Block ciphers for MARS and RC6
C = A*B mod (2^32)
C = (A^2) mod (2^32)
Topic 1, Project 2
By: Michael Condon

1) Design Criteria
   Arithmetic Operation: C = A*B mod (2^32) and C = (A^2) mod (2^32)
   Used for a 32-bit unsigned multiplier [2].
   Hardware Control Signals: Reset, Clock, Num_Ready (for inputting two, or one, 32-bit number)
   Software Functional Arguments: Either one or two 32-bit numbers read from an input text file and output to a separate text file.

2) Application
   A*B mod (2^32) is known as modular multiplication and (A^2) mod (2^32) is known and modular squaring [2]. Each of these is used in developing encryption techniques for the Advanced Encryption Standard. Modular multiplication is used in the MARS technique for doing keyed forward transformation and keyed backwards transformation; while modular squaring is used in the round function of RC6 [1]. Each of these techniques can be implemented in both hardware and software.

3) Optimization Criteria
   Hardware Optimization: Maximum Throughput, Maximum Latency, Minimum Latency.
   Software Optimization: 8-bit Processor, Minimum execution time.

4) CAD Tools
   Hardware Development Tools: Aldec Active-HDL version 5.1 for Windows
   Software Development Tools: lcc-compiler by Chris Fraser and Dave Hanson (Freeware Compiler).
   Both will be used in a Windows 98 environment using a Pentium II 500MHz processor at home and a Pentium III 1.0GHz processor using Windows 2000 at work.

5) Assumptions
   Hardware: My elementary components to use will be those listed by Mentor for their 0.5 um library. You can find a complete listing of available parts at: http://www.mentor.com/partners/hep/AsicDesignKit/dsheet/ami05databook.html
   The general preliminary assumptions for hand calculations I will make are that all gates have the same delay with the exception of DFFs and Multiplexers. I will assume that their delay is equal to 1.5 times that of a single gate (i.e. an NAND gate). For area I will assume that all gates have the same area except DFF and Multiplexers will have an area of 3 times that of a normal gate. I will use actual

normal gate delays used in the above 0.5 micron Mentor library for the VHDL code [3].
Software: I will be using C to develop the software code. I plan on using the standard library with the standard file opening, reading, and closing operations. I will not know anymore about my software code until I begin to develop it.

6) Testing
Hardware: For testing I will input numbers into the functional VHDL code using a VHDL testbench and compare expected results using those input numbers against actual results received by those numbers. I can approximate circuit speed and area through calculation and determine actual speed through output waveforms generated by the VHDL simulator within Active-HDL. I can't determine actual area because that would require using the actual Mentor library cells and a synthesis tool and completing cell layout and routing to get accurate results.
Software: To test the functional code for software I will do the same. I will calculate the expected results and compare them against actual results received from the inputs. The software development tool I will use (lcc-compiler) has a built in function that will determine execution time and memory use for me. I will use this to determine these numbers.
I plan on using a variety of 32-bit numbers to verify functional and timing values. They are listed below in table 1 for modular multiplication and table 2 for modular squaring. All numbers below are listed in decimal notation for ease of reading.

A	B
4294967296	4294967296
0	0
2536426	533362
697630012	3967219630
98332106	0
1255698304	316845
889630021	763394
523	13645
47423	99

Table 1. Modular Multiplication Test Inputs

A
4294967296
0
25366125
5587632
12
2336988654
365542789
6532
442330

Table 2. Modular Squaring Test Inputs

7) References

1] “Hardware Evaluation of the AES Finalists”; by Tetsuya Ichikawa, Tomomi Kasuya, Mitsuru Matsui; website: http://csrc.nist.gov/encryption/aes/round2/conf3/papers/15-tichikawa.pdf

2] “Lecture 0, Organizational issues. Applications of computer arithmetic algorithms”; by Kris Gaj; website: http://mason.gmu.edu/~kgaj/ECE699/viewgraphs/lecture0_organization_3.pdf

3] “AMI 0.5 Micron Data Book”; Mentor Graphics; website: http://www.mentor.com/partners/hep/AsicDesignKit/dsheet/ami05databook.html