CPSC 213, Winter 2010, Term 1 — Quiz 1

Date: Oct 6, 2010; Instructor: Tamara Munzner

NAME: ____ **STUDENT NUMBER:** ______________

LAB DAY/TIME: ________________________

1 For each of the following, give the smallest number greater than or equal to `0x1009` that is aligned as indicated.

1a Aligned for access to 2-byte shorts:

1b Aligned for access to 4-byte longs:

1c Aligned for access to 8-byte long longs:

1d Aligned for access to 16-byte chunks of memory:

2 What is does the following program output on a Little Endian (e.g., Intel) processor?

```
int main (char** argc, int argv) {
    char a[4];
    *((int*)(a)) = 0x01020304;
    printf ("a[2] = %d",a[2]);
}
```

The program outputs this string (you fill in the rest): `a[2] =`

3 Explain the tradeoff between static and dynamic arrays in C by giving one benefit of using each over the other.

3a **Benefit of static arrays** compared to dynamic arrays:

3b **Benefit of dynamic arrays** compared to static arrays:

4 Give an RTL description and SM213 assembly code that a compiler might generate for this code. (There may be some code in between the declarations in the first line and the assignments in the last three lines, so `a` may have been assigned to any value.) Assign static values as the compiler would; explain values you use. The SM213 ISA documentation is provided on the last page for reference.

C:

```
int b[5], i, a*;

i = 2;
b[i] = 5;
a[i] = 3;
```

RTL: **SM213 Assembly:**

OpCode	Format	Semantics	Eg Machine	Eg Assembly
load immediate	0d-- vvvvvvvv	$r[d] \leftarrow v$	0100 00000100	ld $0x1000,r1
load base	1osd	$r[d] \leftarrow m[o \times 4 + r[s]]$	1123	ld 4(r2),r3
load indexed	2sid	$r[d] \leftarrow m[r[i] \times 4 + r[s]]$	2123	ld (r1,r2,4),r3
store base+dis	3sod	$m[o \times 4 + r[d]] \leftarrow r[s]$	3123	st r1,8(r3)
store indexed	4sdi	$m[r[i] \times 4 + r[d]] \leftarrow r[s]$	4123	st r1,(r2,r3,4)
halt	f000		f000	halt
nop	ff00		ff00	do nothing (nop)
rr move	60sd	$r[d] \leftarrow r[s]$	6012	mov r1, r2
add	61sd	$r[d] \leftarrow r[d] + r[s]$	6112	add r1, r2
and	62sd	$r[d] \leftarrow r[d]\ \&\ r[s]$	6212	and r1, r2
inc	63-d	$r[d] \leftarrow r[d] + 1$	6301	inc r1
inc addr	64-d	$r[d] \leftarrow r[d] + 4$	6401	inca r1
dec	65-d	$r[d] \leftarrow r[d] - 1$	6501	dec r1
dec addr	66-d	$r[d] \leftarrow r[d] - 4$	6601	deca r1
not	67-d	$r[d] \leftarrow !r[d]$	6701	not r1
shift	7dss	$r[d] \leftarrow r[d] << s$	7102 71fe	shl $2, r1 shr $2, r1
branch	8-oo	$\text{pc} \leftarrow \text{pc} + 2 \times o$	1000: 8004	br 0x1008
branch if equal	9roo	if $r[r] == 0,\ \text{pc} \leftarrow \text{pc} + 2 \times o$	1000: 9104	beq r1, 0x1008
branch if greater	aroo	if $r[r] > 0,\ \text{pc} \leftarrow \text{pc} + 2 \times o$	1000: a104	bgt r1, 0x1008
jump	b--- aaaaaaaa	$\text{pc} \leftarrow a$	b000 00001000	jmp 0x1000
get program counter	6f-d	$r[d] \leftarrow \text{pc}$	6f01	gpc r1
jump indirect	croo	$\text{pc} \leftarrow r[r] + 2 \times o$	c102	jmp 8(r1)
jump double ind, b+disp	droo	$\text{pc} \leftarrow m[4 \times o + r[r]]$	d102	jmp *8(r1)
jump double ind, index	eri-	$\text{pc} \leftarrow m[4 \times r[i] + r[r]]$	e120	jmp *(r1,r2,4)