Tutorial 2: Programmable Model of Computer

Problem 1: C Functions

Consider the C code in Figure 1. What choice for the blank ensures that `7 = 7` is printed?

```
#include <stdio.h>

int *f (int varNotToSet, int varToSet) {
     int n = 7;
     varToSet = 7;
     return  ____ ;
  /* &n, &varNotToSet, or &varToSet could go here */
}
int main ( ) {
     int *ptr;
     int k1 = 7, k2 = 0;
     ptr = f(k1, k2);
     printf ("7 = ");
     printf ("%d\n", *ptr);
     return 0;
}
```

Figure 1: Program on Data Type Conversion

Problem 2: ARM Programmer's Architecture

Consider the ARM Processor architectures. Answer the following questions.

How many registers are visible to the programmer?

What is the size of these registers?

If the content of register `R2` is 134, draw a picture to show the placement of the number 134 in the register bits.

Problem 3: Data Manipulation Instructions

Consider the assembly code in Figure 3.

Provide a comment for each line assembly instruction.

If register R2 contains 23, what are the contents of registers R1 – R4 after the execution all the assembly instructions?

```
mov r1, r2
add r3, r1, r2
and r4, r3, r2
orr r4, r3, r4
```

Figure 3: Program on arithmetic instructions

Problem 4: Load Store Instruction

Consider the assembly code in Figure 5.

Provide a comment for each line assembly instruction.

If register R2 contains 100, and content of memory at locations 100 and 104 are 45 and 1004, what are the contents of registers R1 – R3 and memory locations 100 and 104 after the execution all the assembly instructions?

```
ldr r1, [r2]
ldr r2, [r2+4]
add r3, r1, r2
str r3, [r2 + 4]
```

Figure 5: Program on load store instructions