ECE 576 – Engineering of Computer Based Systems – Exam Practice Problems

1. Which of the following are true about tightly-coupled coprocessor architectures? *(Circle all that apply.)*

   (a) Lower communication requirements compared to instruction set extensible processor architecture.
   (b) Processor and coprocessor can communicate through shared memory.
   (c) Memory interface can limit coprocessor parallelism and performance.
   (d) Lower communication requirements compared to loosely-coupled coprocessor architecture.
   (e) None of the above.

2. In order to speedup a software application by 5X, of the following, which is the minimum portion of the application that needs to be partitioned to hardware if the partitioned hardware is 15X faster than software?

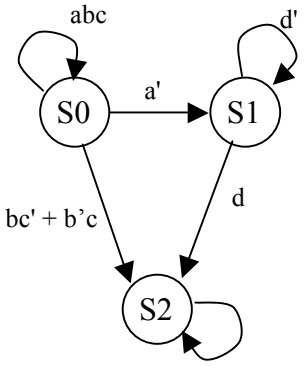
3. Very briefly *(i.e. in the space provided)* define and summarize the importance of any two design metrics of embedded systems and discuss the tradeoffs between these metrics, if any tradeoffs exist.

4. Very briefly *(i.e. in the space provided)* define and summarize the difference between a loosely-coupled and tightly-coupled coprocessor architecture.

5. In which of the following transaction level modeling abstractions would you expect cycle accurate communication model? *Choose all that apply.*

   (a) Specification Model
   (b) Bus-arbitration Model
   (c) Bus-functional Models
   (d) Implementation Model

6. Identify and describe why the following FSM is incorrectly specified.



7. Design a state machine for a digital parking meter system. The parking meter only accepts quarters, and for each quarter deposited, the parking meter will add 15 minutes to the remaining time up to a maximum of 2 hours. The current remaining time in minutes will be displayed on a digital display. However, if the time remaining is less than 1 minute, the parking meter with display the remaining time in seconds while flashing the display. When the time has expired, the display will flash an expired message continually until another coin is deposited.

   Note: This problem is open-ended and you will need to define a set of appropriate inputs and outputs for the system.

8. Partition the following C code to a loosely-coupled coprocessor design.

```
int main() { // Total Cycles: 18086030

   short* A; // A is an array of short integers (16 bits)


   // Somewhere deep within the application surrounded by a shroud of mystery
   // and intrigue lies the following kernel
   for(i=0; i<SIZE; i++) {    // Total Cycles: 13736696, Execs: 60000, Average Iters: 28.6
       if( (lVal < A[i+1]) && (lVal >= A[i]) ) {
           break;
       }
   }
   result = i;

   return 0;
}
```

(a) (5 points) Partition the identified loop to hardware by creating a high-level state machine. ***Note: You do not need to implement communication between processor and coprocessor or interface with memory for array accesses at this time.***

(b) (3 points) Assuming only a single 32-bit memory read-write port is available, draw the architecture for the loosely-coupled coprocessor system labeling the various bus signals.

(c) (6 points) Modify your high-level state machine to communicate with the processor and interface directly with memory assuming memory reads and memory writes require one cycle. ***Note: Be sure to clearly annotate any memory mapped registers required for communication.***

(d) (3 points) Estimate the speedup of the partitioned design over software only execution assuming the hardware coprocessor executes at the same operating frequency of the processor.

(e) (3 points) Estimate the speedup of the partitioned design over software only execution assuming the hardware coprocessor executes at one half the operating frequency of the processor.