Trevor Owen de Clercq
Digital Audio Processing I
E85.2600, Prof. K. Peacock
September 13, 1998

## Academic Computing Facility Report

I activated both my Arts and Media account and internet account on Friday, September 11th at the ACF on the second floor of the Education Building. After my account was activated, I took the opportunity to poke around the facilities themselves. Seating myself at an open computer terminal, I browsed the installed software. The most obviously music related applications were Finale and SoundEdit. I have used Finale before (to notate music), so I ran through a couple examples and found the current version to be much more flexible than its predecessor. As a composer, I am always looking for clearer and less time-consuming ways to disseminate music to prospective players; perhaps this new version more closely approaches the ideal and deserves more research on my part. Concerning my goals in the Music Technology program: since I am interested in the tonmeister sequence, my main thoughts are on how to use the ACF to vary recorded sound. I was hoping to open SoundEdit for a sneak peak, but it was only installed on the MultiMedia computers. Due to the lack of having recorded products on my person, I decided to return at a later date with some recordings and the basic desire to see just how powerful these sound-editing computer facilities are. My ultimate goal is to recombine previously recorded pieces into new compositional models.

## Random Number Sequences

**4-bit system pseudo-code:**

(I'm a little rusty on my programming logic, but hopefully my code makes some sense....)

```
define an array with 4 elements (this is our 4-bit word)
    (the first element in the array equals the MSB; conversely, the last element equals the LSB)
let n equal a counter
at the beginning, all elements in the array should equal zero; n should equal 1.
create a loop (perhaps something like, while n<5); the loop consists of:
    print the elements of the array in order from MSB to LSB (at the beginning it should be 0,0,0,0)
    if the number in array element n equals 0, then switch it to 1
        else the element should equal 0 (switches from 1 back to 0).
    if n=4, then n should switch back to 1
        else n=n+1
    end of loop
```

this little psuedo-program should spit out the sequence of "random" numbers described in the DAP handout from 9/8/98's class.

**8-bit system sequence in base 10:**

0,128,192,224,240,248,252,254,255,127,63,31,15,7,3,1

**8-bit system pseudo-code:**

the 8-bit psuedo-code is identical to the 4-bit psuedo-code, except that an 8 element array is used, the "while" statement in the loop should be "while n<9", and n should flip back to 1 only after it reaches 8.