EEL 6764 Principles of Computer Architecture
Practice #1– Sample Solutions

1 Problems

Complete the following problems at the end of Appendix A.

- **A9 (a)** The possible encodings are show in the table below.

	addr[13:12]	**addr[11:6]**	**addr[5:0]**
3 two-address instructions	'00', '01', '10'	'000000' to '111111'	'000000' to '111111'
63 one-address instructions	'11'	'000000' to '111110'	'000000' to '111111'
45 zero-address instructions	'11'	'111111'	'000000' to '101100'
19 unused encodings	'11'	'111111'	'101101' to '111111'

  In this encoding, three 2-address instructions are encoded with "00", "01", and "10" on bits [13:12]. Then, "11" on bits [13:12] encodes all other instructions. For the one-address instructions, the opcode is extended by using 63 of the possible 64 combinations from the bits [11:6], that is, from "000000" to "111110". Consequently, the opcode of zero-address instructions is extended with the remaining combination of "111111" from addr[11:6]. There are 45 zero-address instructions, so using "000000" to "101100" from bits [5:0] suffices for the encoding.

- **A9 (b)** The 3 two-address instructions can be encoded similar to part (a). The one-address and zero-address instructions must be encoded using "11" bits [13:12]. In order to encode 65 one-address instructions, at least 7 bits are required, so bits [11:5] can be used for this encoding. Then bits [4:0] is left for extending the opcode of zero-address instructions. The maximum number of zero-address instructions that can be encoded for this processor 32, thus encoding 35 zero-address instructions is **not** possible.
- **A10 (a)** See below

a. 1. Stack

Code	Register addresses	Memory addresses	Code size (bytes)
PUSH A		1	9
PUSH B		1	9
ADD			1
POP C		1	9
Total	0	3	28

2. Accumulator

Code	Register addresses	Memory addresses	Code size (bytes)
LOAD A		1	9
ADD B		1	9
STORE C		1	9
Total	0	3	27

3. Register-memory

Code	Register addresses	Memory addresses	Code size (bytes)
LOAD R1, A	1	1	9.75
ADD R3, R1, B	2	1	10.5
STORE R3, C	1	1	9.75
Total	4	3	30

4. Register-register

Code	Register addresses	Memory addresses	Code size (bytes)
LOAD R1, A	1	1	9.75
LOAD R2, B	1	1	9.75
ADD R3, R1, R2	3		3.25
STORE R3, C	1	1	9.75
Total	5	3	32.5

- **A10 (b)** See below

b. 1. Stack

Code	Destroyed data	Overhead data (bytes)	Code size (bytes)	Moved memory data (bytes)
PUSH A			9	8
PUSH B			9	8
ADD	A and B		1	
POP C	C		9	8
PUSH E			9	8
PUSH A		8	9	8
SUB	A and E		1	
POP D	D		9	8
PUSH C		8	9	8
PUSH D		8	9	8
ADD	C and D		1	
POP F	F		9	8
Total		24	84	72

2. Accumulator

Code	Destroyed data	Overhead data (bytes)	Code size (bytes)	Moved memory data (bytes)
LOAD A			9	8
ADD B	A		9	8
STORE C			9	8
LOAD A	C	8	9	8
SUB E	A		9	8
STORE D			9	8
ADD C	D		9	8
STORE F			9	8
Total		8	72	64

3. Register-memory

Code	Destroyed data	Overhead data (bytes)	Code size (bytes)	Moved memory data (bytes)
LOAD R1, A			9.75	8
ADD R3, R1, B			10.5	8
STORE R3, C			9.75	8
SUB R5, R1, E			10.5	8
STORE R5, D			9.75	8
ADD R6, R3, D			10.5	8
STORE R6, F			9.75	8
Total		0	70.5	56

4. Register-register

Code	Destroyed data	Overhead data (bytes)	Code size (bytes)	Moved memory data (bytes)
LOAD R1, A			9.75	8
LOAD R2, B			9.75	8
ADD R3, R1, R2			3.25	
STORE R3, C			9.75	8
LOAD R4, E			9.75	8
SUB R5, R1, R4			3.25	
STORE R5, D			9.75	8
ADD R6, R3, R5			3.25	
STORE R6, F			9.75	8
Total		0	68.25	48

- **A22 (a)** The ASCII interpretation of the 64-bit word using Big Endian byte order is as follows:

52	49	53	43	56	43	50	55
R	1	S	C	V	C	P	U

- **A22 (b)** The ASCII interpretation of the 64-bit word using Little Endian byte order is as follows:

520	55	56	43	53	43	52	49
P	U	V	C	S	C	R	I