ELEC 464 : MICROCOMPUTER SYSTEM DESIGN
FALL 1995

Mid-Term Exam Review - Solutions

Question 1

Since the 68000 uses a 16-bit data bus the four 8k x 8 ROMS will be arranged as two blocks of two (two blocks of 16 k bytes). The first set will appear at $10 0000 and the second at $10 4000. The address patterns that the two banks of devices will respond to are:

`0001 0000 00XX XXXX XXXX XXXX`

and

`0001 0000 01XX XXXX XXXX XXXX`.

The chip select (CS*) pin will be enabled when: 1) a valid address is detected (including AS*) and 2) the appropriate data strobe is asserted.

Since for most EPROMs the output is only enabled when both CS* and OE* are asserted, the OE* can be permanently enabled, or connected to CS*.

The EPROM data outputs are connected to the CPU data lines. The EPROM chip data outputs corresponding to the MS byte is enabled by UDS* and connected to D08 to D15. The EPROM chip data outputs corresponding to the LS byte is enabled by LDS* and connected to D00 to D07.

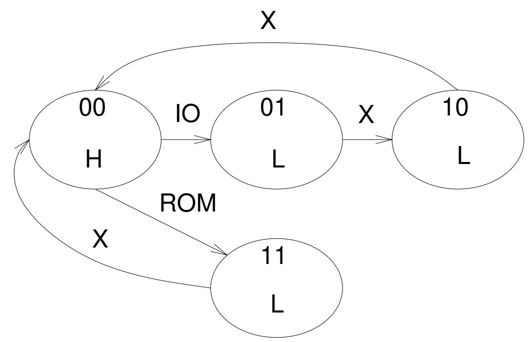
The EPROM address lines A00 to A12 will be connected to the CPU address lines A01 to A13 respectively. The value of the (implied) "A0" is used to select one chip or the other.

Question 2

This design can be implemented with a 4-state state machine.

The two state variables are A and B. The inputs are IO* and ROM*. The output is WAIT*.

The state diagram below shows the states, their encodings, the state transition conditions and the value of the output (WAIT*) in each state.



The truth table showing the output and next state for each state is given below.

Inputs				Outputs		
A	B	IO*	ROM*	WAIT*	A'	B'
0	0	1	0	0	0	1
0	0	0	1	0	1	1
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	X	X	1	1	0
1	0	X	X	1	0	0
1	1	X	X	1	0	0

Listings showing the CUPL code and simulation output are given in a separate listing.

Question 3

The exception handler can be as simple as:

```
ADDQ.L   #1,$430    ; increment time count
RTE
```

A solution involving the use of a data register would be as follows:

```
MOVE.L  D0,-(A7)   ; save D0
MOVE.L  $430,D0    ; get current count
ADD.L   #1,D0      ; increment
MOVE.L  D0,$430    ; save new count
MOVE.L  (A7)+,D0   ; restore D0
RTE
```