic was designed independently as an individual before being added to main design for this circuit. They were designed so that power could be passed directly from one block to another along straight nets making power routing simpler.

There was one error raised by the simulations on the circuit four design. The nature of this regarded the input data not being passed through the retiming metastable circuit because the input data was being taken from the output of this for the next state logic. Subsequently, the *ClockDiv* signal (which drives the metastable flip-flops) was never being set and we were never leaving the default state.

We had little trouble with hierarchical and design rule errors with these relating to net terminations that strayed over the limits of some of the design blocks in the top-level design and circuit four. These were all easily rectified, although finding them, sometimes proved a problem. There was also one hierarchical error caused by the proximity of circuit four to circuit two which was related to the *Q1* net being too close to circuit four and causing the perimeter of the circuit two block to overlap the circuit four block. This was rectified by moving the port for *Q1* in the circuit two block and routing the net in the top-level design rather than in the lower-level block. This created a barrier between the circuit block which overcame the hierarchical error.