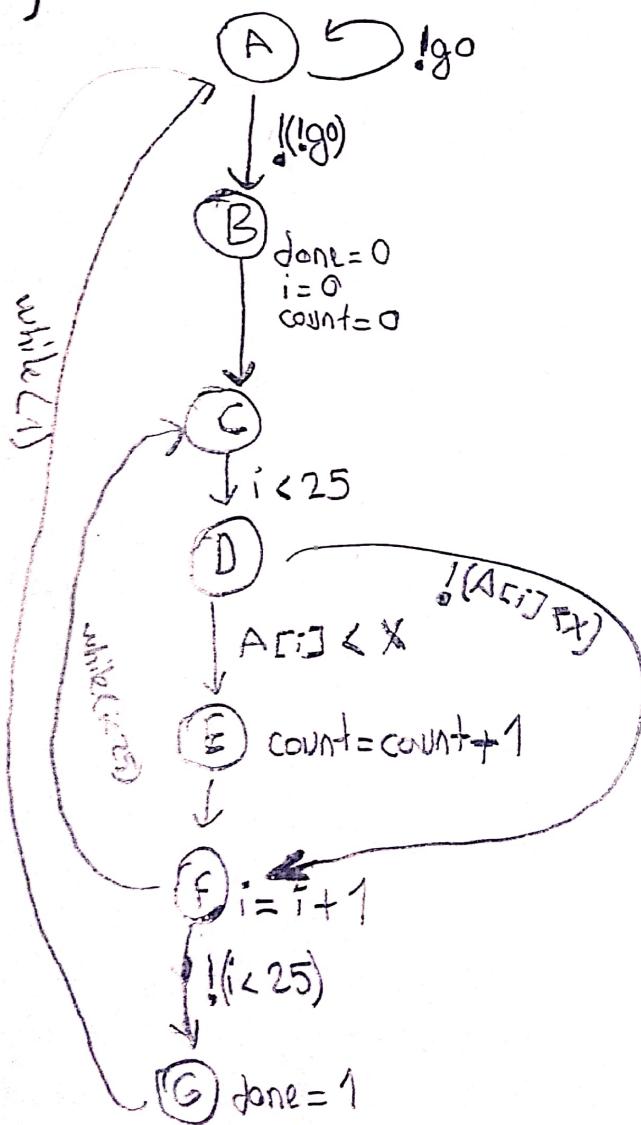


3) Convert the C code, which counts the number of elements less than x in array A consisting of 25 8-bit values into

- a) A high level state machine
- b) Use the HLSM to create datapath
- c) Connect datapath to controller.
- d) Create controllers FSMs.

a)



Inputs: byte A[25], byte x,
bit go

Outputs: byte count, bit done

Frequency:

while(1){

 while(!go);

 done = 0;

 i = 0;

 count = 0;

 while (i < 25){

 if (A[i] < x){

 count = count + 1;

 }

 i = i + 1;

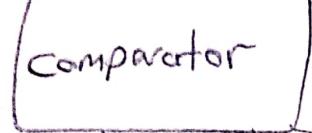
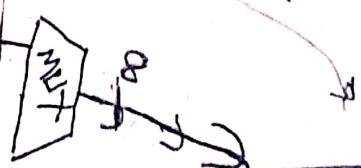
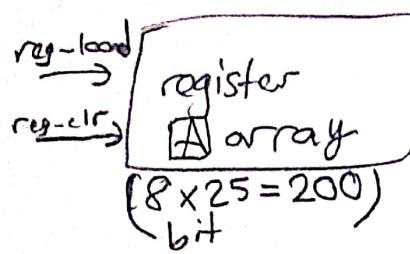
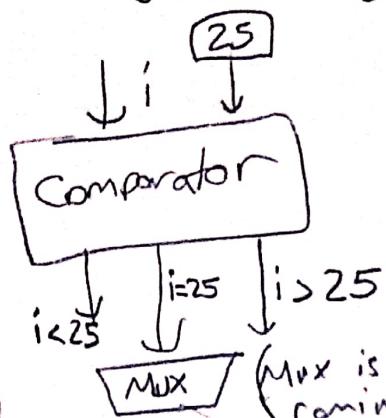
 }

b) We need:

- 2 comparators ($i < 25$ and $AC[i] < X$)
- 2 incrementers (count $\leftarrow 1$ and $i = i + 1$) [Adders]
- 4 registers (A array, count, i and X)

00011001 (Binary form)

$done = 0$
 $i = 0$
 $count = 0$

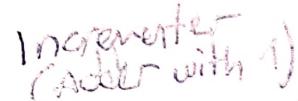
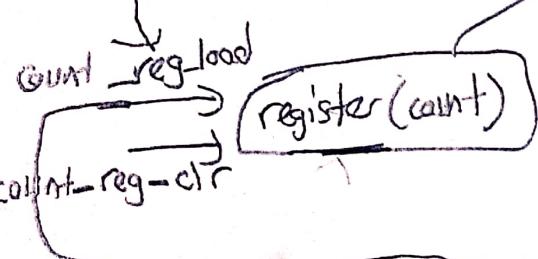


$AC[i] < X$

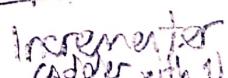
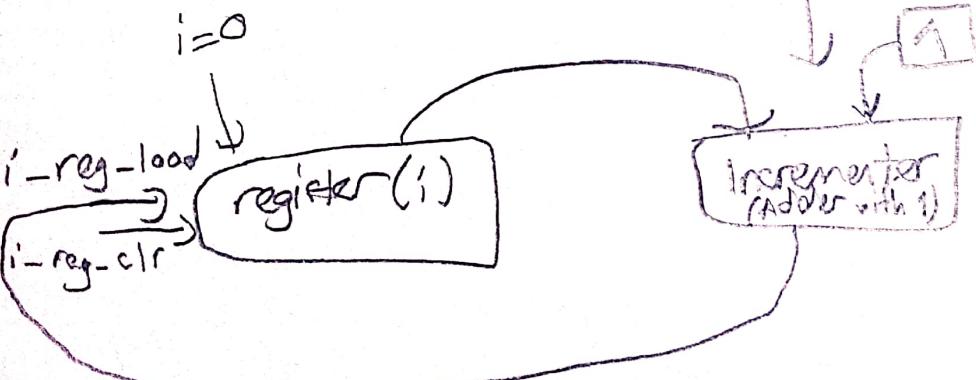
$AC[i] = X$

$AC[i] > X$

$count = 0$



$i = 0$



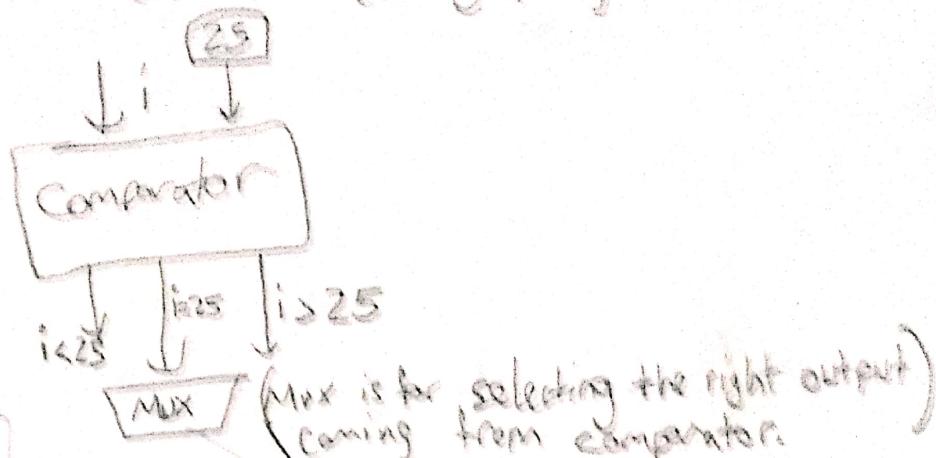
Sakshi Patel
10011000
GATE Final
Group S
26.

b) We need:

- 2 comparators ($i < 25$ and $A[i] < X$)
- 2 incrementers (count + 1 and $i = i + 1$) [Adders]
- 4 registers (A array, count, i and X)

00011001 (Binary form)

$done = 0$
 $i = 0$
 $count = 0$



reg_load
 reg_clr
register (A array)
($8 \times 25 = 200$)
bit

8 bits
register (X)

Comparator 3: $A[i] < X$

Comparator 4: $A[i] = X$

Comparator 5: $A[i] > X$

count = 0

Count reg load
Count reg clr

register (count)

Incrementer (Adder with 1)

$i = 0$

$i = reg_load$
 $i = reg_clr$
register (i)

Incrementer (Adder with 1)

c)

