# EE2016 MIDTERM REPORT

**Ques1)**

In first question , we were required to minimise the number of ldr , str instructions. So we needed 11 registers. Initially the registers r0 and r1 contained the addresses of array and number of elements in array respectively. Initially the value of S[0] is stored . And values of arr[1] to arr[5] are stored in r2 to r6 respectively. I used r11 as I, r10 to give the value of I at which S[i] is minimum and r1 = n-5 .

In the loop, r2,r3,r4,r5,r6 are used to store the values of arr[I] to arr[i+5] respectively. S[i] is stored in r8. If S[i] is less than r9 , then we loaded S[i] into r9 and stored the value of I in r12 . The loop stops at i=k-6 , where k is number of elements in array.

For this code , I used the registers r1-r9,r11,r12 which are saved in stack and restored.

Calculating number of clock cycles:

Let n be the number of elements In the array.

7 push and pop instructions and 1 load for each element in the array , which are n clock cycles each.

And others instructions are only 1 clock cycle each.

Therefore , we get number of clock cycles as ,

$$114*n+1158$$

| Cycles of ldr | 1 | 5 | 10 | 20 |
|---|---|---|---|---|
| Total clock cycles | 1272 | 1728 | 2298 | 3438 |

Ques2

In this , we are given a hint and are required to use minimum number of registers possible. Yes , we can develop a faster algorithm.

I used the registers r0-r6,r12. So I saved and restored 3 registers.

This is not possible in 1st question because for doing this , we neede atleast 2 load instructions per iteration.

Number of clock cycles required are, 292*n+1044

| Cycles of ldr | 1 | 5 | 10 | 20 |
|---|---|---|---|---|
| Total clock cycles | 1336 | 2504 | 3964 | 6884 |