Chapter 5. Results

5.1. Area Results

The area used by each implementation of the design must be scaled to one feature size to be compared. Full custom and standard cell designs were both implemented with a 2.0µ feature size, so this was used as a basis. The sea-of-gates designs were implemented with a 1.6µ feature

	Full Custom (mm^2)	Standard Cell (mm^2)	Sea-of-gates (mm^2)	FPGA (mm^2)
16-bit Adder	0.120	0.386	0.627	30.0
Barrel Shifter	0.103	0.487	1.87	62.6
Stoplight	0.0327	0.139	0.174	15.0
DNA Comparator	0.819	3.01	9.41	113
Life	0.884	2.17	5.07	113
4-bit Multiplier	0.232	0.527	0.983	25.0
8-bit Multiplier	0.847	2.16	7.66	100
Prep1	-	0.614	0.949	20.0
Prep2	-	0.906	2.18	31.3
Prep3	-	0.514	3.01	23.8
Prep4	-	1.22	5.87	82.6
Prep9	-	0.462	1.28	104
Bubble Sort	-	7.43	*	406
8-bit, 8 point IDCT	-	43.4	*	1120
R4000	-	15.9	*	†
Palindrome Detect	-	41.5	*	948
DES Encryption	-	7.81	47.9	406

*These designs were too large for the sea-of-gates router to handle in a reasonable amount of time.
†The R4000 design was too large to fit into a Xilinx 4025 FPGA, the largest supported by our software.

Table 2. Circuit sizes, scaled to a 2.0µ feature size.

size, so each dimension was multiplied by 2/1.6 = 1.25 before making the area calculation. A Xilinx 4000 series FPGA with 49 CLBs using a 1.2μ feature size requires an area of 4.6mm x 4.8mm [3]. The area per CLB is therefore approximately 450,612μ^2. To calculate the area of our circuits at 2.0μ, this area was scaled by $(2.0/1.2)^2$ = 2.778, and multiplied by the number of CLBs in a circuit's implementation. The scaled implementation sizes for each technology are shown in table 2. The geometric mean of the ratios of an implementation's size to its standard cell size is given in Figure 12. The geometric mean was used because its result is more meaningful when dealing with ratios.

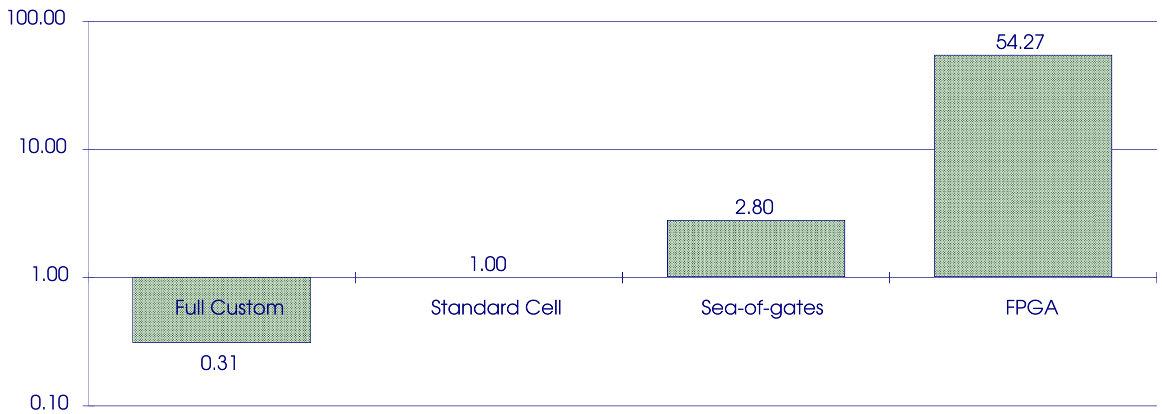

Figure 12. Geometric mean of ratios of implementation size to standard cell size.

Another way to compare area considerations of the different technologies is by examining manufactures' claims of maximum gate capacities of a die in each technology. The definition of a gate is not standard however, so these comparisons are only estimates. The Xilinx 4000 series of gate arrays, currently being fabricated at 0.6μ, support a maximum of 130,000 gates on their largest models, down to a minimum size of 2000 gates on their smallest FPGAs. Gate array companies are announcing up to 600,000 gates using a feature size of 0.6μ. Using the smallest feature sizes, there are up to two million gates available. Standard cells can be packed with densities of up to 700,000 gates at 0.6μ, or over five million at 0.25μ. With the same technology, theses figures show that standard cells designs have at most twice the number of gates as gate arrays, and five times the number of gates as FPGAs. However, is it essential to realize that there are limitations on which technologies are available with specific feature sizes. If an appropriate FPGA doesn't exist with feature sizes less than 0.6μ, the most dense standard cells contain almost forty times the number of gates as the most dense FPGAs.

5.2. Performance

Many factors which affect a circuit's delay are independent of design. Besides wire widths and lengths, which are specified in a design, delay depends on the thickness of materials in the integrated circuit, resistivity, doping, temperature, oxide thickness, permitivity, and carrier mobility. Mobility itself depends on a number of process parameters. These factors can vary from foundry to foundry and among processes with different feature sizes.

The tools used to create each technology's implementations in this study each targeted fabrication processes with different feature sizes. Xilinx tools assume a 0.6µ feature size, the sea-of-gates tool generated designs with a 1.6µ feature size, and the standard cell implementations were created with a 2.0µ feature size. The full custom designs were created using MOSIS scalable design rules, so any feature size could be achieved. To compare the delays involved in the different technologies, a means of comparing implementations in different feature sizes was necessary. This can be done with a basis for converting data in one feature size to another.

Since full custom designs in this study were created using MOSIS scalable design rules, designs created with Magic could be tested in fabrication technologies with different feature sizes. Each full custom design was analyzed at four different feature sizes, 2.0µ, 1.6µ, 1.2µ, and 0.6µ. The results from each circuit were used to determine how a change in size affects a circuit's performance. The ratio of the delay at each feature size to the delay at 2.0µ was determined, as well as the geometric mean of these ratios at each feature size. Dividing the delays from an implementation by the geometric mean for that implementation's feature size gives an approximation of the implementation's delay for a 2.0µ feature size. This process scales results of the performance analyses to one feature size, which allows delay comparisons to be conducted independent of implementation feature size.

The full custom designs were therefore used to generate a basis for converting among feature sizes, by analyzing their delays at four different feature sizes. This is shown in Figure 13a. The designs in this study were optimized for size instead of delay, but this should not affect the ratio of the delays of different feature sizes. DNA comparator's 0.6µ implementation were too large for spice to handle.

The geometric mean of the ratios of a circuit's delay to its delay at 2.0µ is shown in Figure 13b. Theoretically, as the feature size grows, the delay should grow at least linearly. Figure 13b shows that this did not occur. The reason the curve dips between 1.2µ and 1.6µ is likely to be differences in the process parameters. Transistors created at different foundries, at different feature sizes, or even in different batches can have varying behaviors. Foundries test transistors created in their processes and produce parameters that can be used to model them. These

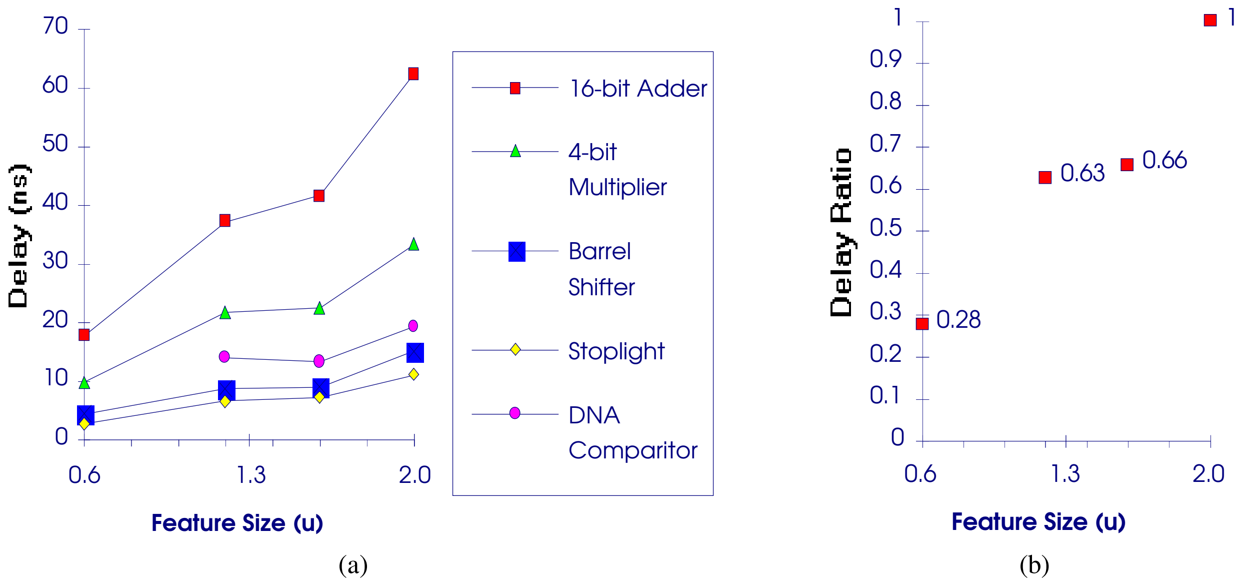

Figure 13. (a) Delay of full custom implementations using four different feature sizes. (b) Geometric mean of ratios of the delay at four feature sizes to the delay at 2.0µ.

parameters therefore come from characteristics of transistors produced at different foundries and differ for different processes. The 2.0µ transistor model parameters used in this study were generated from ORBIT's process data. The 0.6µ and 1.2µ parameters were from Hewlett-Packard's process characteristics, and the 1.6µ parameters were included with the sea-of-gates CAD tool, Ocean. Differences in these parameters most likely account for the strangely shaped curve.

Because of the de facto restrictions on the size of circuits spice can handle in a reasonable amount of time, the simulator supplied with each technology's CAD tool was used instead to analyze each implementation's delays. Unfortunately, the simulators did not find the same delays for the same implementations. To compare two of the simulators, a standard cell implementation of a full adder was created. This implementation was viewed and copied into the full custom editor. Thus now there were two copies of the same implementation which could be simulated with different simulators. The standard cell simulator found the delay from input A to output Sum to be 1.995ns and the delay from A to output Cout to be 1.620ns, while the full custom simulator found the delay from A to Sum to be 2.82ns and the delay from A to Cout to be 1.08ns. Thus for one output the full custom simulator found the delay to be longer, while for the other the delay was found to be shorter.

Use of these different tools and different transistor parameters together led to erratic data. The performance differences among some of the technologies were not expected to be greater than a factor of three, so these differences in the simulators were enough to make our data unreliable. This made it impossible to compare the relative performance of implementations of this study's circuits in different technologies.

A comparison of manufacturers' claims about performance can give an idea of relative performance. Xilinx claims an internal performance of 150MHz for its FPGAs, while some foundries claim up to 650MHz for their gate arrays and standard cell designs.

5.3. Other Issues

The other significant issues when deciding on a technology are cost and time. FPGAs can cost between \$10.50 and \$1500 each, depending on the size, speed grade, and packaging. There is no minimum order, and no NRE payment is required. Once a design has been created, it can be loaded into the FPGA instantly (within milliseconds), so there is essentially no turnaround time.

Gate arrays require a large NRE cost to be paid, as well as a per chip cost. The NRE is generally between \$16k and \$35k, though some companies offset this price by adding to the per chip cost. Each chip will cost between 5 and 8 dollars. Once the design has been completed, it takes time to fabricate it. One company, claiming to have the fastest turn-around time, can return the complete gate array in three days. Generally this takes between two and four weeks.

Standard cell designs (and full custom) require an NRE payment of almost twice that of gate arrays, but the per chip cost is less. The NRE will generally be between \$50k and \$60k, and the per chip cost will be closer to \$2.50. Standard cell turn around times can easily be up to sixteen weeks.

Figure 14 shows the relationship of these prices. The graph is logarithmic in the x-axis so that all of the data will be visible, but each series in the graph is a straight line. Figure 14 shows that for quantities larger than 153, it is cheaper to create gate array designs than FPGA designs. For FPGA quantities over 100, prices range from \$10.50 to \$310. This analysis was done

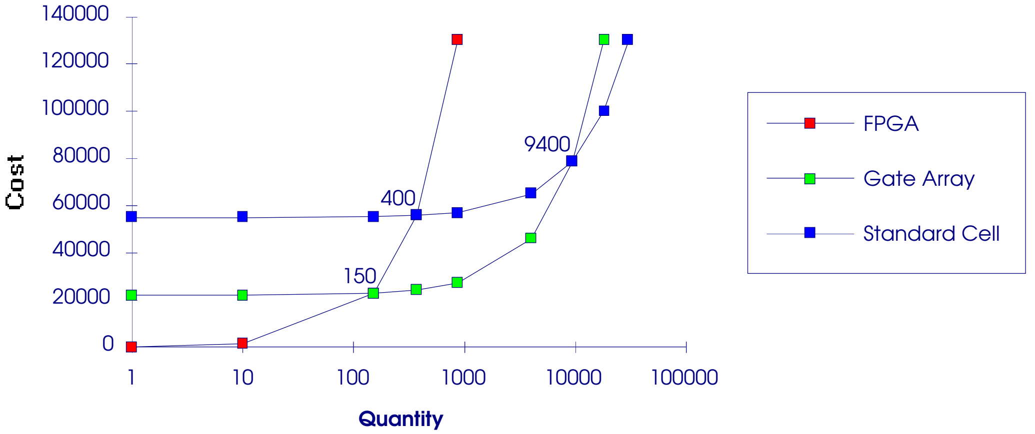

Figure 14. Cost/Quantity relationship for different technologies.

assuming an FPGA cost of $150 per chip, a gate array NRE of $22,000, and a gate array per chip cost of six dollars. As the design size increases, gate arrays will see a price advantage with less units since a design requiring a large (and therefore expensive) FPGA will fit in a more average size (and priced) gate array.

To combat the cost disadvantage at relatively low quantities, Xilinx has recently introduced their HardWire™ Array product line. This is basically a gate array that mimics the structure of the FPGAs. If larger quantities of an ASIC are needed, a designer can switch from an FPGA to a HardWire™ solution without altering the design. Xilinx guarantees functionality equivalent to that of the FPGA design. This solution costs more than the average gate array but it is cheaper than FPGAs for large quantities and saves a designer all the re-design and testing time that would be necessary to switch to a traditional gate array.

Figure 14 also shows the standard cell curve, assuming an NRE of $55,000 and a standard cell unit price of $2.50. Standard cells become advantageous in cost at close to 10,000 units. Again, for larger designs this number will be less, and for smaller designs more units are necessary for a price advantage.