# Results

The design showed a significant speed-up over the CPU-based solution. We ran the test with a genome containing 1,000,000,000 base pairs against anywhere from two to 96 tags.

The time and speedup comparison is shown in Figure 1. Because the FPGA has 96 comparison units operating in parellel, it is able to test up to 96 tags simultaneously with no degradation in time. The CPU, however, experienced an average 50% decline in tags/unit time for each doubling of the tag count. The FPGA showed a speedup over the CPU of 10x at the low end to a speedup of 390x at the high end (96 tags).



Figure 1: Measured Time and Speedup

The 10x advantage provided by the 25MHz FPGA over the 2.6GHz CPU was evident even with only two tags tested. Figure 2 highlights this difference. The speedup is due to having two comparison units in parallel and using parallel adders to count the 1's in the xor.



Figure 2: Speedup with Few Tags

Figure 3 shows estimated peformance for tag sizes greater than 96. We did not synthesize an FPGA with 128 comparison units. We feel that with sufficient layout optimization, the Virtex5 has enough capacity to implement 128 comparison units with no degradation in speed. Comparing more than 128 tags requires the genome to be tested multiple times. The speed degradation would be linear; each doubling in tag size would require double the number of data transmissions and take twice the amount of time. The CPU is expected to suffer the same degradation, allowing the FPGA to maintain its advantage.



Figure 3: Detail of comparison driver

The correlation between the number of comparison units and slice utilization is shown in Figure 4. The layout with 64 and 96 comparison units was accomplished only with the aid of SmartXPlorer. Its use of the GlobalOpt Design optimization setting allowed the layout of 96 comparison units to finish in 60 minutes. Notice the geometric increase in slice utilization seen at the higher numbers of comparison units. This might be avoided by constraining a single comparison unit's layout within a block of the FPGA and replicating that layout over the desired number of comparison units. That technique may result in more efficient utilization of LUTs and routing of signals within each block.



Figure 4: FPGA Utilization

The graph in Figure 5 documents the effort and payback of creating the FPGA design versus a software-based implementation. Each metric is normallized to the smaller of the pair. The team that implemented the design is far more experienced at software coding than at FPGA design. Given the development techniques available for each environment, we doubt that any FPGA designer could implement the design as quickly as the software design could be implemented. Even so, the performance improvement and reduction in power consumption validate the initial concept of using the FPGA.



Figure 5: Detail of Comparison Unit