EEL375
Major (2009)
Total: 30 marks Time: 2 hrs
Open Notes

Q1. Design an underwater ultra-sonic ranging system. The distance to the object can vary from 1 to 100m. The ultrasonic transducer will send a short 5 micro second pulse into the water in the direction of interest. The sound wave travels at 1500 m/s and reflect off the object it runs into. The reflected wave also travels at 1500 m/s. back to the transducer. The reflected pulse is sensed by the same transducer. Your system will trigger the electronics (give a 5 micro second pulse), measure the time of flight and calculate distance to the object. Provide complete hardware and software design of the system. (6)

Q.2 (a) Optimise the following loop, if possible:

```
for(i=0; i<n; i++)
        for (j=0; j <M; j++)
                z[i][j]=a[i]*b[i][j]
```

Q.2 (b) Show that data layout can affect execution time of a code using register indirect addressing. (3+1)

Q. 3 Suggest design of a digital camera system (with hardware block diagram and features of system software). Will you use only a DSP or a combination of GPP and DSP? (4)

Q.4 Deadlock is the situation in which multiple concurrent threads/processes are permanently blocked because of resource requirements that can never be satisfied. Prove that with priority ceiling protocol deadlock can never occur. (3)

Q,5 Discuss how register file organization can effect performance of VLIW processors. (3)

Q.6 When two local area networks are to be connected, an embedded system, called a bridge is inserted between them, connected to both. Design a mechanism for the bridge to handle packets transmitted on either network to determine whether the packets are to be forwarded and to forward the packets if needed. If the packet size is 1 K bytes can you derive an expression, in terms of system parameters, for the maximum data rate on each of the networks that can be tolerated without having the bridge loose packets? (5)

Q.7 (i). When will you prefer scratch pad memory over cache memory for embedded system design?

Q.7(ii) In a virtual memory system, the execution of an instruction may be interrupted by a page fault. How does a new page be fetched when such a page fault occurs? What state information has to be saved so that this instruction can be resumed later? How will this situation be handled in an ARM9 based embedded system? (2+3)