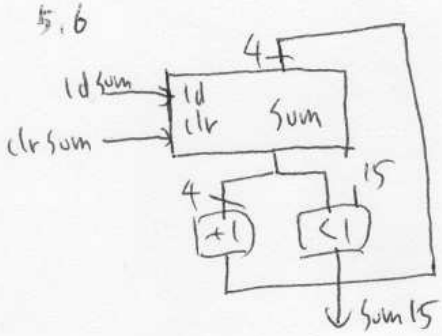
5.6




$s_1$ $s_0$

init : 0 0

count : 0 1

idle : 1 0

TC : 1 1

← define state

$n_1 = (s_1 + s_0)\,cnt'\,clr' + s_1'\,s_0\,cnt\,clr'\,sum15$

$n_0 = s_1'\,s_0'\,cnt + (s_1 + s_0)\,cnt\,clr'$

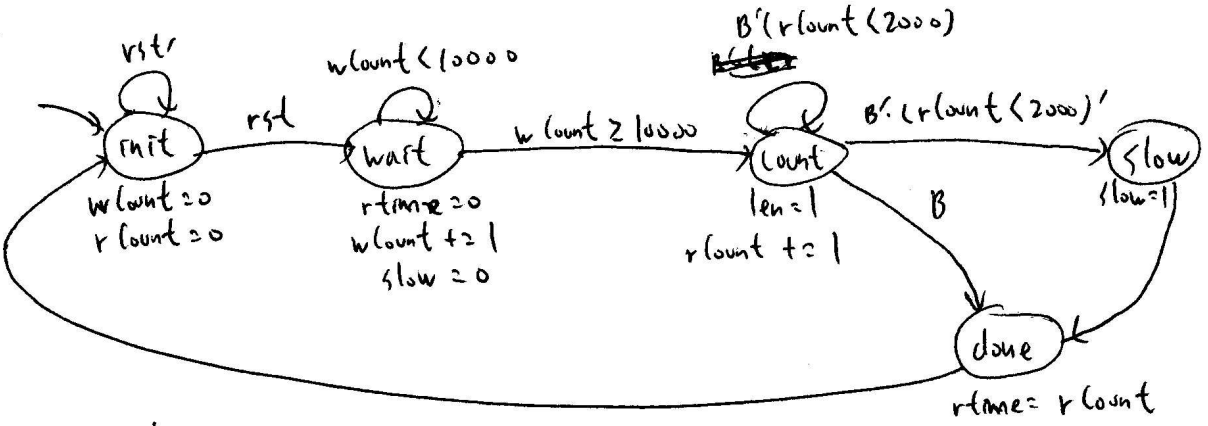
$t_c = s_1\,s_0$

$ldSum = s_1'\,s_0$

$clrSum = s_1'\,s_0' + s_1\,s_0$

| $s_1$ | $s_0$ | cnt | clr | sum15 | $n_1$ | $n_0$ | $t_c$ | ldSum | clrSum |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

5. 12




inputs : clk, rst. B

outputs : len, slow, rtime

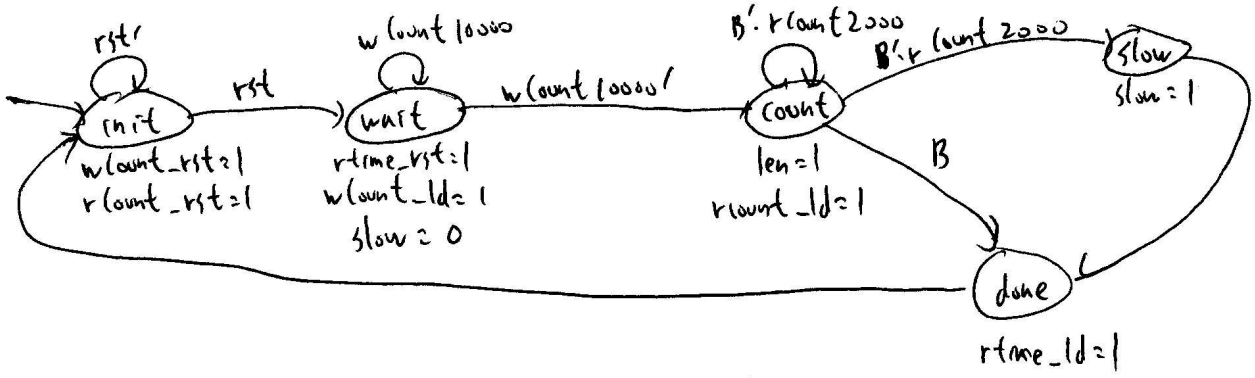
registers: wCount, rCount.

↑ High level

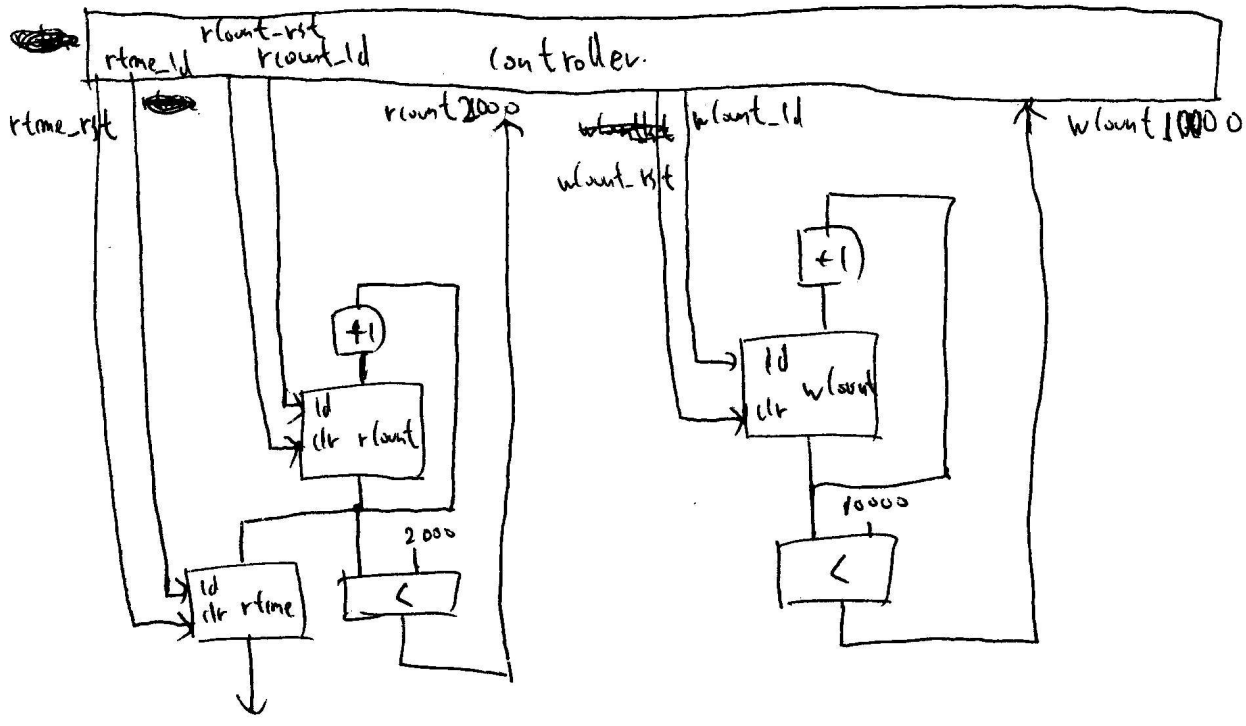
---

inputs : clr, rst, B, rCount2000, wCount10000

outputs : len, slow, wCount_rst, rCount_rst, rTime_rst, wCount_ld, rCount_ld, rtime_ld.




↑ FSM. controller.



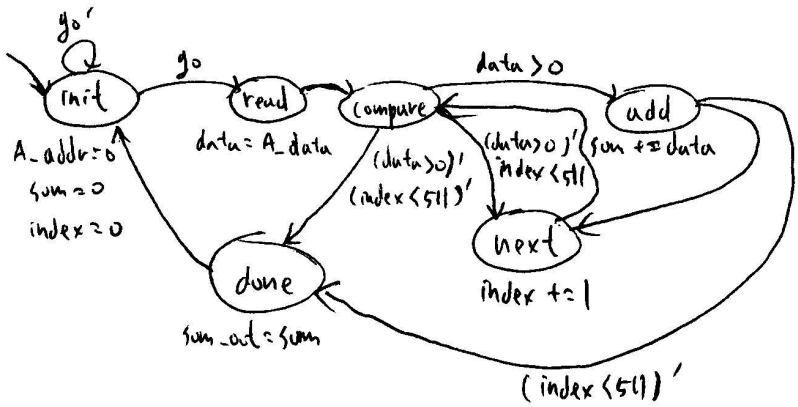
5.14

Step 1.

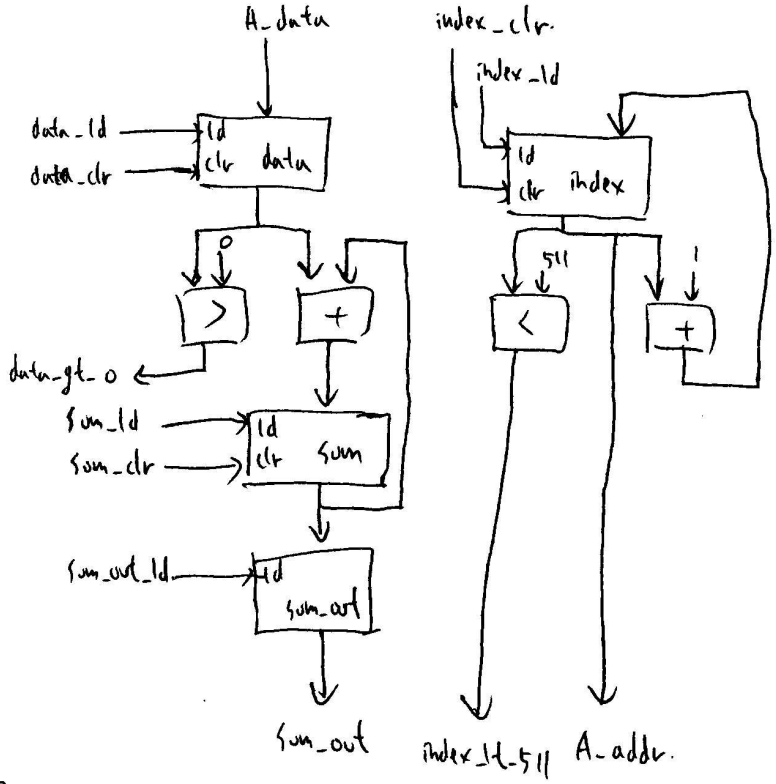
inputs: go, A_data.

outputs: A_addr, sum_out

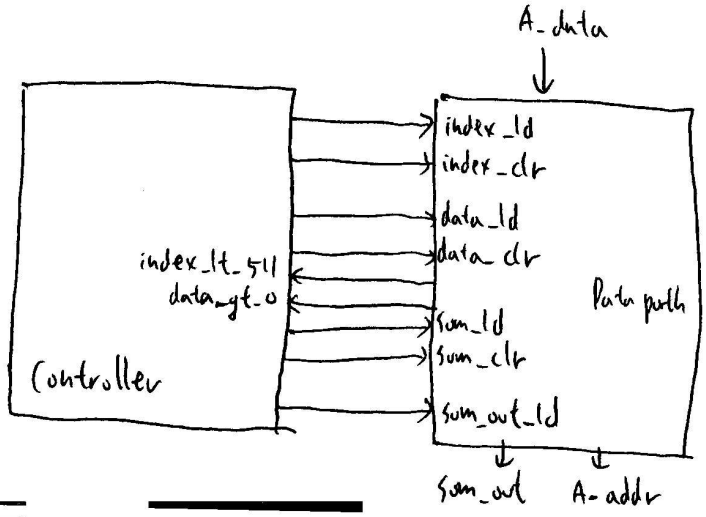
registers: sum, index, data.




Step 2.




Step 3.



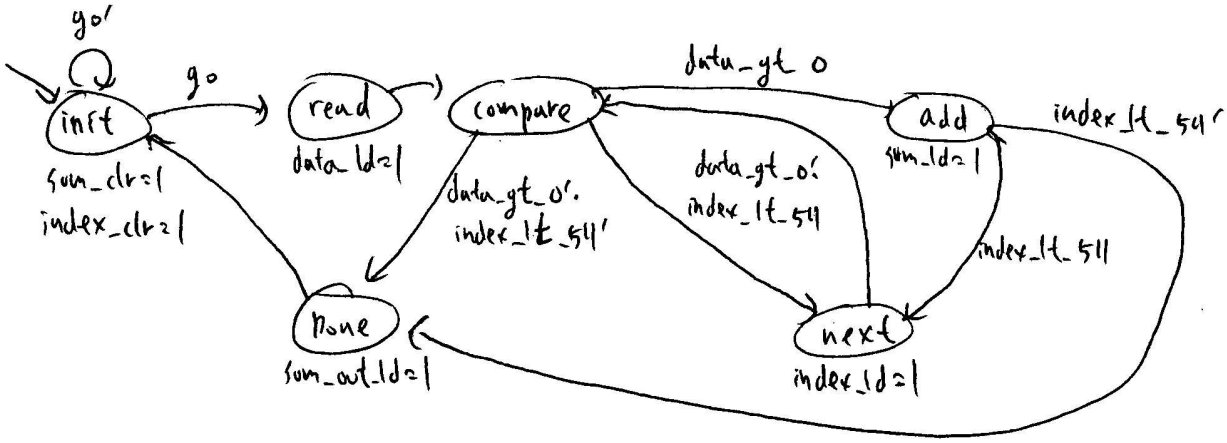
5.14

Step 4.

Inputs: go, data_gt_0, index_lt_511

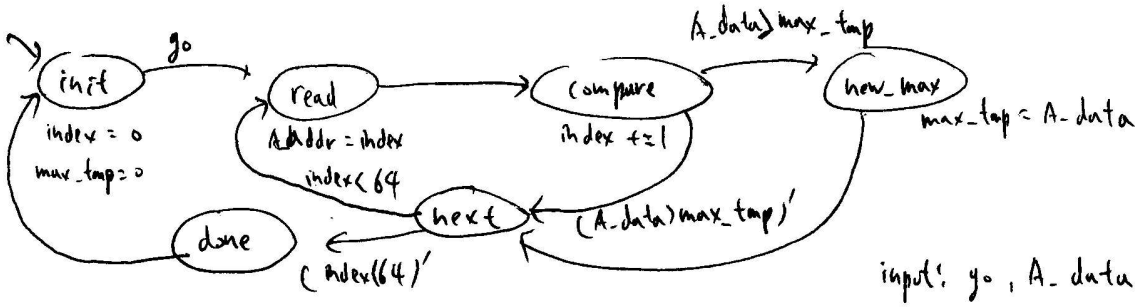
Outputs: sum_clr, sum_ld, index_clr, index_ld, data_ld, sum_out_ld

registers: sum, index, data




5.16

Step 1.

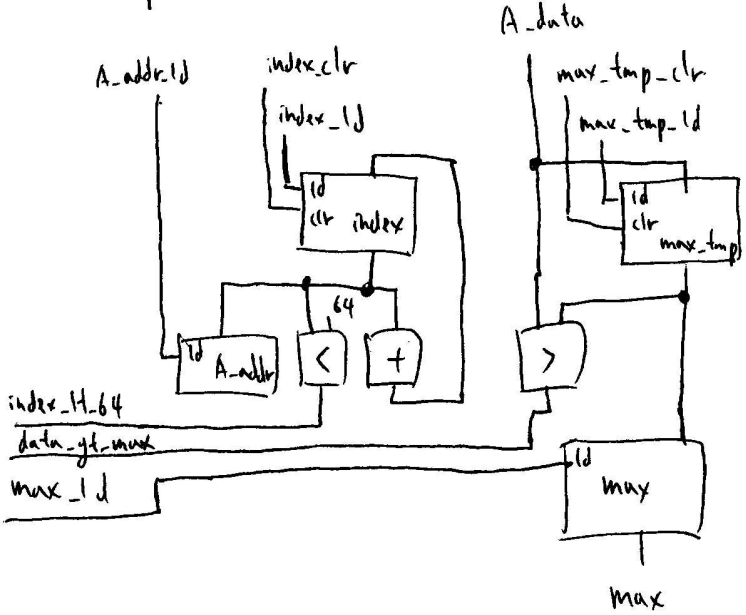



input: go, A_data

output: A_addr, max

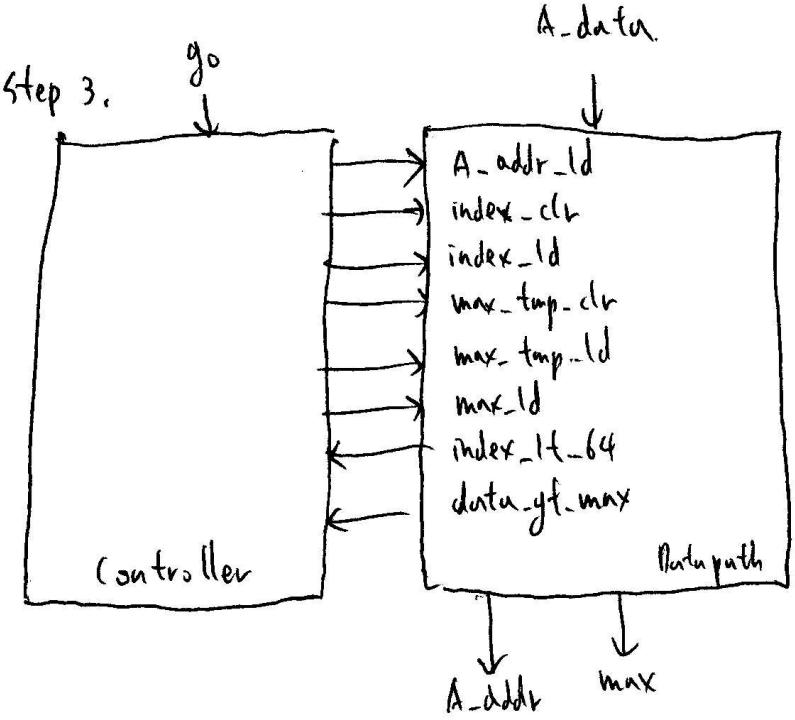
registers: max_tmp, index

Step 2.



5.16

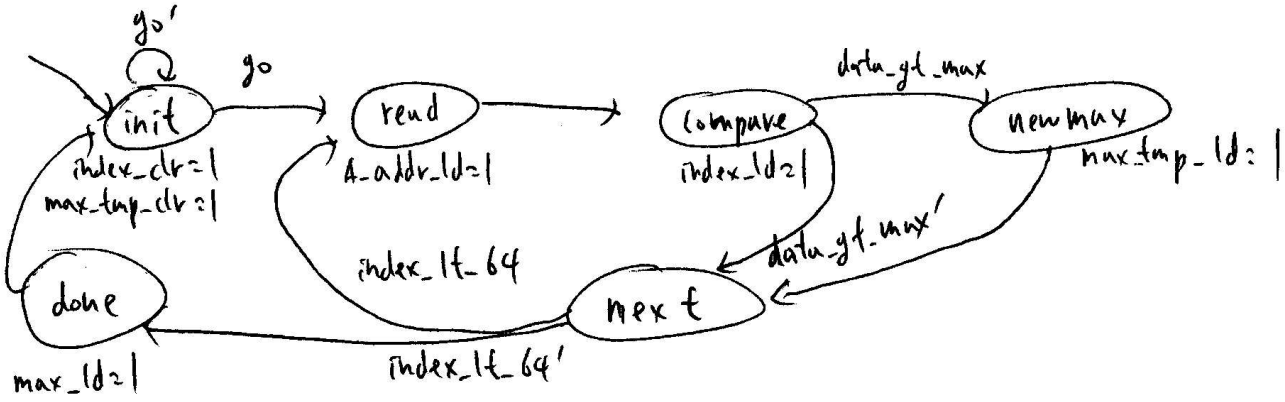
Step 3.




Step 4.

inputs: go, index_lt_64, data_gt_max

outputs: A-addr_ld, index_clr, index_ld, max_tmp_clr, max_tmp_ld, max_ld.

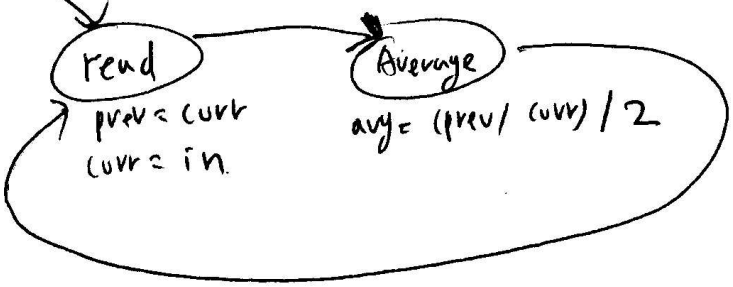



5.18. Step 1.

inputs: in

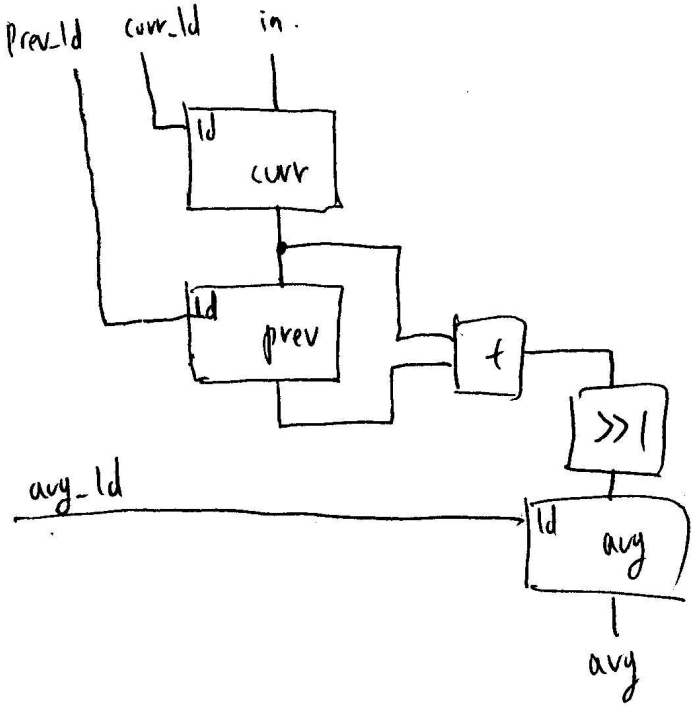
Outputs: avg

registers: curr, prev.

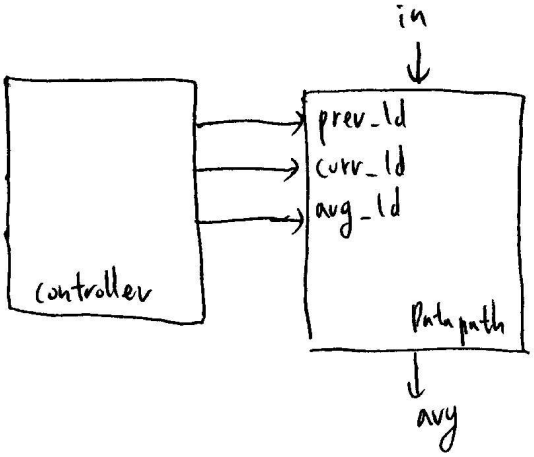


5.18.

Step 2.



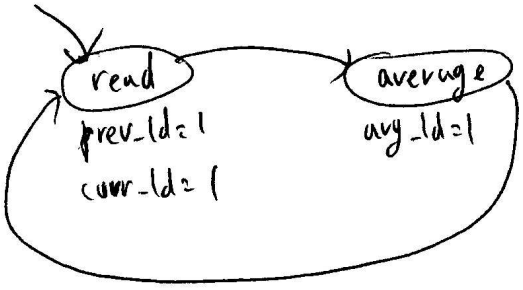

Step 3.




Step 4.

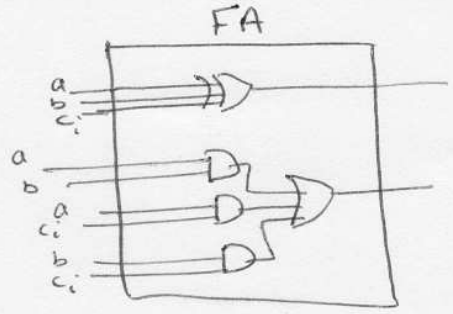
input : none

outputs : curr_ld, prev_ld, avg_ld



5.22

(a)




For 1 FA :

Critical path delay: 4 ns

For 8 => 32 ns

(b)

Each FA has one internal wire => 8 * 1 ns

Between each FA is also a wire => 7 * 1 ns

First $c_i$ + Last $C_o$ = 1 + 1 = 2 ns

17 ns

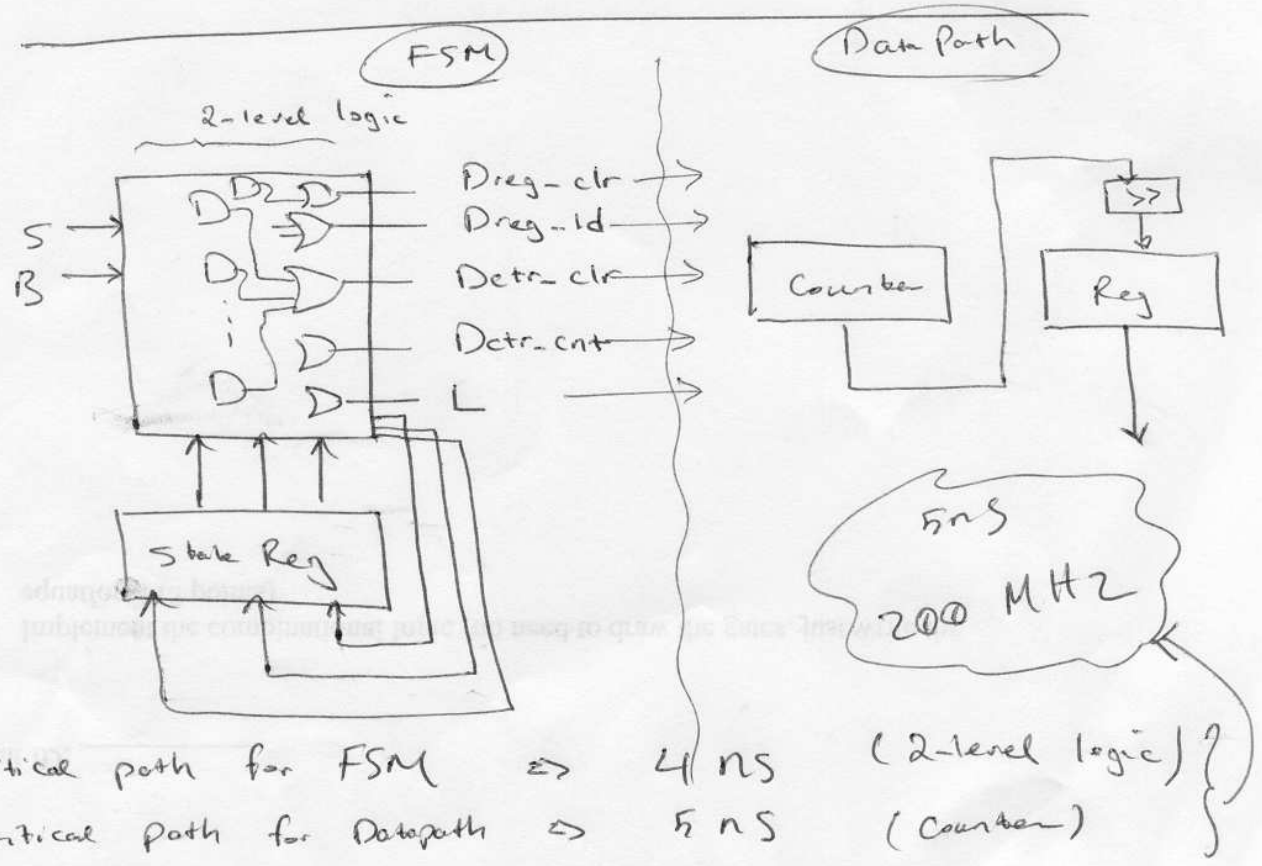
wire delay

32 ns + 17 ns = 49 ns

5.23

(b)

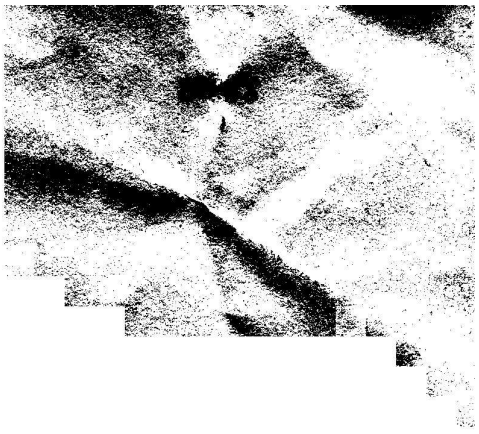
(c)

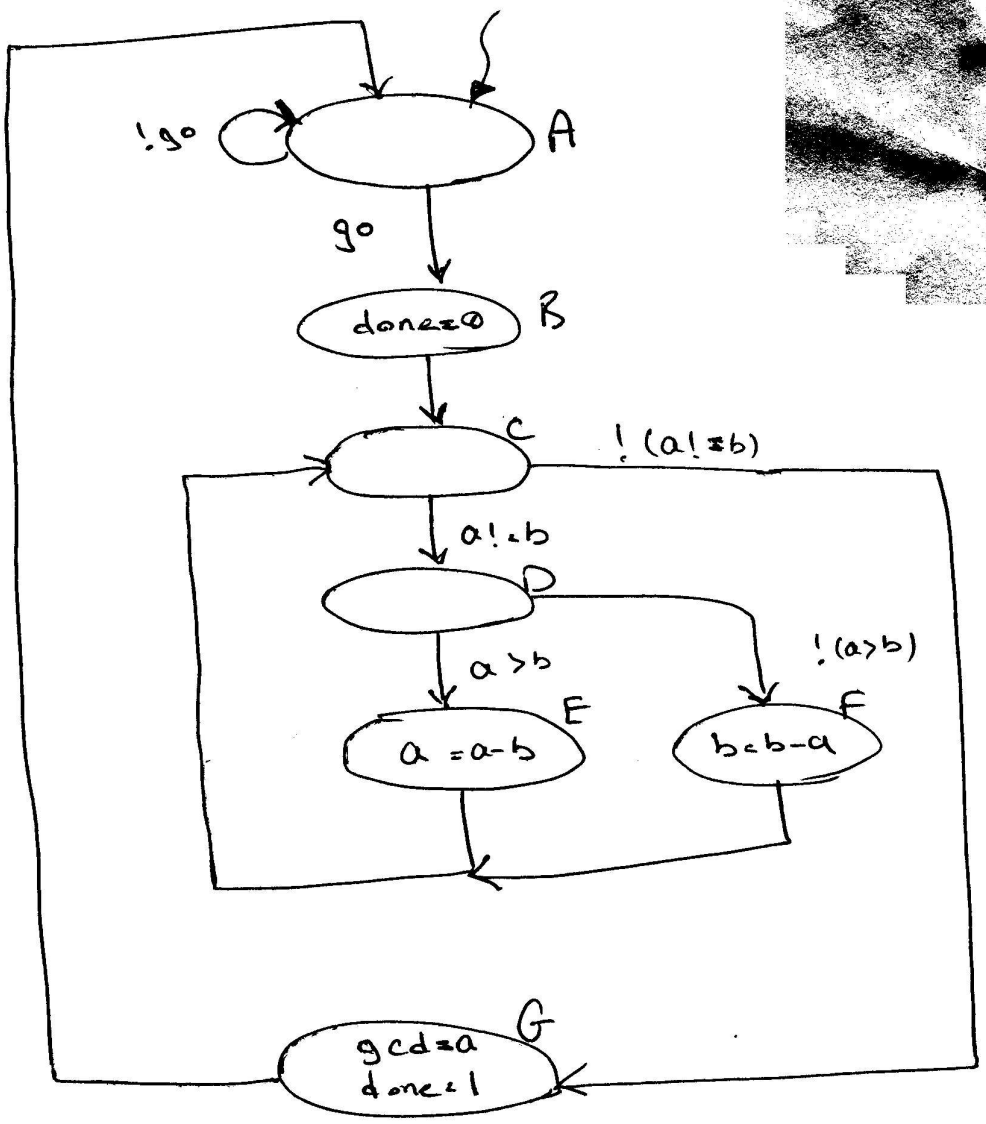



Critical path for FSM => 4 ns (2-level logic)

Critical path for Datapath => 5 ns (Counter)

5.24






5.25

Step 1: Capture the high-level state machine:

Refer to 5.24

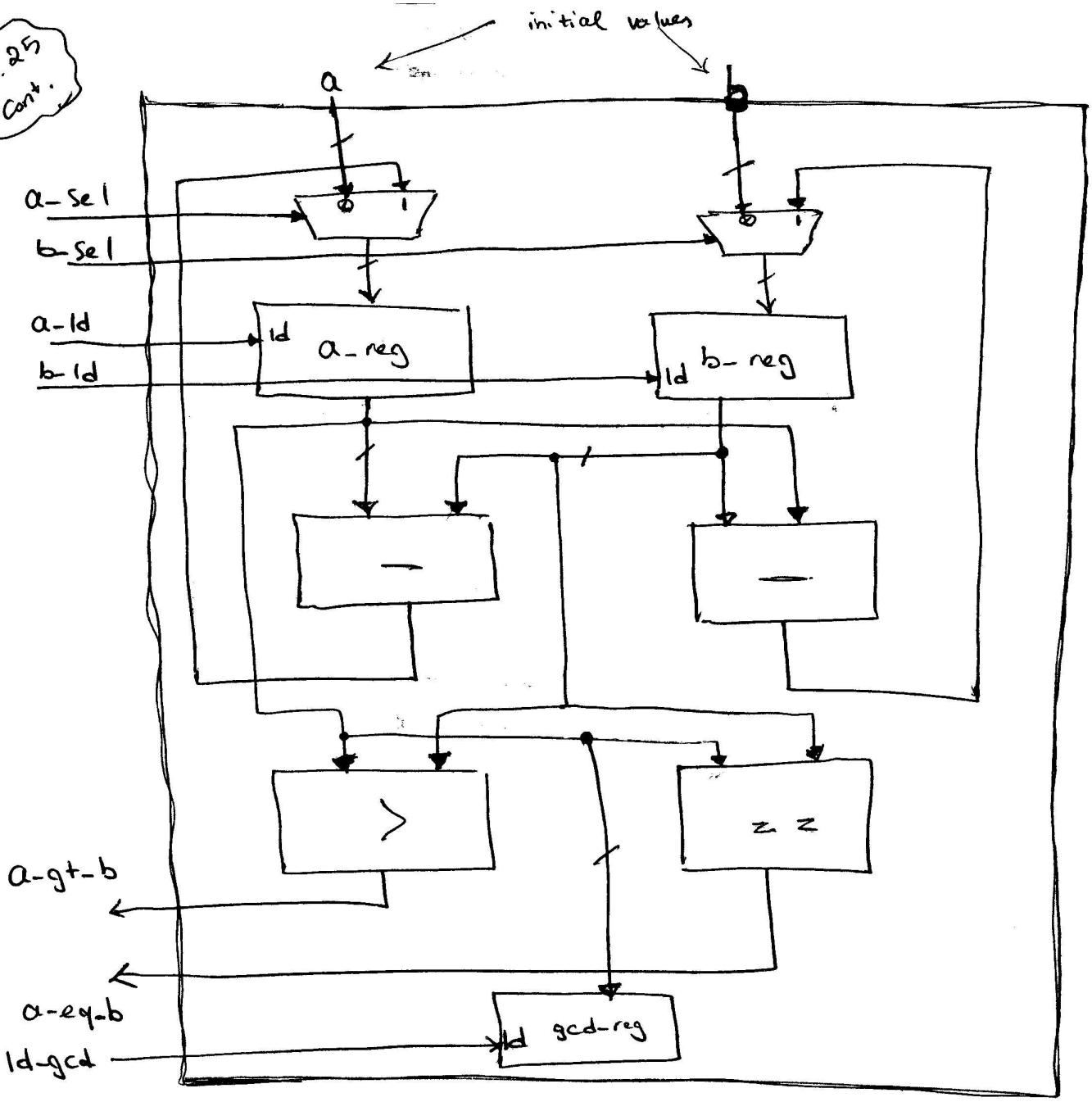
Step 2: Create a datapath

Assume No Sharing:

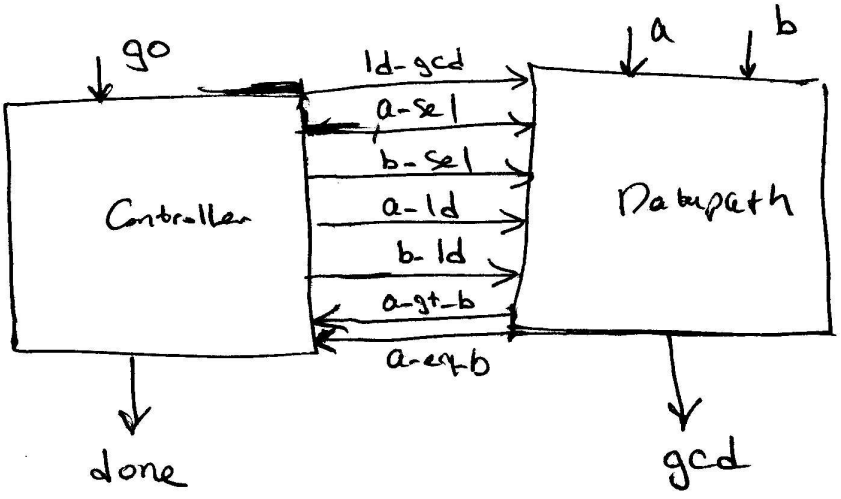
What we need =>

| | |
|---|---|
| 3 | registers for a, b, gcd |
| 2 | Subtraction |
| 1 | > |
| 1 | == |

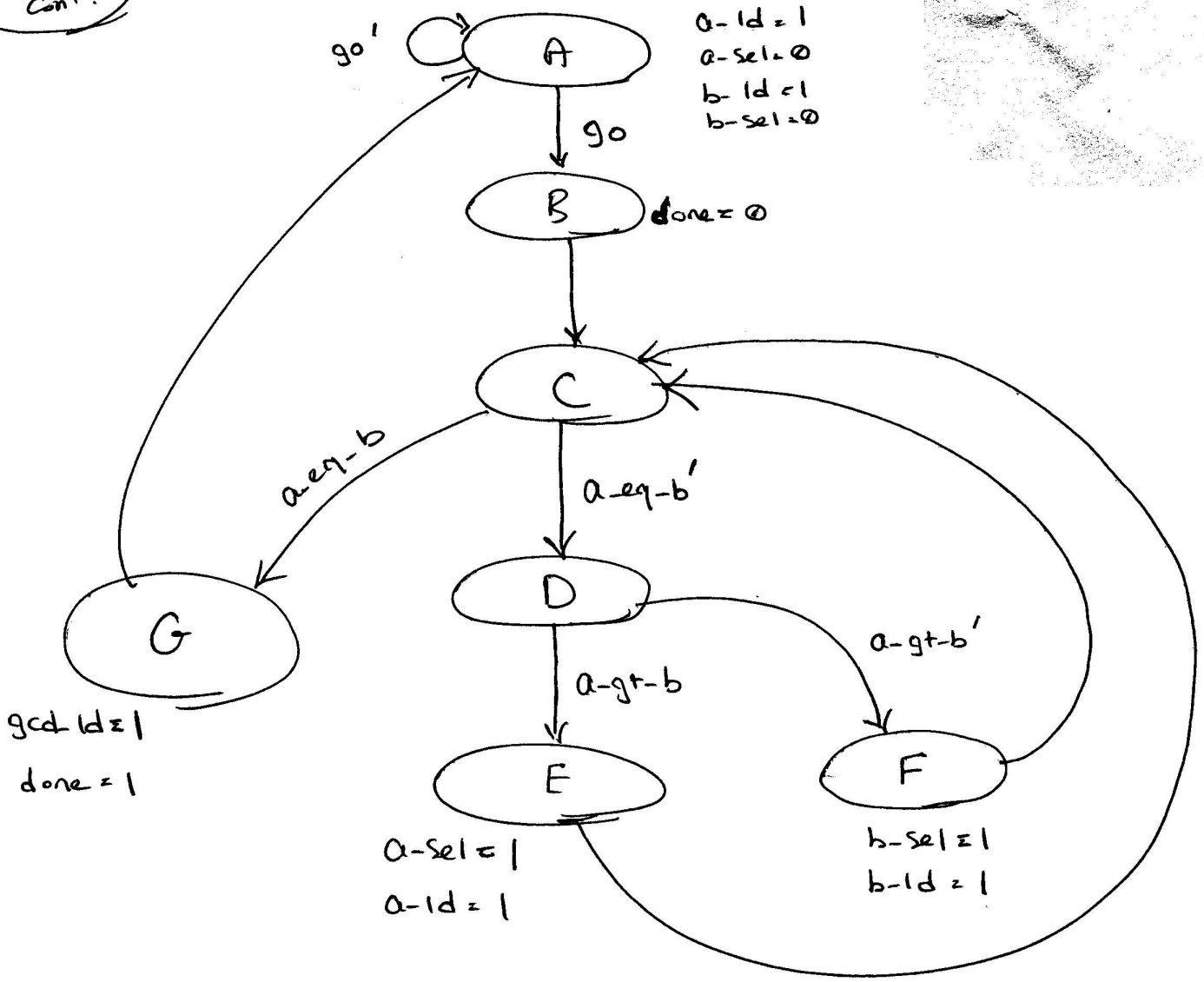
5.25 Cont.



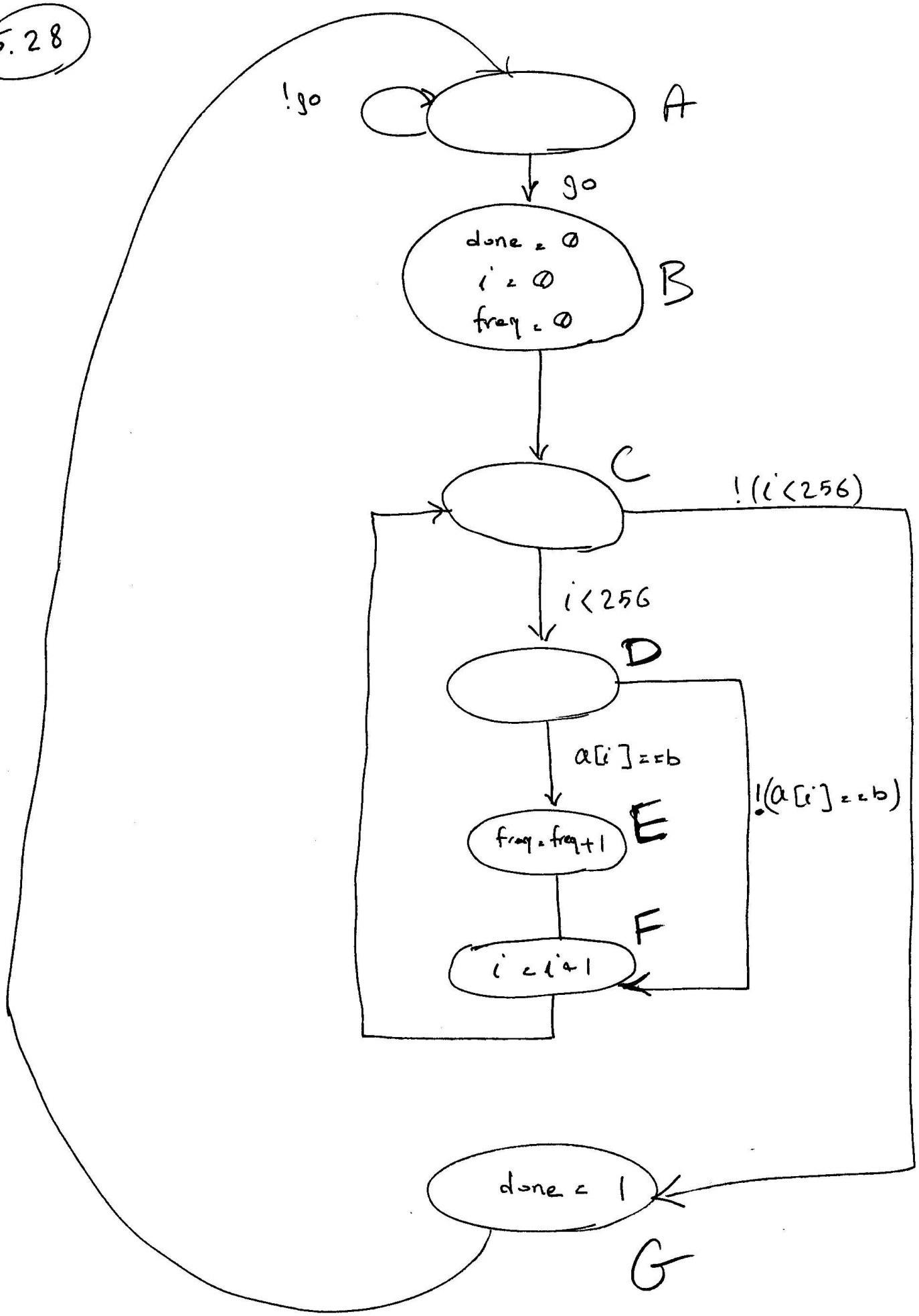

Step 3 Connect datapath to FSM



5.25 cont.


go'
A
a-ld = 1
a-sel = 0
b-ld = 1
b-sel = 0
go
B
done = 0
C
a_eq_b
a_eq_b'
D
G
gcd_ld = 1
done = 1
a_gt_b
a_gt_b'
E
F
a-sel = 1
a-ld = 1
b-sel = 1
b-ld = 1

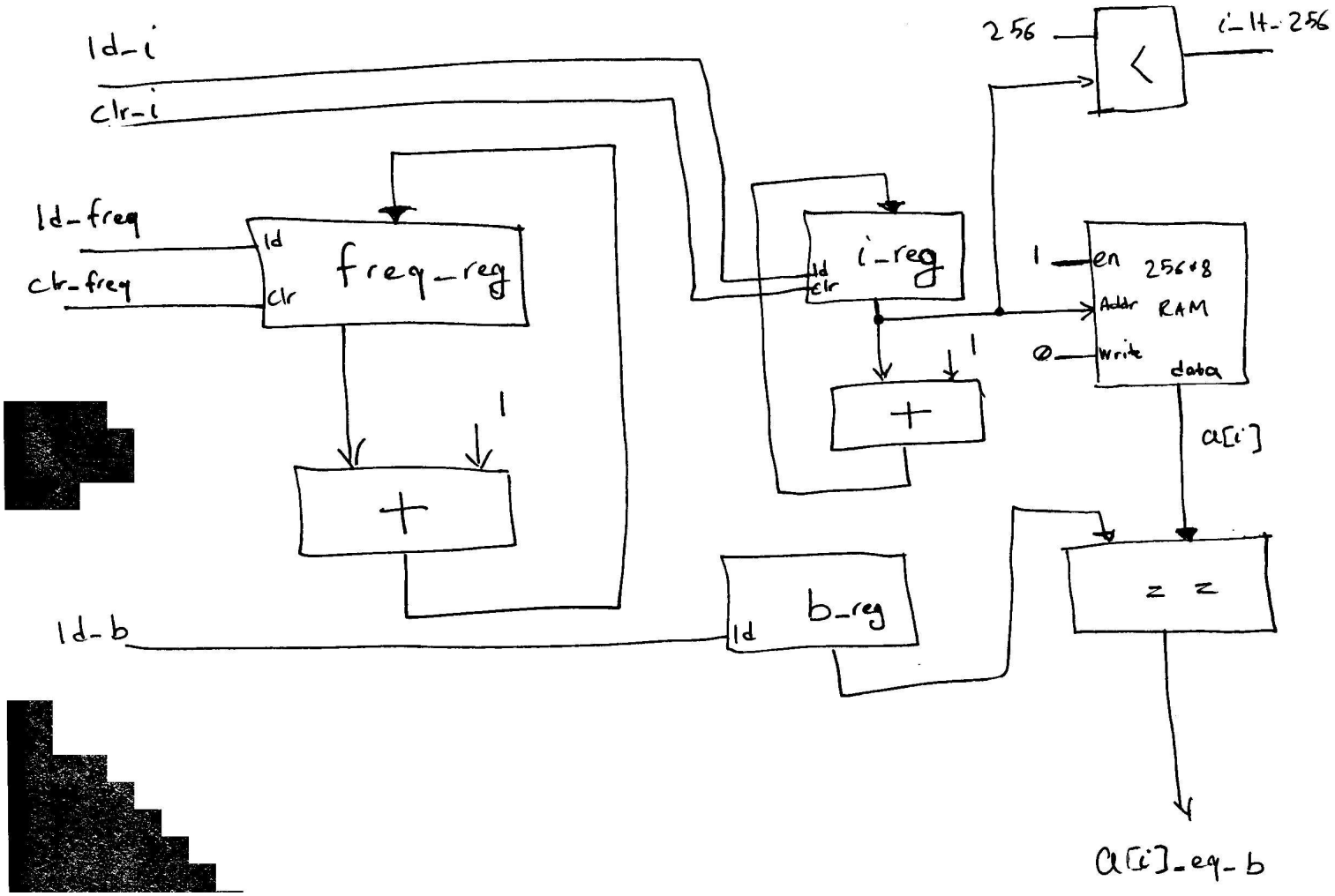
5.28


!go
A
go
done = 0
i = 0
freq = 0
B
C
!(i<256)
i<256
D
a[i]==b
!(a[i]==b)
freq = freq+1
E
F
i = i+1
done = 1
G

5.29 Step 1 : FSMD => 5.28 solution

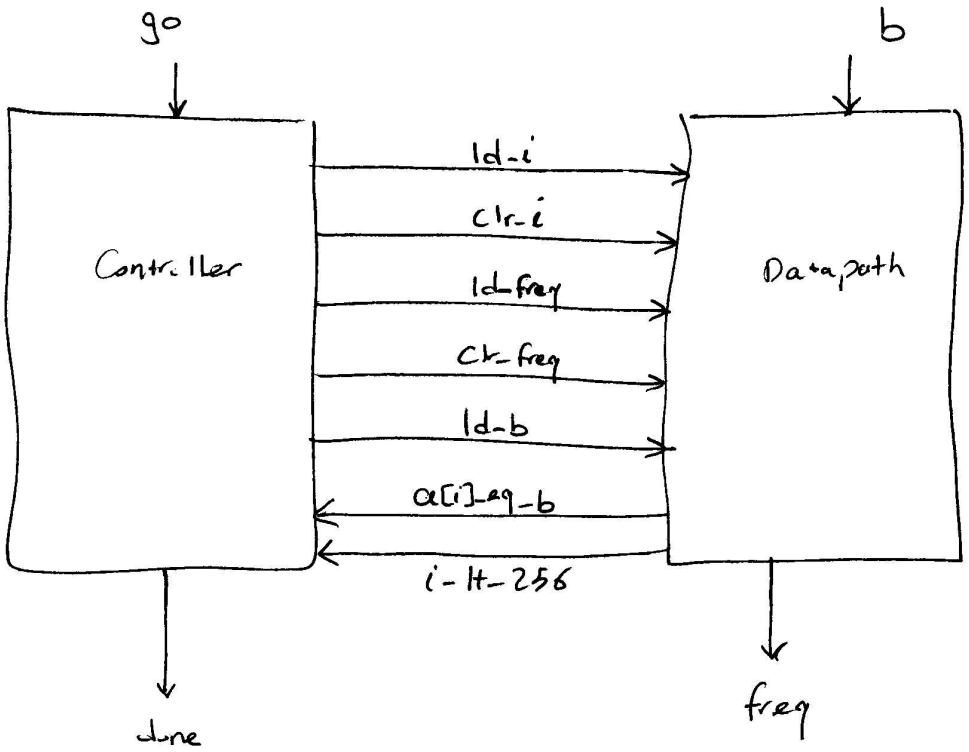
Step 2 : Datapath

what we need
- 3 Registers for i, b, freq
- 2 Adders for incrementing i, freq
- 1 (==) for (a[i]==b)
- 1 RAM 256*8 for a[i]
- 1 (<) for (i<256)



5.29 Cont.

Step 3 : Datapath Controller connection




Step 4

FSM :