EE 3061 **Project Assignment 1** **Due: 15 March 2002**

In time for class, tentatively set for Friday, 15 March at 16:00, prepare a Verilog project proposal. (A sample appears at the end of the guidelines.) The proposal should be about one page and should contain the following information:

- Your name and a project title.
- A brief description of what it does.
- A description of the module(s) and their ports. This can be a preliminary description.
- Some ideas on how it would be implemented, not necessarily in Verilog. (For example, a block diagram is okay.)

Projects will be worked on by groups of students. A proposal can be submitted by groups of up to four students.

Projects can include parts of a dynamically scheduled processor (to be discussed in class) or can be unrelated. Some project ideas are listed below, followed by a sample project proposal.

- Pocket Calculator
- Content-Addressable Memory
- Pipelined FP Adder Functional Unit (Addition, subtraction, etc.)
- Wallace Tree multiplier.
- Addressable FIFO
- Set-Associative Cache

Sample Project Proposal

Addressable FIFO

The addressable FIFO stores data in a first-in, first-out manner, but allows an arbitrary element to be changed. The module has the following ports:

```
output full, empty;

output [15:0] head_data_out;
input  pop;

output [3:0] tail_idx;
input  [15:0] tail_data_in;
input  push;

output [15:0] middle_data_out;
input  [15:0] middle_data_in;
input  [3:0] middle_data_idx,
input  middle_data_write;

input  reset, clk;
```

When output full is 1 the fifo is full and cannot accept more data. The data on input `tail_data_in` enters the fifo on the positive edge of the clock when input push is asserted and if output full is zero. Output `tail_idx` is the entry number of the pushed data before the positive edge. (It can be though of as the entry number of the next item to be entered.) When output empty is 1 output `head_data_out` is the data at the head of the fifo. If input pop is 1 at the positive edge the head data item is removed and the next oldest item appears at the output. Output `middle_data_out` is the data associated with the entry number on input `middle_data_idx`. If input `middle_data_write` is 1 at a positive edge that entry is written with input `middle_data_in`. If input reset is 1 on a positive edge the fifo is reset and should be empty.

The fifo will be implemented with an array (memory) with one element per fifo entry. Head and tail pointers will be used to implement the fifo ordering.