I科104 E94041335 鄭建文

Computer Organization & Assembly Language

Midterm Exam – 2013/11/15

Dept. of Engineering Science, National Cheng Kung University

1. Answer "True" or "False" to the following statements. (24%)

T (A) The interface between application software and system software is called *instruction set architecture*.

F (B) The compiler always maps a C language statement to several MIPS instructions.

T (C) *Datapath* and *control* are two important components which constitute a CPU.

T (D) Power consumption is typically an additional constraint for designing a mobile device.

T (E) The yield decreases as there are more and more defects on the wafer.

T (F) Benchmark refers to programs used to measure performance.

F (G) Memory is a hierarchy of devices with faster and more expensive ones closer to CPU.

F (H) The "shamt" field is not used by the `add` instruction but is used by the `sw` instruction.

2. Two machine designs have been proposed, A and B. Assume that Machine A's clock rate is *500MHz* and Machine B's clock rate is *400MHz*. Both machines have three classes of instructions. The *cycle counts per instruction* for the three classes are as follows:

	Machine A	Machine B
Class X	4	2
Class Y	3	4
Class Z	2	1

Two programs of interest that will be run on these machines require the following *number of instructions (in billions)* for each instruction class:

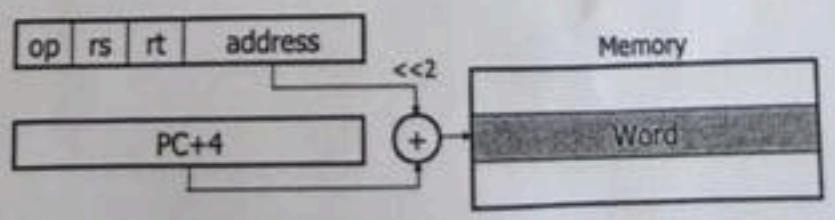
	Program 1	Program 2
Class X	1	1
Class Y	2	2
Class Z	5	10

(A) Which program is faster on Machine A? Please calculate the average CPI for both programs. (4%)

(B) Which program will execute faster according to MIPS (million instruction per second) on Machine A? (4%)

(C) How much faster (or slower) is Machine A compared to Machine B if two programs were run successively on these two machines? (6%)

3. What is the following addressing mode? What MIPS instruction(s) are appropriate for this addressing mode? Also, use the following diagram to describe how this works in details. (12%)



4. Compile the following C program into MIPS instructions. Assume that the usage of registers is specified as (`f: $s0`). (16%)

```
int function_x (int* a, int h){
    int f;
    if(h >= 0)      f = a[10] + h;
    else            f = a[10] - h;

    return f;
}
```

5. Assume that the variables `f`, `g`, `h`, `i`, and `j` are assigned to registers `$s0`, `$s1`, `$s2`, `$s3`, and `$s4`, respectively. Also, assume that the base address of the integer arrays `A` and `B` are in registers `$s6` and `$s7`, respectively.

For the C statements below, what are the corresponding MIPS instructions?

(A) `f = - g + h + B[1];` (4%)

(B) `f = A[ B[g] + 1 ];` (4%)

(C) `B[3] = 15*f + A[1];` (4%)

6. Make a comparison among the following three MIPS instructions: `j`, `jr` and `jal`. (6%)

7. Given a 32-bit bit pattern: 00110010 01010001 00101010 01010010

(A) What is the corresponding hexdecimal representation if it is an integer? (4%)

(B) What is the corresponding string if it is an ASCII string? (4%)

(C) What is the corresponding instruction if it is an MIPS instruction? (4%)

(D) Assume that $t0=$t1=$t2=$t3=23_{ten} and $s0=$s1=$s2=$s3=51_{ten}. If the instruction decoded in (C) is then executed, which register is updated? What is the new value (in decimal) for this register? (4%)

References:

ASCII value	Character	ASCII value	Character	ASCII value	Character	ASCII value	Character	ASCII value	Character	ASCII value	Character
32	space	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	|
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	DEL

instruction	format	op	rs	rt	rd	shamt	funct	address
add	R	0	reg	reg	reg	0	32_{ten}	n.a.
sub	R	0	reg	reg	reg	0	34_{ten}	n.a.
addi	I	8_{ten}	reg	reg	n.a.	n.a.	n.a.	constant
lw	I	35_{ten}	reg	reg	n.a.	n.a.	n.a.	address
sw	I	43_{ten}	reg	reg	n.a.	n.a.	n.a.	address