Problem 3.1

The maximum length sequence is shift register sequences with maximum possible period $2^n - 1$. A shift register will generate a maximum length sequence if and only if its connection polynomial is a special type known as a primitive polynomial.

$$h(D) = 1 + D + D^5 = [1 \ 0 \ 0 \ 0 \ 1] \quad y = [1 \ 0 \ 0 \ 0 \ 0]$$

PN sequence $[1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]$  

Problem 3.2

a.

$$y(t) = 2 \cos(1000\pi t) \quad \text{and} \quad v(t) = 6 \cos(1000\pi t)$$

b. For both $y(t)$ and $v(t)$, we need 1000 Hz as the Nyquist sampling rate

Problem 3.3

(A short list is ok)

(source: FAQ from Berkeley Design Technology Inc.)

The essential difference between a DSP and a microprocessor is that a DSP processor has features designed to support high-performance, repetitive, numerically intensive tasks. In contrast, general-purpose processors or microcontrollers (GPPs/MCUs for short) are either not specialized for a specific kind of applications (in the case of general-purpose processors), or they are designed for control-oriented applications (in the case of microcontrollers). Features that accelerate performance in DSP applications include:

- Single-cycle multiply-accumulate capability; high-performance DSPs often have two multipliers that enable two multiply-accumulate operations per instruction cycle; some DSP have four or more multipliers
- Specialized addressing modes, for example, pre- and post-modification of address pointers, circular addressing, and bit-reversed addressing. Most DSPs provide various configurations of on-chip memory and peripherals tailored for DSP applications. DSPs generally feature multiple-access memory architectures that enable DSPs to complete several accesses to memory in a single instruction cycle
- Specialized execution control. Usually, DSP processors provide a loop instruction that allows tight loops to be repeated without spending any instruction cycles for updating and testing the loop counter or for jumping back to the top of the loop
DSP processors are known for their irregular instruction sets, which generally allow several operations to be encoded in a single instruction. For example, a processor that uses 32-bit instructions may encode two additions, two multiplications, and four 16-bit data moves into a single instruction. In general, DSP processor instruction sets allow a data move to be performed in parallel with an arithmetic operation. GPPs/MCUs, in contrast, usually specify a single operation per instruction.

While the above differences traditionally distinguish DSPs from GPPs/MCUs, in practice it is not important what kind of processor you choose. What is really important is to choose the processor that is best suited for your application; if a GPP/MCU is better suited for your DSP application than a DSP processor, the processor of choice is the GPP/MCU. It is also worth noting that the difference between DSPs and GPPs/MCUs is fading: many GPPs/MCUs now include DSP features, and DSPs are increasingly adding microcontroller features.