1. (20 points) Intel Processor Architecture

a. You see the following signals on the PC processor busses during an instruction execute cycle:

<table>
<thead>
<tr>
<th>Signal</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address Bus</td>
<td>0x3f8</td>
</tr>
<tr>
<td>Data Bus</td>
<td>0x20</td>
</tr>
<tr>
<td>M/IO#</td>
<td>0</td>
</tr>
<tr>
<td>W/R#</td>
<td>1</td>
</tr>
</tbody>
</table>

Show in gas syntax the instruction that is being executed, instructions needed to setup the processor for this instruction to be executed, and explain in functional terms what is happening.

b. In your own words, explain the advantages that a processor with more signals in its address bus and data bus has over another processor with fewer signals on those busses.

2. (10 points) Tutor

a. Show the Tutor command for setting a breakpoint at the usual start address for your SAPC program.

b. When you hit a Tutor breakpoint while debugging your code, show the commands (and any interpretation required) to display the contents of the 32 locations on the top of the stack.
3. (20 points) Assembly Language
   a. Memory at address 0x00100250 contains the following 16 bytes of data:
      00100250 58 02 10 00 01 23 45 67 89 ab cd ef 00 01 10 00

      Show the contents of the %eax register after execution of these instructions:
      movl 0x00100250, %ecx
      movl 4(%ecx), %eax

      %eax ________________

   b. Explain what is incorrect about this gas instruction:

      popl $0x3fffc0

   c. Show the hex value of the eax register and state of the specified condition flags after executing these instructions:

      movl $0x202000, %eax
      subw $0x4000, %ax

      %eax ________________ CF = ____ SF = ____ ZF = ____ OF = ____

4. (20 points) RS-232 Interface

   a. Draw a trace of the voltage on the transmit data signal for transmitting an ASCII 'Q'. Label all bits appropriately, e.g. start, stop, MSB, LSB, etc.

      One bit time
      +12 V

      -12 V

   b. In your own words, explain the meaning of receiver overrun in the UART and its consequences. As a programmer, what do you need to do to avoid it?
5. (30 points)
Write a gas assembly language version of the C library function `isdigit()`. As specified in the C library, your `isdigit` function should be a C callable function with the function prototype:

   int isdigit(int c);

It should compare the argument `c` to the range of ASCII decimal digits (0-9) and provide a return value according to the following rules:

   If `c` is an ASCII decimal digit, return a non-zero value
   If `c` is not an ASCII decimal digit, return a zero value

Note: A character is stored in an integer in the least significant byte.
EXAM #1 SOLUTIONS:

1. Intel Processor Architecture
   a. Instruction accesses an I/O Device because M/IO# = 0. It must be the COM1 data port because the address = 0x3f8. It is a write because W/R# = 1 and there is an ASCII space character ‘ ’ is on the data bus. Therefore:

   \[
   \begin{align*}
   \text{movw} & \ 0x3f8, \ %dx \quad \leftarrow \ \text{Set up instructions} \\
   \text{movb} & \ '$', \ %al \\
   \text{outb} & \ %al, \ (%dx) \quad \leftarrow \ \text{This is the instruction!}
   \end{align*}
   \]

   It is outputting an ASCII space character (‘ ’) to the COM1 port.

   b. A processor with more address signals can support a larger memory space and can run programs that need more memory. A processor with more data bus signals can move data to/from memory or I/O devices faster and will usually execute any given program faster.

2. Tutor
   a. b 100100
   b. Use \text{rd} to display the registers and find the value of the \%esp. Use \text{md} (value in \%esp) 20

3. Assembly Language
   a. 00100250 58 02 10 00 01 23 45 67 89 ab cd ef 00 01 10 00
      \vdots
      To \%ecx
      \vdots
      To \%eax

      The value moved from memory at 0x00100250 to the \%ecx is 0x00100258.
      The value moved from memory at 4 + 0x00100258 to the \%eax is 0x00100100

   b. The destination for the popl is shown as immediate data. You can’t specify immediate data as the destination for a pop instruction.

   c. Note that subtract operation is done only on lower half of \%eax

   \%eax 0x20e0000 \ CF = 1 \ SF = 1 \ ZF = 0 \ OF = 0

4. RS-232 Interface
   ASCII ‘Q’ = 0x51 (from ASCII Code Chart)

   \[ \begin{array}{cccccc}
   \text{One bit time} & \text{LSB} & \text{MSB} & \text{Stop Bit} = 1 \\
   \text{+12 V} & 1 & 0 & 0 & 0 & 1 \\
   \text{-12 V} & \text{Start Bit} = 0 \\
   \end{array} \]

   b. An overrun occurs when the software is late in checking the DR bit and reading the data in the receive holding register. The next character is written over the previous character which is lost. The code must always get around to reading the DR bit and the data within a character time.
5. YOU ONLY NEEDED TO WRITE THE isdigit.s FUNCTION!

u18(25)% cat isdigitc.c
/* isdigitc.c: c test code for assy isdigit function
bob wilson
03/07/2003 */

#include <stdio.h>

int main(void)
{
    printf("Character / is %s a digit.\n", (isdigit(0x2f))? "" : "not");
    printf("Character 0 is %s a digit.\n", (isdigit('0'))? "" : "not");
    printf("Character 9 is %s a digit.\n", (isdigit('9'))? "" : "not");
    printf("Character : is %s a digit.\n", (isdigit(':'))? "" : "not");
    return 0;
}

u18(26)% cat isdigit.s
# isdigit.s: intel assy version of library function isdigit()

.text
.globl _isdigit

_isdigit:
    movb 4(%esp), %al            # get character argument
    cmpb $'0', %al               # is it >= ascii digit 0?
    jb  retnot                  # if not, return not a digit
    cmpb $'9', %al               # is it <= ascii digit 9?
    ja  retnot                  # if not, return not a digit
    movl $1, %eax                # it is a digit, so return 'true'
    ret

retnot:
    xorl %eax, %eax              # it is not, so return 'false'
    ret

.end

u18(28)% make A=isdigit
...

u18(29)% mtip -f isdigit.lnx
...
Tutor> ~downloading isdigit.lnx
....Done.

Download done, setting EIP to 100100.
Tutor> go 100100
Character / is not a digit.
Character 0 is a digit.
Character 9 is a digit.
Character : is not a digit.
Exception 3 at EIP=00100110: Breakpoint
Tutor> ~q
Quit handler:
killing process 3531 Leaving board #1
u18(30)%