CS341 *** TURN OFF ALL CELLPHONES *** Practice
Final Exam B. Wilson

NAME ____

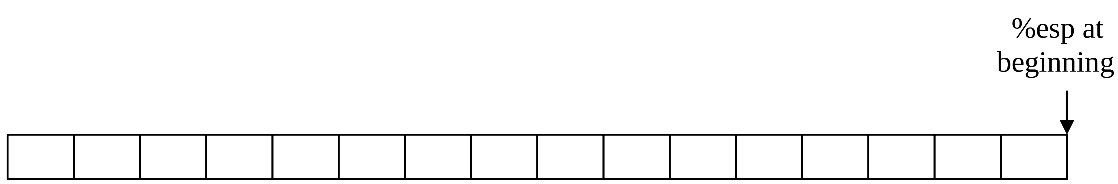
OPEN BOOK / OPEN NOTES: I GIVE PARTIAL CREDIT! SHOW ALL WORK!

1. Processor Architecture (20 points)
a. In a Harvard architecture processor like the ATMEGA, could you store assembly language instructions in an array, cast the address of the array to a function pointer, and execute it like I did in the i386 lecture demo? Explain your answer.

b. With an Operating System like Linux/UNIX, explain when you need to use a “software interrupt” (int $n) instead of just a normal “function call”.

2. Stack Management (20 points)
Show the contents of the memory in the stack area (in hex byte by byte) and the stack pointer value resulting from these i386 assembly language instructions located at these hex memory addresses (**where indicated**).

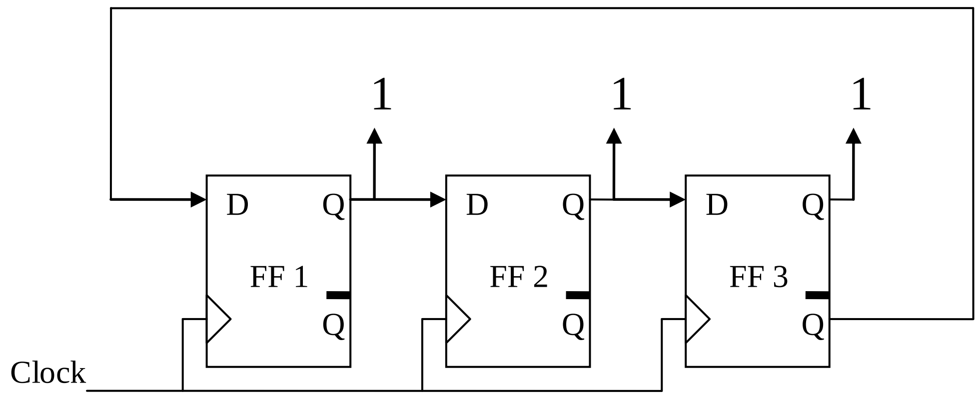
```
10f210      movl  $0x1234, %eax       # %ebp is 0xffffe342 here
10f215      pushl %eax
10f217      call  foobar              # foobar is located at 10f234
10f21c      addl $4, %esp
            ...
      foobar:
10f234      pushl %ebp
. . .       movl %esp, %ebp
            . . .       # show stack contents while executing this code
            movl %ebp, %esp
            popl %ebp
            ret
```



3. Labs (10 points)
a. In Lab 4 instead of using a checksum, explain how a “magic number” could be used.

b. In Lab 8, what would happen if you installed the diode in the wrong direction? Explain your answer.

4. (20 points) Sequential Hardware Logic
Study the following sequential logic diagram for a shift register which starts with the values of the Q signals on the D flip-flops as shown. This arrangement of flip flops is called a “Moebius counter”.



a. What will be the values of the Q signals on the D flip-flops after these clock pulses?

	Q1	Q2	Q3	
0	1	1	1	Initial Values
1	___	___	___	After 1 clock pulse
2	___	___	___	After 2 clock pulses
3	___	___	___	After 3 clock pulses
4	___	___	___	After 4 clock pulses
5	___	___	___	After 5 clock pulses
6	___	___	___	After 6 clock pulses

b. What do you notice about the sequence of values on the Q outputs of the three flip-flops?

c. In this course, what name did we use for that kind of sequence of binary values?

5. (30 points) Assembly Language Programming Problem
Write a gas assembly language version of the C library function toupper( ). As specified in the C library, the toupper function should be a C callable function with the function prototype:

```
int toupper(int c);
```

If c is a lower case letter, toupper(c) returns the corresponding upper case letter; otherwise it returns c. An ASCII code chart is attached for reference.

American Standard Code for Information Interchange

Here is the **<u>ASCII Encoding</u>**, a correspondence of keyboard characters with integers from 0 to 127 (0x7F in hexadecimal, 0177 in octal)

char	hex	oct	char	hex	oct	char	hex	oct	char	hex	oct
NUL	00	000	SP	20	040	@	40	100	`	60	140
SOH	01	001	!	21	041	A	41	101	a	61	141
STX	02	002	"	22	042	B	42	102	b	62	142
ETX	03	003	#	23	043	C	43	103	c	63	143
EOT	04	004	$	24	044	D	44	104	d	64	144
ENQ	05	005	%	25	045	E	45	105	e	65	145
ACK	06	006	&	26	046	F	46	106	f	66	146
BEL	07	007	'	27	047	G	47	107	g	67	147
BS	08	010	(	28	050	H	48	110	h	68	150
HT	09	011	)	29	051	I	49	111	i	69	151
NL/LF	0A	012	*	2A	052	J	4A	112	j	6A	152
VT	0B	013	+	2B	053	K	4B	113	k	6B	153
NP/FF	0C	014	,	2C	054	L	4C	114	l	6C	154
CR	0D	015	-	2D	055	M	4D	115	m	6D	155
SO	0E	016	.	2E	056	N	4E	116	n	6E	156
SI	0F	017	/	2F	057	O	4F	117	o	6F	157
DLE	10	020	0	30	060	P	50	120	p	70	160
DC1	11	021	1	31	061	Q	51	121	q	71	161
DC2	12	022	2	32	062	R	52	122	r	72	162
DC3	13	023	3	33	063	S	53	123	s	73	163
DC4	14	024	4	34	064	T	54	124	t	74	164
NAK	15	025	5	35	065	U	55	125	u	75	165
SYN	16	026	6	36	066	V	56	126	v	76	166
ETB	17	027	7	37	067	W	57	127	w	77	167
CAN	18	030	8	38	070	X	58	130	x	78	170
EM	19	031	9	39	071	Y	59	131	y	79	171
SUB	1A	032	:	3A	072	Z	5A	132	z	7A	172
ESC	1B	033	;	3B	073	[	5B	133	{	7B	173
FS	1C	034	<	3C	074	\	5C	134	|	7C	174
GS	1D	035	=	3D	075	]	5D	135	}	7D	175
RS	1E	036	>	3E	076	^	5E	136	~	7E	176
VS	1F	037	?	3F	077	_	5F	137	DEL	7F	177

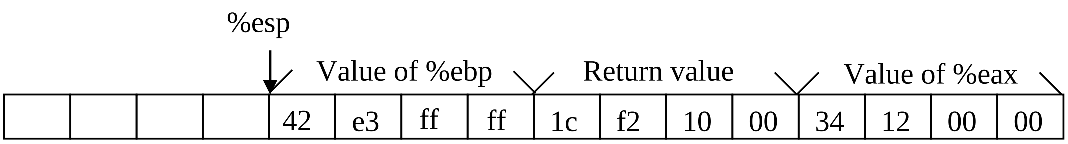
FINAL EXAM SOLUTIONS:

1. Processor Architecture

a. No, because these instructions are stored in the array in the data address space – not the program memory address space needed for execution. It works in the i386 because its Von Neumann architecture keeps code and data in the same address space.

b. You need to use a “software interrupt” to switch the processor from user mode into kernel mode to use restricted instructions and protected memory locations. A normal function call does not do that.

2. State of the stack and stack pointer



3.

a. A “magic number” is an arbitrary sequence of hex digits that would not be likely to appear by chance in the EEPROM memory. If its value is found in the appropriate location, your code assumes that configuration has been initialized. If not, your code needs to reinitialize the configuration.

b. When the transistor is turned on, the diode would be forward biased, have a low resistance, and carry most of the current around the motor. The motor wouldn’t get enough current to operate.

4. Sequential Hardware Logic

a.

Clock	Q1	Q2	Q3	
0	1	1	1	Initial Values
1	0	1	1	After 1 clock pulse
2	0	0	1	After 2 clock pulses
3	0	0	0	After 3 clock pulses
4	1	0	0	After 4 clock pulses
5	1	1	0	After 5 clock pulses
6	1	1	1	After 6 clock pulses

b. In the sequence of binary digits, only one Q value changes state with each clock pulse.

c. That sequence of binary digit values is “gray coded”.

5. Assembly Language Programming Problem

```
# toupper.s: assy version of library toupper function

        .text
        .globl  _toupper
_toupper:
        movl    4(%esp), %eax       # get argument character
        cmpb    $'a'                # if below 'a'
        jb      return              # or
        cmpb    $'z'                # if above 'z'
        ja      return              # return with original character
        subl    $'a' - 'A', %eax    # else change character value
return:
        ret
        .end
```