UCR CS122B, Winter 2003
Advanced Embedded System Design
Prof. Frank Vahid
Final Exam

Name:______________________________ UCR email:______________________________

1. (12 points – 2 points each) Software
   a. Ganssle says "The seduction of the keyboard is the downfall of all too many embedded projects." In one or two short sentences, explain what was meant by that statement:

   b. For a reasonable programmer creating reasonable commercial software, having say 1-2 defects per 1000 lines of code, approximately how many lines of code can we expect the programmer to produce per month? (circle one)
      1 20 300 5,000 10,000 50,000 100,000 10,000,000 1,000,000

   c. The space shuttle software has about 400,000 words, but is perhaps some of the most high quality, bug free code in the world. About how much does that software cost to maintain per year? (circle one)
      $100 $10,000 $10,000,000 $100,000,000,000 $100,000,000,000,000,000

   d. The Software Engineering Institute's CMM (Capability Maturity Model) defines five levels of software maturity. CMM evaluates the maturity of (circle one):
      1. A company's software development processes
      2. Individual programmers
      3. Programming language
      4. Debuggers
      5. Microprocessors

   e. Which of the following is true, according to Ganssle's summary of DeMarco and Lister's 1987 Peopleware report? (circle one)
      1. Creating a quiet work environment can more than double designer productivity
      2. Productivity is greatest when there are frequent interruptions of work (breaks up the monotony)
      3. Productivity is independent of work environment
      4. Bad programmers are more productive than good progammers
      5. Designer productivity is not important and we should stop paying attention to it.

   f. As the size of a software project increases, what happens to productivity (lines of code produced per month)? (circle one)
      1. Productivity increases, due to the economy of scale
      2. Productivity does not change significantly
      3. Productivity decreases, due to increasing software complexity
      4. Productivity spirals, as opposed to falling like a waterfall
      5. Productivity becomes defined as a PID controller

2. (18 points) Real-time and interrupts

a. (6 points) A particular system's main function must repeat a particular task once every 100 ms. The task's execution itself takes 10 ms. The system must also service three peripherals. Peripheral 1 will interrupt every 100 ms, and its ISR requires 20 ms. Peripheral 2 will interrupt every 50 ms, and its ISR requires 10 ms. Peripheral 3 will interrupt every 20 ms. What is the maximum time that peripheral 3's ISR may take? Show your work!

b. (12 points) A system with an interrupt should do the following. Upon being interrupted, the system reads two values x and y using a fast function called RdPkt(&x,&y), which takes 1 ms. Those values x and y should then be used as inputs to some complex function of x and y using a slow function call z = Compute(x,y), which takes 100 ms. The result z should be stored in a queue Q using a fast function call Push(Q, z), which takes 1 ms. On average, interrupts arrive every 500 ms. But, those interrupts could arrive in bursts. The minimum time between interrupts is 50 ms.

1. Can we achieve the goal of not missing any interrupts? Explain why or why not.

2. Write C-like code for the system, being sure to minimize or eliminate the possibility of missed interrupts. Show clearly what goes in the main function and what goes in the ISR. For full credit, your system should ensure that items are pushed onto the queue in the proper order. Hint: you may want to define your own queues (assume a queue object is available to you with Push and Pop functions; don't write your own queue routines).

3. (10 points – 2 points each) Simulation
   a. Describe a key advantage of simulation (e.g., applets, SystemC, or VHDL) over actual physical execution (e.g., on a physical chip embedded in a real environment) in the following contexts.
      1. Before the physical system is built
      2. When the real environment is safety critical
      3. When debugging
   b. Describe a key advantage of physical execution over simulation in the following contexts:
      1. When the amount of test vectors we want to test is huge
      2. When the environment is complex and thus hard to model

4. (10 points – SystemC)
   a. (3 points) Explain the rationale behind the development of SystemC, compared with VHDL.

   b. (7 points) Write a SystemC model of a full adder (whose internals are described behaviorally, not as a connection of gates).

5. Optimization (15 points)

a. (5 points) Consider three implementations of 32-bit adders: a 32-bit ripple-carry adder, a 32-bit carry lookahead adder, and a 32-bit carry lookahead adder built by rippling eight 4-bit carry-lookahead adders. Create a plot, with size on the x axis, and delay on the y axis, with a point plotted for each of the three adder types. Then, describe the design of a fourth adder, and plot its point, such that the fourth adder's point is not a Pareto point.

b. (2 points) Give an informal definition Amdahl's Law:

c. (4 point) A program a.out when executed calls function f1 100 times and f2 10 times. f1 in turn calls function g1 10 times. Assume g1 takes 10 ms per call, f2 takes 10 ms per call, and f1 takes 500 ms per call (which includes g1's time). Suppose a gprof-like profiling tool were used on this program.

1. What would the ordered list of functions look like?

2. What would the call-graph display look like?

d. (4 points) You analyze a program and find that 50% of the program's time is spent in a small loop A, and another 40% in a small loop B.

1. What is the maximum (ideal) speedup of the program you could obtain by partitioning A to an on-chip FPGA?

2. Assume that partitioning A to an FPGA results in A executing only 2 times faster than in software. What is the speedup of the program?

3. With a 2x speedup for A, what is the ideal program speedup you could obtain by then partitioning B to an on-chip FPGA?

6. Low power (10 points)

   a. (4 points) Power consumption in CMOS can be divided into dynamic and static power.

      1. Define in a short sentence (not an equation) what is meant by dynamic power:

      2. Define what is meant by static power:

      3. Dynamic power of a CMOS net is proportional to C*f*_____ (fill in the blank), where C correpsonds to the capacitance, and f is the switching frequency of the net. The item you filled in the blank corresponds to ____.

   b. (2 points) Starting from the fact that electricity costs about $0.20 per kilowatt-hour, estimate the cost of leaving a personal computer on for a full month, assuming the PC consumes 120 Watts. Show your work.

   c. (4 points) Say a particular battery can store 50,000 Joules. You want to run a microprocessor from this battery for one year. The microprocessor runs a periodic task that requires the processor to be active only 10 ms for every 1 second. When idle, the microprocessor consumes only 0.01 mW.

      1. What must the power consumption of the microprocessor be less than when the microprocessor is active, to achieve one year of operation?

      2. Is this power consumption a reasonable value for an 8-bit microcontroller (i.e., could you expect to find an off-the-shelf microprocessor with such power consumption?). Explain.

7. Control (15 points)
   a. (2 points) Define a control system, and give two examples of control systems, and two of embedded systems that are not control systems.
   b. (3 points) Draw the diagram of a closed-loop control system, with disturbance.
   c. (4 points) Write the equation for a PID controller, and describe the intuition/purpose of each of the P, I and D terms.
   d. (4 points) Write C code for a PD controller (note: no I).
   e. (2 points) We discussed P, PD, PI, and PID controllers. Explain whether a DI controller makes sense.

8. Testing and safety (10 points)
    a. What is the difference between black-box and white-box testing?

    b. What is a regression test suite?

    c. What is statement coverage?

    d. What statement coverage (percentage) should we reasonably expect to achieve during testing?

    e. What is the relative percentage of time one should expect to spend on design versus testing.

    f. Explain the difference between forward and backward safety analysis.

    g. (3 points) We discussed Leveson's view that most improvement to safety could be made by improving the safety culture among designers. Explain in paragraph what this means. Include an example.