Digital Design and Computer Architecture (E85)

J. Spjut

1) Problems

1. Implement the following high-level code segments using the `CMP` instruction and conditional execution. Assume the integer variables g and h are in registers R1 and R2, respectively.

a)

```
if (g > h)
     g = g + h;
else
     g = g - h;
```

b)

```
if (g >= h)
     g = g + 1;
else
     h = h - 1;
```

c)

```
if (g <= h)
     g = 0;
else
     h = 0;
```

2. The strcpy function copies the character string from src to dst.

```
// high-level code
void strcpy(char x[], char y[]) {

  int i = 0;
  while (x[i] != 0) {
    y[i] = x[i];
    i = i + 1;
  }
  y[i] = '\0';
}
```

a)Implement the strcpy function in ARM assembly code. Use R1 for i.

b) Draw a picture of the stack before, during, and after the strcpy function call. Assume the stack pointer is at 0x7FFFFF00 just before strcpy is called.

3. Consider the following high-level function.

```
// high-level code
int f(int n, int k) {
     int b;

     b = k + 2;
     if (n == 0) b = 10;
     else b = b + (n * n) + f(n - 1, k + 1);
     return b * k;
}
```

a) Translate the high-level function into ARM assembly language. Clearly comment your code. You can use the multiply instruction. Keep local variable b in R4.

b) Step through your function from part (a) by hand for the case of f(2, 4). Draw a picture of the stack. Write the register name and data value stored at each location in the stack and keep track of the stack pointer value (SP). Clearly mark each stack frame. Assume that when f is called, R4 = 0xABCD and LR = 0x400004. What is the final value of R0?

4. Show how the following ARM program would be loaded into memory and executed.

```
# ARM assembly
main:
     ADD  SP, SP, #-4
     STR  LR, [SP, #0]
     LDR  R0, X
     LDR  R1, Y
     BL   diff
     LDR  LR, [SP, #0]
     ADD  SP, SP, #4
     MOV  PC, LR
diff:
     SUB  R3, R0, R1
     MOV  PC, LR
```

a) First show the instruction address next to each assembly instruction.

b) Draw the symbol table showing the labels and their addresses.

c) Convert all instructions into machine code

d) How bit (how many bytes) are the data and text segments?

e) Sketch a memory map showing where data and instructions are stored.

4) Time

Please indicate how many hours you spent on this problem set. This will not affect your grade (unless you omit it completely), but will be helpful for calibrating the workload for next semester's class.