John L.  
"Computer Organization & Design : The Hardware / Software Interface", Morgan Kaufmann Publishers.

Let us assume a) the stack grows towards lower addresses, b) stack pointer points to the top filled location, c) Each item on stack is a full word (32 bits) and stack addresses are word align.  Given these assumptions, you can do the following operation. For push Ri, use str Ri, [sp, #-4]! For pop Ri, use ldr Ri, [sp], #4  Saving and restoring of multiple registers can be done using stmfd and ldmfd as follows.

For pushing r0 to r4 and lr, use stmfd sp!, {r0-r4, lr}

For popping r0 to r4 and lr, use ldmfd sp!, {r0-r4, lr}

Typically, you would be popping some registers and lr at the exit of a function. This means that the above ldmfd instruction would be followed by mov pc, lr. In such a case you can directly pop lr value into pc by using instruction ldmfd sp!, {r0-r4, pc}.

2. Command to add directly

mov r0, #2

mov r1, #2

cmp r0, r1

addeq r0,#1

3. Instructions in multiply group share F=00 with DP group, but have some differences in other fields.

4. pre and post index:

Pre-indexing

ldr r1, [r2, #4]     means       r1 <= memory[r2 + 4];

ldr r1, [r2, #4]!    means       r2 <= r2 + 4; r1 <= memory[r2];

Post-indexing

ldr r1, [r2], #4     means       r1 <= memory[r2]; r2 <= r2 + 4;

5. Sir/Ma'am, could you please explain why do we need 8 bits for specifying shift for operand2? Why can't we simply mention the register number in the 4 bits assigned to us, since we have only 16 registers available so what is the function of the shift amount? Please sir/ma'am, could you explain.

This 8 bit field specifies the shift amount, that is, how much you want to shift and shift type, that is, the manner in which you want to shift.

\* The manner of shifting or type of shifting has 4 options - LSL, LSR, ASR and ROR. Two bits out of the 8 bits specify this.

\* Shift amount can be a constant (i.e. immediate) or defined by contents of a register. One bit out of 8 bits specifies which one of these two alternatives is being used.

\* If shift amount is a constant, 5 bits are used to specify this.

\* If shift amount is given by a register, 4 bits are used to specify the register number.

 many bit position