EXAM 1: NEXT WEEK

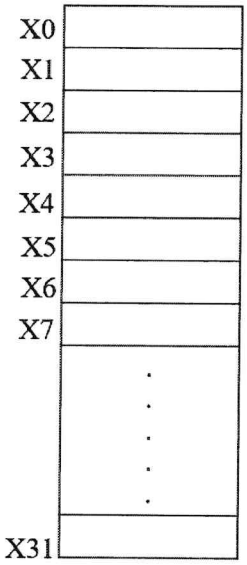
Software simulation (ARMSim)

Assembly language will be simulated using ARMSim Simulator. LEGv8 and ARMSim (RMv7) are almost similar with little distinction.

LEGv8 VS ARMv7 (ARMSim) Registers

ARMv8

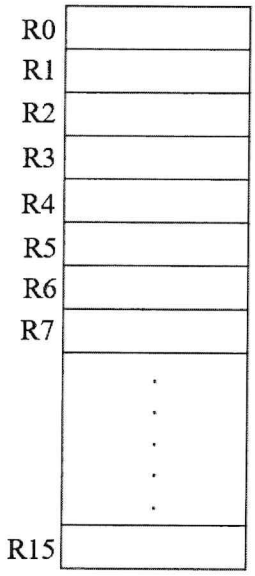
LEGv8 Registers



32 registers
64 bit

ADD X0, X1, X2

ARMv7 Registers (ARMSim)



16 registers
32 bit

ADDI R0, R1, R2

LEGv8	ARMSim
ADDI	ADD
SUBI	SUB
MOVI	MOV
LDUR	LDR
LDURB	LDRB
STUR	STR
CMPI	CMP

ADD R1, R1, #2

MOV R1, #10

Program: A C++ program adds two given numbers a=10 and b=20 and prints resulting number. Convert the following C++ code to assembly code and simulate using ARMSim Version 2.1.

C++ Code:

```
#include<iostream>
using namespace std;
int main()
{
   int a=10, b=20, c;
   c= a + b;
   cout<<c;
   return 0;
}
```

X2 → a

X3 → b

LEGv8 Instructions: ARMv8

```
MOVI X2, #10      @ X2 =10
MOVI X3, #20      @ X3 = 20
ADD X1, X2, X3    @ X1= X2+X3 =30
```

simulator - ARMv7

```
MOV R2, #10      // a = 10
MOV R3, #20      // b = 20
ADD MOV R1, R2, R3   // c = a+b
```

Program: A C++ program to find the factorial of 3. Convert the following C++ code to assembly code and simulate using ARMSim version 2.1.

C++ Code:

```
#include <iostream>
using namespace std;

int main()
{
 int i, j=1, n=3;
 for (i = 1; i <= n; i++)
 {
   j = j * i;
 }
 cout<<j;
}
```

ARMSim — ARM V7

R2 → n R5 → j R6 → i

```
MOV R5, #1      // j = 1
MOV R2, #3      // n = 3
MOV R6, #1      // i = 2
LOOP: CMP R6, R2    // compare i and n.
B.GT exit       // i > n — Go to exit
MUL R5, R5, R6  // j = j * i
ADD R6, R6, #1  // i++
B LOOP
exit:
```

i > n → Go to exit

// Write a C++ program to add two given integer numbers using user defined function.

```
#include <iostream>
using namespace std;

int add (int x, int y); // function declaration
int main() // main code
{
    int a=5, b=4, c;
    c = add(a, b); // function call
    cout<<"Result:"<<c;
    return 0;
}

int add (int x, int y) // function definition
{
    return (x + y);
}
```

e 9 | 9 5 | b 4

5 4

5 + 4 = 9

OUTPUT

9

Result: 9

BL → Branch Linking. R1 R2 R9

a b c.

@ Write a C++ program to add two given integer numbers using user defined function.

```
106   MOV  R1, #5        // a = 5
104   MOV  R2, #4        // b = 4
108 → BL addition
112   B Exit
→addition:
      ADD R9, R1, R2     // R9 = 9
      BX LR
Exit:
```

108