Sum of Absolute Differences (SAD)

SAD algorithm

Inputs: A, B, 256 8-bit elements ; go (bit) ; Outputs: sad (32 bits)

1, Wait for 'go'

2, Initialize 'sum' and 'index' to zero

3, Check if 'done' ($i >= 256$)

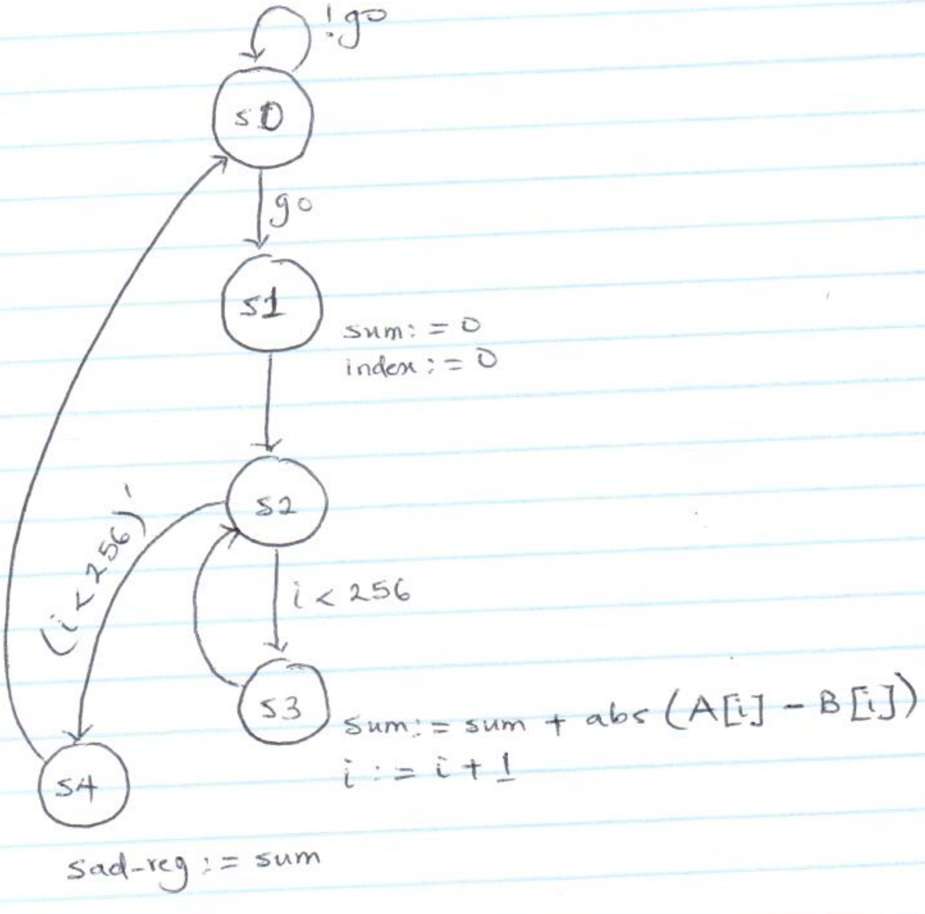
4, Add difference to 'sum', increment 'index'

5, If done, write to output, 'sad_reg'.

Inputs: A, B [256] (8 bits), go (bit)

Outputs: sad (32 bits)

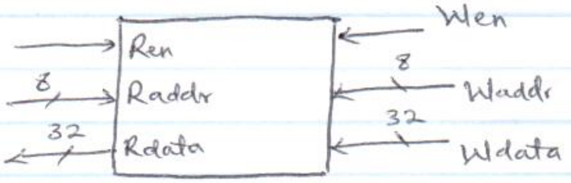
Local storage : sum, sad_reg (32 bits) ; i (9 bits)



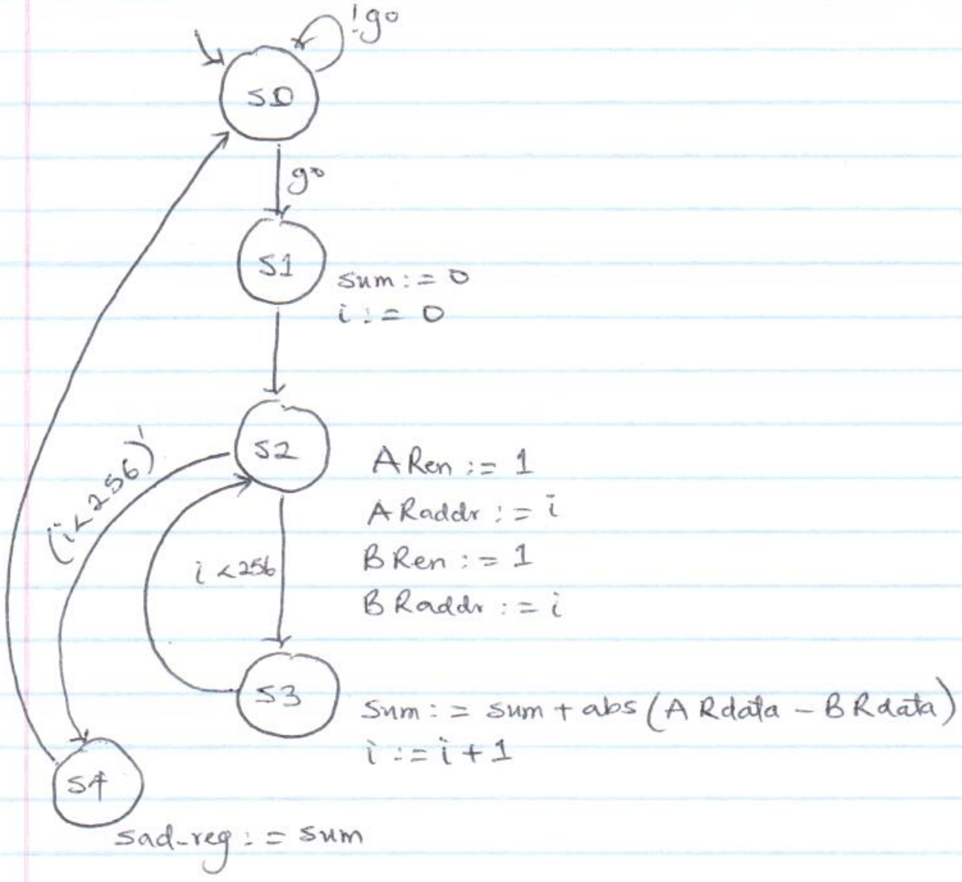
HLSM with memory (SAD HLSM)

* Create datapath + controller

Assume memories store A and B



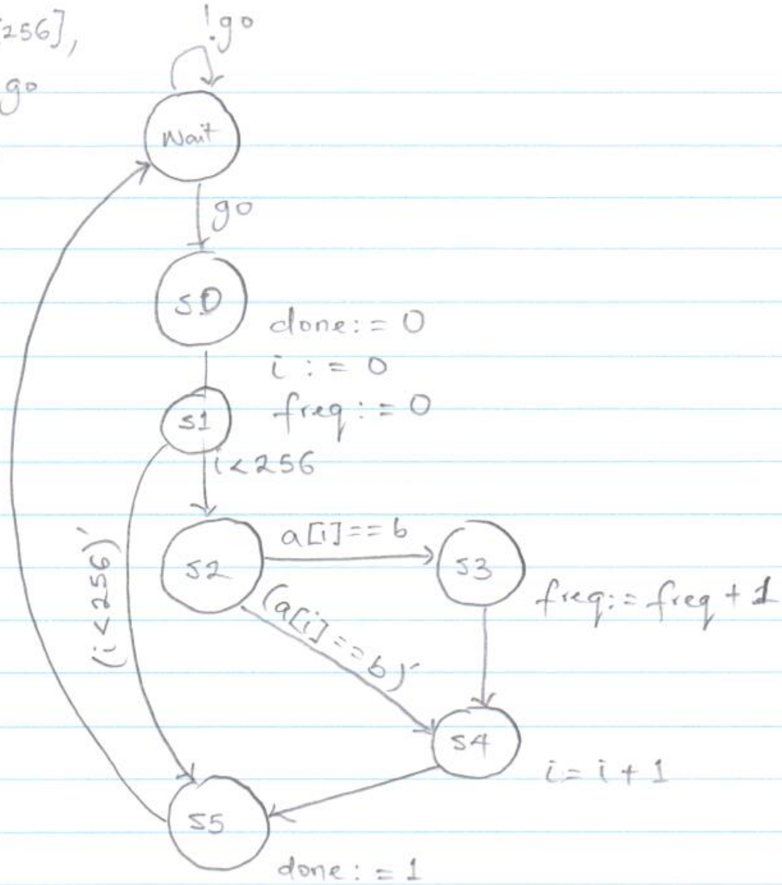
Assume synchronous read port



Exercise 1

Inputs: byte a[256],
byte b, bit go

Outputs: byte freq
bit done


!go
Wait
go
s0
done := 0
i := 0
freq := 0
s1
i<256
(i<256)'
s2
a[i]==b
s3
freq := freq + 1
(a[i]==b)'
s4
i = i + 1
s5
done := 1