ELEC 464 : MICROCOMPUTER SYSTEM DESIGN
FALL 1995

Mid-Term Exam Review - Part 1

The following questions are meant as a review of the material in the first part of the course. They are typical of the types of problems that you would be required to solve when designing a microprocessor system.

1. Memory System Design

Draw a schematic showing how four 8k x 8 ROMs would be connected to a 68000 microprocessor. The 32k of ROM should start at memory location $100000.

Each ROM has the following pins: chip select (CS*), output enable (OE*), data outputs (D0 to D7) and address inputs (A0 to A12).

Show how the 68000's address (A01 to A23) and data buses (D00 to D15), and AS*, UDS* and LDS* would be connected to the ROMs (assume other 68000 signals such as DTACK* are taken care of by other circuitry). Design the glue logic (including address decoders) using simple logic gates. In the case of address and data buses indicate which 68000 address/data lines are connected to which ROM address/data lines.

2. Glue Logic Design

Design a PAL-based circuit that generates a 2-clock-period-long pulse (WAIT*) when a signal (IO*) indicates access to a slow peripheral and a 1-clock period pulse when access is to a faster device (ROM*). The inputs to the device (in addition to the clock) are IO* and ROM*. The output is WAIT*. The WAIT* pulse begins on the first rising edge of the clock following either IO* or ROM* and lasts for the appropriate duration regardless of what IO* or ROM* do during the WAIT* pulse. Only one of IO* or ROM* will be active at any time and they will go inactive for at least two clock cycles after WAIT* is de-asserted.

Base your design on a state machine. Show the truth tables for any combinational circuitry. Give the CUPL Pin declarations and logic equations required to implement such a circuit.

3. Exception Handling/68000 Assembler

Write an exception handler in 68000 assembly language for a time-of-day interrupt. Hardware on the system will cause the exception handler to execute every millisecond. Its only purpose it to increment a time variable that is available to all other tasks running on the machine. The count is a 32-bit value kept at memory location $430–$433.

The execution of your exception handler should be transparent to whatever was running when the interrupt occurred (i.e. you should not modify any registers or memory locations except the supervisor stack). You may assume the supervisor stack has been set up and it has at least 16 bytes available for your exception handler to use. You may also assume the count value was initialized when the system was rebooted.