**Topic: 64 bit Fast Adder** A Comparison with and without Multimedia Extension

**Project Specification**

The aim of this project is to analyze and report the relative performance and area characteristics of two different implementations of a 64 bit adder: Adder64 and Adder64-MMX.

Although both implementations are built on the same Carry Select Adder Blocks, the Adder64-MMX features multiple data size operations. It supports a subset of the famous Intel Multimedia Extension (MMX) instructions. Specifically, the MMX enhanced design will support Add operations on four data types: the Byte, Half-Word, Word and Double Word. This additional functionality requires a mode control logic, carry/overflow logic, and a carry/overflow policy. A saturate-on-overflow policy will be used to deal with out-of-bound values.

The functional design of both circuits is presented here including estimated gate counts, critical delay paths and expected area. The relative speed, power consumption and area were key in the choice of a CSA scheme over the Carry Look-ahead, or Carry Skip Adder. Optimum performance will therefore require variable block sizes to compensate for multiplexer path delays.

As hinted above, the 64 bit system will be a modular design made up of simpler 4-bit (Full Adder based) Ripple Carry Adder blocks.
Figure 1 This shows an 8-bit Carry Select Adder with 4 bit RCA blocks.

Carry Select Adder

The carry select adder is a fast adder design that speeds up the calculation of wide bit add operations by computing two sums of stages of the long number and selecting one as the final result once the carry-in bit of the previous stage is known. Note that in Figure 1, the 8-bit adder is divided into 2 4 bit stages: one RCA stage and one CSA stage. The carry-out of the first 4-bit RCA stage is used to select one of the already computed sums. Each sum is computed assuming either one carry-in of 1 or 0. The figure below shows a 16 bit Carry Select Adder with un-equal bit stages.
Depending on the speed of the critical path (carry-select path) the bit width range of the CSA blocks will be adjusted. Currently however, the design assumes a 1 full adder delay for the select logic and accordingly, the bit widths are increased by one full adder each stage. Stage 1: 4 bits, Stage 2: 5 bits and so on. The result is a 10 stage 64-bit adder with a single full adder as the last stage.

**The 2 bit Full Adder**

This is the workhorse of this entire design. The classic full adder design is used to build more elaborate n-bit adders. The figure below shows the Full Adder Logic cell.

**Figure 2 The internal circuit logic and Block symbol**
The 2 by 1 selector cell

By the end of this project this will probably be the second most regularly used cell block. The 2 by 1 selector is the main unit that allows for the CSA functionality. It will as a result be on the critical path and creative techniques will be used to minimize the fan-out, delay and power dissipation. The figure below shows the classic selector while the next figure shows a modified cell trading area for reduced fan-out. An original fan-out of 10 is reduced to 6, the increase in delay is hidden because it is seamlessly overlapped with other computation in parallel.

![Figure 3 The Classic 2 by 1 MUX](image1)

![Figure 4 Fanout aware MUX design](image2)

The 8-bit CSA

Unlike the Adder64, the Adder64-MMX will be designed using equal bit blocks. Each block will be an 8-bit CSA adder block made up of a four-bit RCA feeding
carry to full CSA stage. A mode bit is used to determine whether to propagate the carry out of the 8-bit block or not. This is how multiple data types supported by the architecture will be implemented. The 8-bit CSA cell with mode bits and carry flow control is shown in the figure below.

**Figure 5** 4-bit Adder Block