Topics for exam 1

Logic circuit interfacing (connecting one logic device to another or to many)

- DC current load calculation
- Noise margin calculation
  - Be able to apply the above two calculations for interfacing various things to a logic gate
- Calculation of a current limiting resistor such as that used with an LED
- What are pull-up or pull-down resistors used for?
- Know typical voltage and current values for inputs and outputs of CMOS gates with Vdd=4.5

Architectural features of the NXP KL25L Processor we are using

Refer to figure 1.17 in the textbook. Also, figure 1 on page 3 of reference [1] on the class web page. These two figures describe the same family of microcontrollers with the list of subsystems arranged differently. Figure 1.17 is more explicit about the architecture because it shows the bus connections between sub-systems. You should be familiar with the architecture shown and also with the list of resources of the various types. For example, you should know that analog subsystems include a 16-bit ADC, 12-bit DAC, etc. (NOTE: There is a typo in figure 1.17. It claims a 16-bit DAC. That should read 16-bit ADC).

You should know the types of memory in our microcontroller and how much of each (page 2, ref [1]).

For the CPU core you should know the registers it has, their word size, and generally how they are used (see chapter 4 of the text)

Configuration details

Know how to configure an I/O pin on the microcontroller for desired operation. For example, to be GPIO output, or GPIO input, or analog input, or SPI, etc. (you don't need to know a specific bit combination for a particular type of I/O, that varies depending on the specific I/O pin). Know the sequence of things that need to be done and registers involved. Don't forget that the clock needs to be turned on to a given peripheral circuit, such as PORTA, before writing to that peripheral's registers.

You should know the names of the registers involved with GPIO operations such as data in register (PDIR or the name used in programming GPIOx_PDIR), data out PDOR, data direction PDDR, and ports used for setting, clearing, or toggling output bits using PSOR, PCOR, or PTOR respectively. Each bit in a port (each port pin) has its own Pin Control Register (PCR) used to set up the mux on that pin

Software concurrency

Be familiar with the concepts and strategy used to create software that responds efficiently to external signals, like a push of a button, while other computation tasks are occurring. Know how to do this with polling and with interrupts.

Know the basics of how interrupts are dealt with by the CPU when they occur.

Chapters 1 through 4 have been assigned as reading and lecture has covered many details thereof. Review your reading.