• Open book and open notes, **90-minute** examination to be done in pencil.
• No electronic devices are permitted.
• All work and solutions are to be written on the exam where appropriate.

---

### Point System (for instructor and TA use only)

<table>
<thead>
<tr>
<th>Page</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page 2</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Page 3</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Page 4</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Page 5</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Page 6</td>
<td>18</td>
<td></td>
</tr>
</tbody>
</table>

**TOTAL**: ____________ out of 101

---

Grade Review Information:  *(NOTE: deadline of request for grade review is the day the exam is returned.)*

________________________________________________________________________________________
________________________________________________________________________________________
________________________________________________________________________________________
________________________________________________________________________________________
________________________________________________________________________________________
________________________________________________________________________________________
________________________________________________________________________________________
________________________________________________________________________________________
1. A student would like to use a microprocessor with the following characteristics:

   I. A 16 bit Address Bus, a 16 bit Data Bus and Control Signals R/-W & -Data_Strobe (-DS).
   II. Upon reset, the processor loads the PC with address 0x2000 to begin fetching the first instruction.

Given the following devices (and as many as you need of each) 32Kx8 EPROMs, 8Kx8 SRAMs, 74HC273s, 74HC244s and any combinatorial logic you need, show the required circuitry (both decode circuit & memory) to place 4Kx16 of EPROM at the appropriate place in the memory map. Label all signals and assume full address decoding. (7 pt.)

2. Next, show the decode circuit and memory required to place 8Kx16 SRAM starting at 0x5000. Label all signals and assume full address decoding. (7 pt.)
3. Finally, show the decode circuit and devices required to create an input/output port at 0x8000 assuming 512 continuous images. Label all signals. (9 pt.)

4. What is the address range that corresponds to the 512 I/O port images? ____________________________ (2 pt.)

5. A student has wired up 8 low true switches and 8 low true LEDs on their board as shown below:

Switches S8:5 (most significant nibble) => Inputs GPIO27:24, Switches S4:1 (least significant nibble) => Inputs GPIO3:0
LEDs L8:1 (where L8 is the MSB) => Outputs GPIO15:8

Write the DSP assembly code to continuously echo out the switches to the LEDs. Assume the GPIO direction and mux registers have already been initialized. Also, when you write to the LEDs do not modify any unused I/O such as GPIO31:28, GPIO23:16 and GPIO7:4. (12 pt.)

GPADAT .set 0x6FC0 ________________
6. For the following code, show the values of the flags and registers after the compare and add instructions.

```
MOV AL, #0xCCCC
MOV AH, #0x4ABC
CMP AL, AH ; after this instruction executes show the contents of Z, C, V and N below. (3 pt.)

Z = _____  C = _____  V = _____  N = ______

ADD AL, AH ; after this instruction executes show the contents of AL, C, V and N below. (5 pt.)

AL = __________ Hex  C = ______  V = ______  N = ______
```

7. Write an assembly subroutine to sum N 16 bit signed numbers for a given signed vector. N will be passed to the routine via AL and the starting address will be passed via AH. i.e. AL = N and AH = starting vector address. Pass back the sum in AL and set AH = 0 if no sum falls out of the 2’s complement 16 bit range otherwise set AH = -1 if any sum results in a value out of the 2’s complement range. Use AR0 as a pointer to the vector and AR1 as your counter. The best code will result in the highest points. (12 pt.)
For the next several problems, consult the attached program and generated LIST file and answer the following questions.

8. What is the effective address for the instruction MOV AH,0x4ABC (line 13)? _________________ Hex (2 pt.)

9. At run/execution time, what is the value of the operand in MOV AR0,#value1 (line 17)? _______________ Hex (2 pt.)

10. What is the effective address for the instruction MOV *AR0,AL at run time (line 20)? _________________ Hex (2 pt.)

11. What value is the Stack Pointer being initialized to? ________________________________ Hex (2 pt.)

12. What is the value of the SP & stack memory contents upon execution of MOV AH,#0x0 (line #36)? (6 pt.)

\[ SP = \underline{\text{Hex}} \quad \text{Stack Memory Address & Contents Below} \]

<table>
<thead>
<tr>
<th>Addr (Hex)</th>
<th>Data (Hex)</th>
</tr>
</thead>
</table>

13. What is the value of the SP after the POP AR0 (line 53 is executed)? _________________ Hex (2 pt.)

14. At load time, what addresses in DSP memory will be modified? (2 pt.)

15. At run time, what addresses in DSP memory will be modified? (2 pt.)

16. What is the final computed result stored in value3 after MOV *AR0,AL (line 27) is executed? ___________ Hex (2 pt.)

17. What does SUB1 compute? i.e. It approximately computes ... (2 pt.)

18. What is the largest unsigned input value we can pass via AL to SUB1 before that will not result in a 16 bit unsigned overflow result? i.e. The answer will still fit in 16 bits. (2 pt.)
19. Assuming that the DSP has an internal 16 bit data bus and 22 bit address bus, fill out the following cycle execution table for execution of the two instructions listed below. **Write all values in Hex.** (12 pt.)

<table>
<thead>
<tr>
<th>Cycle#</th>
<th>R/W</th>
<th>PC (Hex)</th>
<th>A15:0 (Hex)</th>
<th>D15:0 (Hex)</th>
<th>Reg Driving the Addr Bus</th>
<th>Device Driving the Data Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

20. A student has connected a simple as shown in Fig. 1 found in the attached ASM file program. Assume that the GPIO pins have been already set up for input and output. Write the code to detect if just a 2, 5 or 8 key is pressed and then write the key number pressed 2, 5, 8 or zero (if no key is pressed) to an external latch placed at 0x4000. (6 pt.)

```asm
GPADAT .set 0x6FC0
```

---

Page 6
.global _c_int00  
.data  ;section starts at 0xA000 in Linker Command File
.value1 .word 0x4567  
.value2 .word 0x0  
.value3 .word 8  
.text  ;section starts at 0x9000 in Linker Command File
.c_int00:  
28A9  MOV  AL,#0xCCCC ;used to test flags for problem #6 in Exam I F11  
00000001 CCCC  
28AB  MOV  AH,#0x4ABC  
00000003 4ABC  
54A8  CMP  AL,AH  
94A8  ADD  AL,AH  
28A0" MOV  AR0,#value1 ;simple prog. to move data around for questions  
00000007 0000  
28AC  MOV  AL,*AR0  
00000012 0080' LC   SUB1  ;subroutine I/O passed through AL  
00000013 0016  
96C0  MOV  *AR0,AL  
28C0  MOV  *AR0,#0x6789  
6789  
28AD  MOV  SP,#data_section+0x80  
000000e A080  
28A0" MOV  AR0,#value3  
000010 0002  
92C0  MOV  AL,*AR0  
000012 0080' LC   SUB1  ;subroutine I/O passed through AL  
000013 0016  
96C0  MOV  *AR0,AL  
6F00  END1: B END1,UNC  
22A0  PUSH  AR0  ;protect registers modified by subroutine  
22A8  PUSH  AH  
9B00  MOV  AH,#0x0  
96A0  MOV  AR0,AL  
FPC2  LSR  AL,3  
95A9  ADD  AH,AL  
92A0  MOV  AL,AR0  
FPC5  LSR  AL,6  
95A9  ADD  AH,AL  
92A0  MOV  AL,AR0  
95A9  ADD  AH,AL  
95A9  ADD  AH,AL  
92A8  MOV  AL,AH  
2AA8  POP  AH  
2AA0  POP  AR0  
7614  LRET
.global _c_int00
.data_section .set 0xA000
.data
;section starts at 0xA000 in Linker Command File
value1 .word 0x4567
value2 .word 0x0
value3 .word 8
.text
;section starts at 0x9000 in Linker Command File

_c_int00:
    MOV AL,#0xCCCC
    ;used to test flags for problem #6 in Exam I Fall
    MOV AH,#0x4ABC
    CMP AL,AH
    ADD AL,AH

    MOV AR0,#value1
    ;simple program to move data around for questions
    MOV AL,*AR0
    INC AR0
    MOV *AR0,AL
    MOV *AR0,#0x6789

    MOV SP,#data_section+0x80
    MOV AR0,#value3
    MOV AL,*AR0

    LC SUB1
    ;subroutine I/O passed through AL
    MOV *AR0,AL

    END1:
    B END1,UNC

SUB1:
;AL has input value & passes back output result
;Helpful Info: 1/8 = 0.125, 1/16 = 0.0625, 1/32 = 0.03125, 1/64 = 0.015625
    PUSH AR0
    ;protect registers modified by subroutine
    PUSH AH
    MOV AH,#0x0
    MOV AR0,AL

    LSR AL,3
    ADD AH,AL

    MOV AL,AR0
    LSR AL,6
    ADD AH,AL

    MOV AL,AR0
    ADD AH,AL
    ADD AH,AL
    ADD AH,AL
    MOV AL,AH

    POP AH
    POP AR0
    LRET

Figure 1. Keypad for Problem #20