Practice Questions Solution ECE441

1. What is meant by the terms partial address decoding and full address decoding?

Full decoding:

Device appears at one address, all address lines are used in t he decoding logic.

Partial decoding:

Device appears at multiple addresses; all address lines are NOT used. The address lines NOT used are "DON'T CARE" in the decoding logic, which causes multiple addresses.

2. What is a double bus fault?

Trying to access non-existent memory gives no DTACK which results in watchdog firing which give bus error. After bus error exception, we go to supervisor mode. If we get another bus error. This results in a double bus fault.

3. Describe the importance of status and control registers in a programmable input/output device.

Control register configures or tailors device to your specific needs such as number of outputs/inputs, parity, handshake etc.

Status register latches or holds a condition so the microprocessor can read what occurred such as receive buffer full, interrupt occurred, what input caused interrupt.

4. Carefully describe the order of the timing events involved in a synchronous read bus cycle.

1) address lines and data line set up (if write operation)
2) AS*,UDS*,LDS* are asserted by microprocessor
3) VPA* is generated by decode logic
4) VMA* is asserted by microprocessor
5) E clock becomes synchronized
6) Data lines valid(if read operation)

5. A PIA must be initialized for the following conditions:

PA0-PA2, PA6-PA7:	Output
PA3-PA5:	Input
PB0-PB6:	Output
PB7:	Input

CA1 is active for positive transition and does not generate an interrupt.
CA2 is active for positive transition and is enabled to generate an interrupt.
CB1 is active for positive transition and is enabled to generate an interrupt.
CB2 is operating in pulse mode.

Determine the content of the following registers after reset and the above initialization:

DDRA $ C7 11000111
DDRB $ 7F 01111111
CRA $ 1E 00011110
CRB $ 2F 00101111

6. Design a memory and input/output board consisting of the following components:
   ROM $10000 - $1FFFF, Use 32K x 8 ROM Chips
   Output Port, One Byte, Location $20000, Use 74373 Chip
   One PIA, to be accessed using D0-D7 data lines of the microprocessor, with a starting address at $20001, Assume the interrupt level 7 is used for both the IRQA* and IRQB* lines.

   a) You must show the address map below for your design.

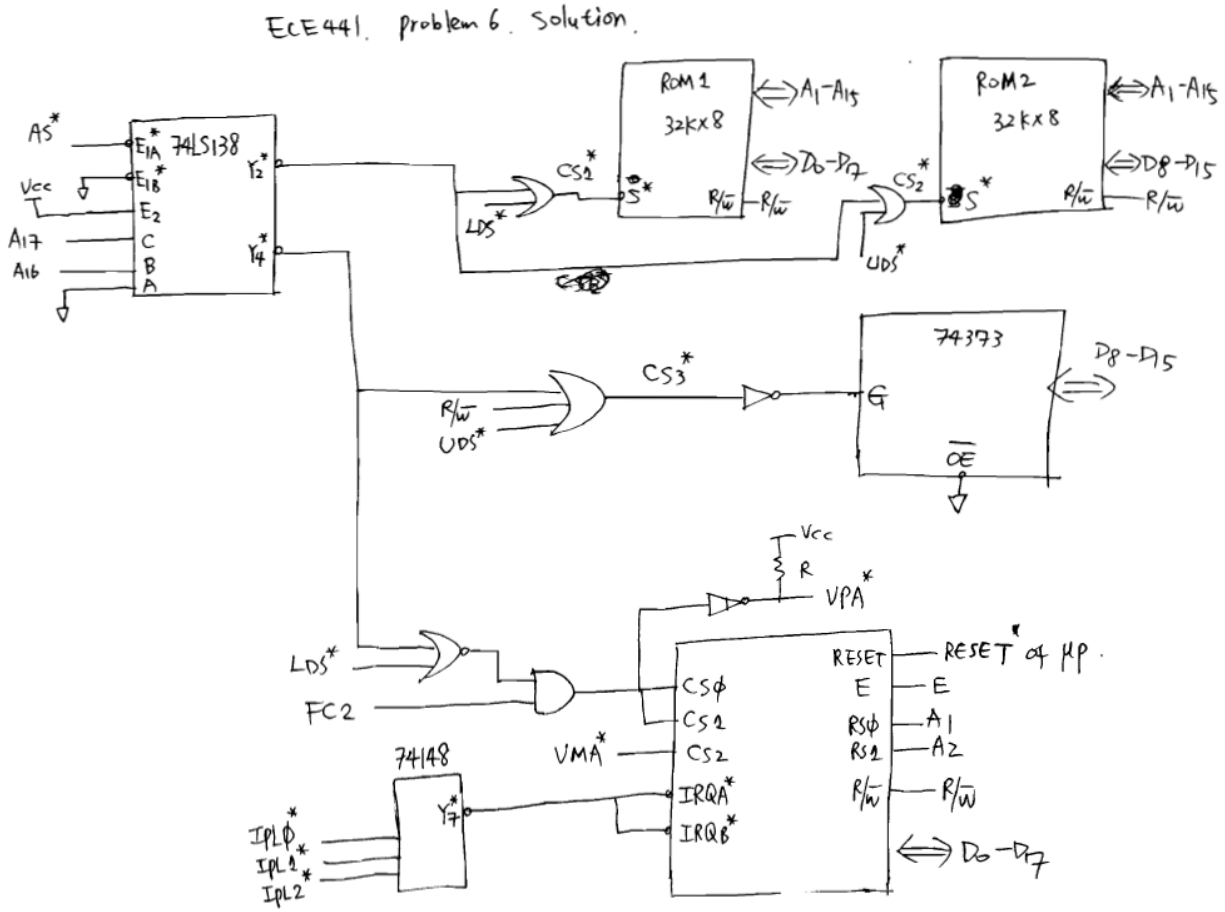
	A23 … A18	A17	A16	A15…A3	A2	A1	UDS*	LDS*
ROM1	0 … 0	0	1	x … x	x	x	0	d
ROM2	0 … 0	0	1	x … x	x	x	d	0
OUTPUT	0 … 0	1	0	0 … 0	0	0	0	d
PIA	0 … 0	1	0	0 … 0	x	x	d	0
OTHERS	0 … 0	0	0	x … x	x	x	d	d

   b) Find (include value 0 or 1) and list the required signals (control, minimum number of address lines, etc.) for decoding and selecting ROM,. Output port and PIA. Note that there are other devices using memory in the range of $000000 - $00FFFF.

      1) Use partial decoding. Assume no other device in system above $20008; otherwise, A23-A18 would be used in decoding.

      2) All the required signals needed in the design are listed as following
      ROM:
      A17 = 0, A16 = 1, UDS*, LDS*, AS*, R/W*, DTACK*, D0 – D15
      Output:
      A17 = 1, A16 = 0, UDS* = 0, AS*, R/W* = 0, DTACK*, D8 –D15
      PIA:
      A17 = 1, A16 = 0, LDS* = 0, AS*, R/W* = 0, VPA*, VMA*, D0 – D7, E, RESET, FC2 = 1

   c) Show your hardware design for accessing ROM, Output Port and PIA. Assume ROM is faster than the microprocessor and no delay data acknowledge is required. Mark all required signals and use any TTL logic that you think is suitable for your design.



7. How many bytes will the following program move from one region of memory to another region of memory? Explain your answer. Note: Register A1 = $A100 and A0 = $A000. ORG is $1000 (Be careful answering this question).

```
        ORG $1000                       * function: block move
        MOVE.W #$1000, D0               * set block length to D0
REP     MOVE.W 0(A0, D0), 0(A1, D0)     * move the word from 0(A0, D0) to 0(A1, D0)
        DBRA D0, REP                    * decrease the block length and keep moving data
```

Totally 2 bytes will be move from $B000-$B001 to $B100-$B101. The only problem of this program is that in the 2nd run of block move, D0 will be changed to $0FFF. In this case, the move word operation on odd memory location 0(A0, D0) and 0(A1, D0) will be illegal. An address error will be generated.