

Digital Computer Design

1. Consider memory storage of a 32-bit word stored at memory word 42 in a byte-addressable memory.

(a) What is the byte address of memory word 42?

(b) What are the byte addresses that memory word 42 spans?

(c) Draw the number 0xFF223344 stored at word 42 in both big-endian and little-endian machines. Clearly label the byte address corresponding to each data byte value.

2. The NOR instruction is not part of the ARM instruction set, because the same functionality can be implemented using existing instructions. Write a short assembly code snippet that has the following functionality: $R0 = R1 \text{ NOR } R2$. Use as few instructions as possible.

3. The NAND instruction is not part of the ARM instruction set, because the same functionality can be implemented using existing instructions. Write a short assembly code snippet that has the following functionality: $R0 = R1 \text{ NAND } R2$. Use as few instructions as possible.

4. Consider the following high-level code snippets. Assume the (signed) integer variables g and h are in registers R0 and R1, respectively.

```
(i)   if (g >= h)
        g = g + h;
      else
        g = g - h;
(ii)  if (g < h)
        h = h + 1;
      else
        h = h * 2;
```

Write the code snippets in ARM assembly language.

5. Consider the following high-level code snippet. Assume that the base addresses of array1 and array2 are held in R1 and R2 and that array2 is initialized before it is used.

```
int i;
int array1[100];
int array2[100];
...
for (i=0; i<100; i=i+1)
    array1[i] = array2[i];
```

Write the code snippet in ARM assembly.

6. The high-level function strcpy copies the character string src to the character string dst.

```
// C code
int i = 0;
```

```
do {  
    dst[i] = src[i];  
} while (src[i++]);
```

Write the code snippet in ARM assembly. Use R4 for i. Assume that the base addresses of dst and src are held in R1 and R2

7. Consider the following high-level code snippet. Assume that the base addresses of temp is held in R1. Use R4 for i. Write the code snippet in ARM assembly.

```
int i;  
int temp[100];  
...  
for (i=0; i<100; i=i+1)  
    temp[i] = temp[i]*128;
```