EE2016
Midsem

Question 1)

In the first question, we are required to minimize the total number of LDR/STR instructions. Therefore, we require 6 registers to store values of A[i] to A[i+5]. In addition, we require a register to store the minimum value of S obtained, one to store the corresponding index and one to store i to keep track of the number of iterations of the loop. Therefore in total, we require overall 11 registers in this code. Of these 11 registers, 4 registers need not be saved. Therefore, a total of 8 registers are to be saved and restored.

Initially, r0 is stored with address of A[0] and r1 is stored with “n”. Before the loop, A[5] to A[0] are stored in r1 – r6.

Registers:

r0 : To store A[i+5]

r7 : To store S minimum

r8 : To store index starting from n-7 which decrements by 1 to track no of iterations

r9 : To store n-6-i0. Here i0 is the index which corresponds to S minimum

r10 : To store present value of S

r11 : To store n-6

Just before the loop starts, r10 is stored with S0. Since it is the first S calculated, it is stored as the minimum S and n-6 is stored in r9.

When loop starts, i = 1 and A[i+4] is stored in r1, A[i+3] is stored in r2, A[i+2] is stored in r3, A[i+1] is stored in r4 and A[i] is stored in r5. Now A[i+5] is stored in r6. And using these registers S1 is calculated and stored in r10. Now we check whether S1 is less than S minimum or not. If it is less, then we store S1 in r7 and store r8 as index corresponding to S1 in r9 and decrements r8. If it is not less, then is just decrements r8.

After incrementing r8, we now have r8 = n-8 which is same as i =2. Now, A[i+4] is stored in r6, A[i+3] is stored in r1, A[i+2] is stored in r2, A[i+1] is stored in r3 and A[i] is stored in r4. Now A[i+5] is stored in r5. Using these registers, S2 is calculated and stored in r10. S2 is compared with S minimum. If S2 is smaller than S, it is stored as minimum and corresponding r8 is stored in r9 and r8 decrements. If not smaller, then just r8 decrements.

Thus after each decrement in r8, the values A[i] to A[i+5] are stored cyclically in registers r1-r6.This process continues untill r8 becomes less than zero. Once r8 is less than zero is breaks out of the loop and brances to X. r9 contains n-6-i0, i0 is the index corresponding to S which is minimum. Using r11 which is stored with n-6, r9 is subtracted from r11 and stored in r9. Now r9

is stored with i0 which is the index corresponding to minimum S. This value is moved to r0 which returns this value to the main program.
The totla number of clock cycles taken by the function = 116*L + 966

Here L is clock cycles taken by LDR

Number of clock cycles requires for an array of size 100 = 1546
Number of registers required to be saved = 8 (r4,r5,r6,r7,r8,r9,r10,r11)

Cycles taken by LDR/STR	1	5	10	20
Cycles taken by the function	1082	1546	2126	3286

Question 2)

In the second question, we are required to minimize the total clock cycles given that LDR instruction takes equal time as other instuctions. In this question, we require 9 registers. Initially r0 is stored with address of A[0], r1 is stored with n.

Registers :

- r0 : To store address of A[i-1]
- r1 : To store n
- r2 : To store n-6-i0. i0 is the index corresponding to minimum S.
- r3 : To store address of A[i+5]
- r4 : To store S of previous iteration
- r5 : To store S of present iteration
- r6 : Is used as scratch
- r7 : To store minimum S
- r8 : To store index which decrements every iteration to keep track of them

We use the fact that

Si = A[i+5] – A[i-1] - Si-1

Initially r4 is stored with A[5]. Using r6 to load values of elements of array temporarily, we use r6 and add to or subtract it from r4 to get S0 in r4. This r4 is moved into r7, i.e., it is stored as S minimum as it is the first S we calculated. r8 is stored with n-7 and r2 is stored with n-6.

Now, in the loop, A[i+5] is stored in r5, and A[i-1] is stored in r6. Now subtracting r6 and r4 from r5 we get S in r5. This S is compared with the minimum S till now. If the present S is less than the minimum S, it is stored in minimum S and corresponding r8 is also stored in r2 and r8 is decremented. If it is not less, only r8 is decremented.

The loop continues till r8 is positive or zero. Once r8 is negative, it breaks from the loop.

r2 is now stored with n-6-i0. Using the value in r1 and subtracting r2 from n-6 and storing it in r2, we get i0 in r2. This r2 is stored in r0 which returns the value to main function.

This scheme can’t be used in first problem as the no of clock cycles taken by LDR is 5 times more than normal instructions which drastically increases the total clock cycles if we use two or more LDRs in the loop.

The total number of clock cycles taken by function in question 2 = 204*L + 765
Here L is clock cycles taken by LDR

No. of clock cycles taken by the function in question 2 = 969
No. of registers to be saved = 5

Cycles for LDR	1	5	10	20
Total Cycles for Q1	1082	1546	2126	3286
Total Cycles for Q2	969	1173	1377	1581