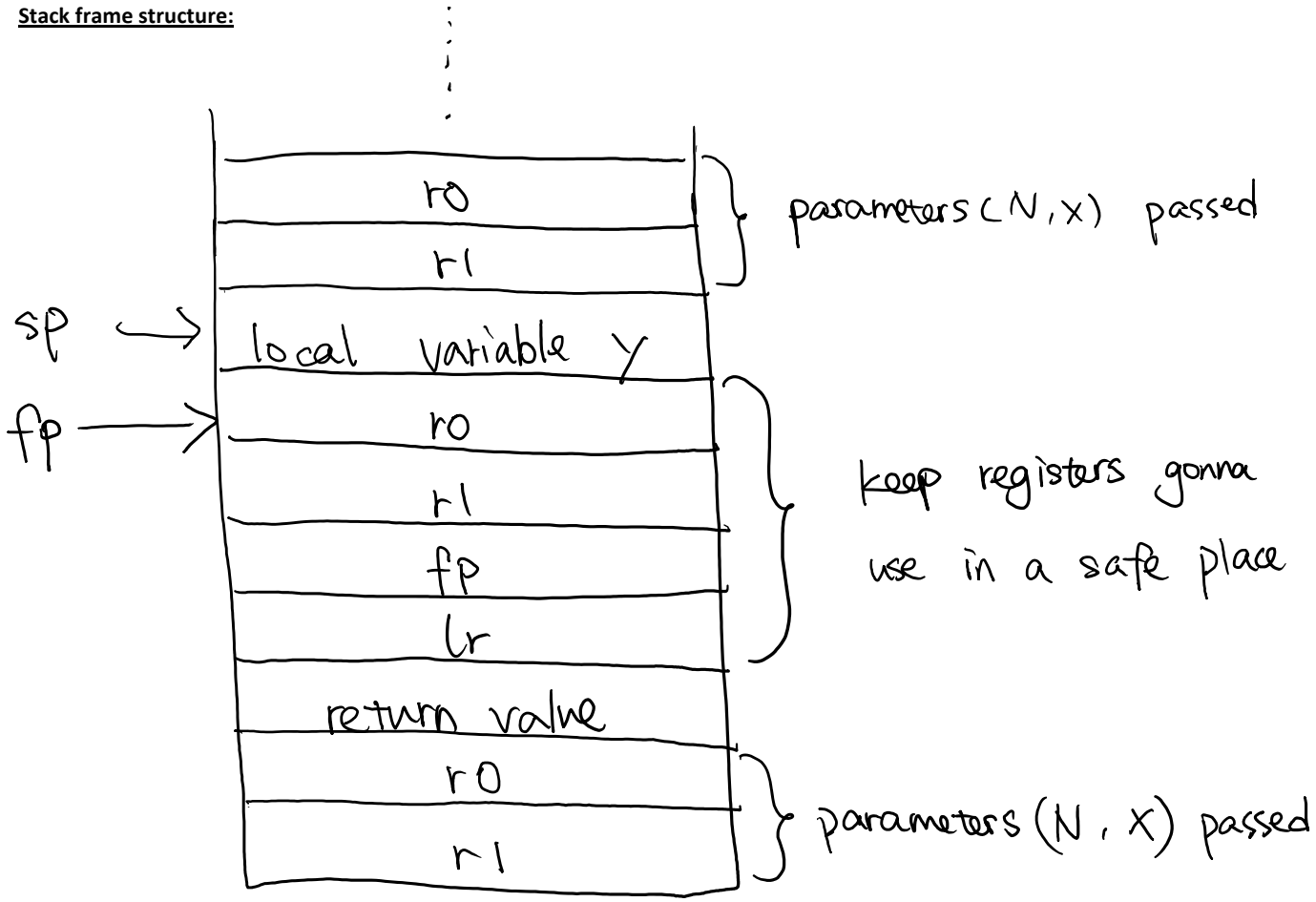
Assignment 5

2018年4月3日 20:58

Code:

```
                AREA question,CODE,READONLY
                ENTRY

N               EQU 4                       ;define N=4
X               EQU 3                       ;define X=2

                MOV     r0,#X               ;load the value of x into r0
                MOV r1,#N                   ;load the value of n into r1
                ADR sp,Stack                ;let sp points to the top of the stack
                STMFD sp!, {r0,r1}          ;pass x and n as parameters by pushing them onto stack
                SUB sp, sp, #4              ;reserve space for the return RESULT
                BL POWER                    ;call power function

                LDR r2, [sp], #12           ;load return value into r2 and deallocate the stack
                ADR r3, RESULT              ;let r3 points to RESULT
                STR r2, [r3]                ;store return value of POWER in RESULT

EXIT            B EXIT                      ;end of the program

POWER           STMFD sp!, {r0,r1, fp, lr}  ;push registers to be used, fp and lr to the Stack
                MOV fp, sp                  ;let frame pointer points to the top of the stack
                SUB sp, sp, #4              ;reserve space in the stack for local variable y
                LDR r1, [fp, #24]           ;load parameter n to r1
                CMP r1, #0                  ;base case, check if n is 0
                MOVEQ r1, #1                ;if n is 0, return 1
                BEQ RETURN                  ;then also branch to RETURN

                LDR r0, [fp, #20]           ;if n is not 0, continue to loa parameter x that was passed
                TST r1, #2_00000001         ;use last bit of n to test if n is even or odd
                BNE ODD                     ;if last bit of n is 1, n is odd, branch to ODD
                BEQ EVEN                    ;if last bit of n is 0, n is even, branch to EVEN

EVEN            ASR r1, #1                  ;divide n by 2,using arithmetic right shifting by 1
                STMFD sp!, {r0,r1}          ;pass x and n parameters by pushing them onto stack
                SUB sp, sp, #4              ;reserve a space on the Stack for the return value
                BL POWER                    ;call power function

                LDR r0, [fp, #-16]          ;get returned value from above recursive call to POWER
                STR r0, [fp, #-4]           ;store returned value in local variable y on the stack
                MUL r1, r0, r0              ;multiply y by y and store result into r1
                B RETURN                    ;branch to RETURN

ODD             SUB r1, r1, #1              ;decrement n by 1
                STMFD sp!, {r0,r1}          ;pass x and n parameters by pushing them onto the Stack
                SUB sp, sp, #4              ;reserve a space on the Stack for the return value
                BL POWER                    ;call power function

                LDR r1, [fp, #-16]          ;get return value from above recursive call to POWER
                MUL r1, r0, r1              ;multiply x by previous returned value,which is in r1, and store result in r1

RETURN          STR r1, [fp, #16]           ;store value in r1 to the return value location on stack
                MOV sp, fp                  ;deallocate the stack frame
                LDMFD sp!, {r0,r1,fp,pc}    ;restore modified registers along with fp, and move lr into pc

                space 0x200                 ;reserve space for stack
Stack           DCD     0x00                ;top of the stack
RESULT          DCD 0x00                    ;reserve space to store final result

                END
```

Stack frame structure:



Question: How many stack frames are needed to calculate x^n, when n = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12?

Answer:

Value of n	Number of stack frames needed
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12