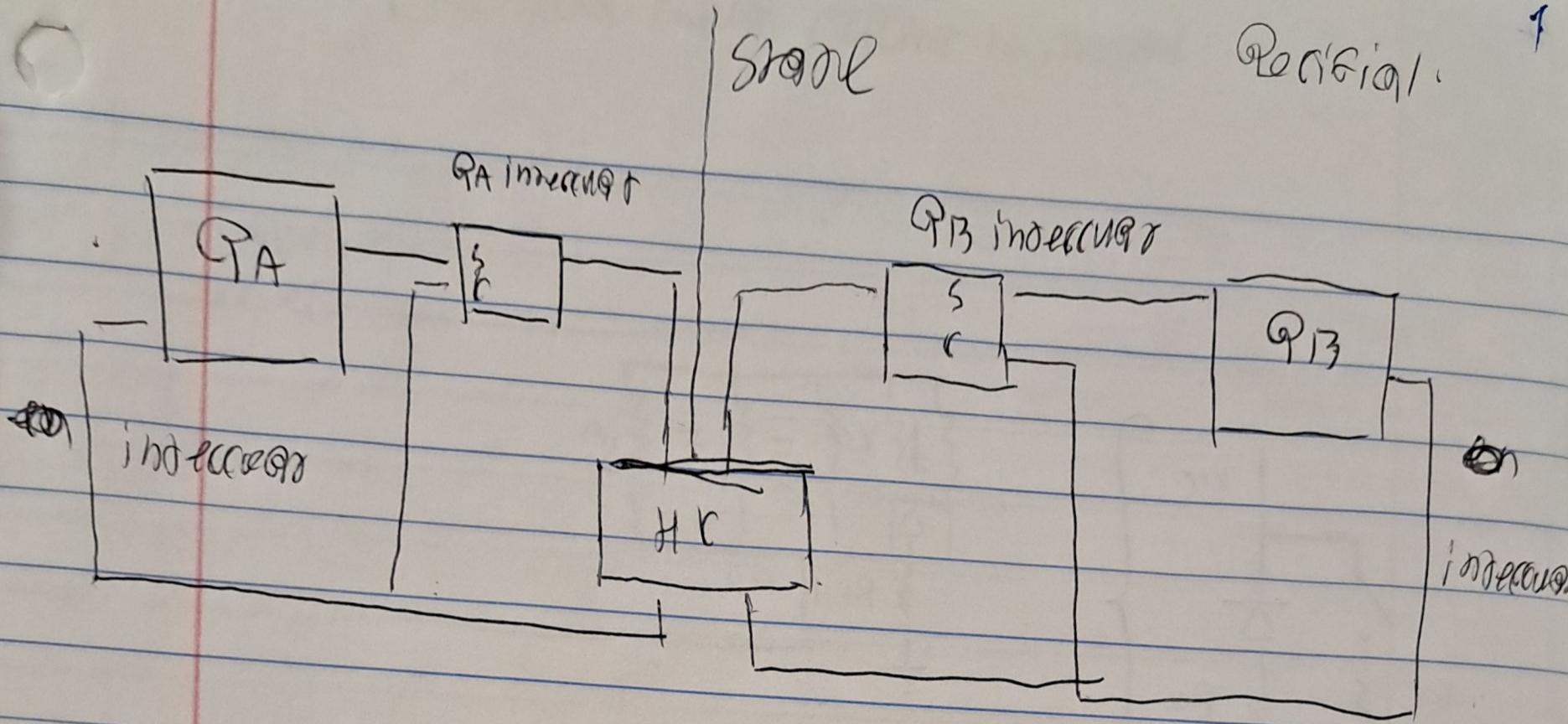


# 16 bit Computer Journal | T.O.C.

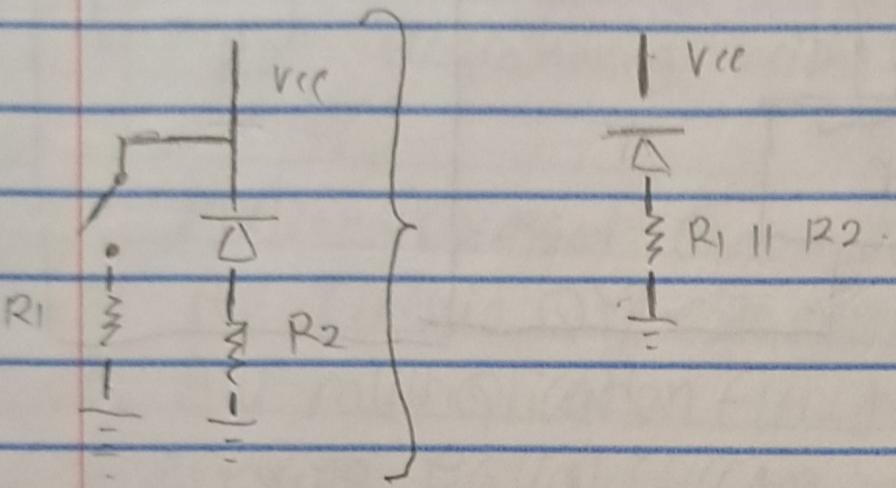
- > PCB Modular design Plan 13
- > User Control Input Schematic 6-7
- > Input module blueprint 8-9.
- > Multi Programming with Peripheral device 10.
- > Multi-Processor Ram access system idea 1-5
- > AIU circuitry & programming 19
- > AIU multiplication function design. 22-24
- > keeping equality check in modular circuit 1
- > Computer design checklist. 1, 15
- > Raspberry Pi program beginnig. (26-30)
- > Wire color coding by (30-31)
- > Ram based memory creation (36)
- > C# GUI for programming hardware (37)
- > FPGA Shopping hw 40.
- > FPGA Human interfacing beginnig. 49.
- > module display output idea 42



$Q_A$  and  $Q_B$  are in waiting instructions, they both signal requests to use the com bus. In an interruptable state, the HC chip selects the highest Priority Processor and prioritizes com access. This is done by sending an interrupt signal to the Processor which interrupt is wait instruction to a com usage instruction. This transitions to the next instruction, releases the interrupt request and processes the next processor.

This implies that all the while, the interrupt signal is active, the request latch is active, and the HC stays in the same state.

This HC chip is separate from the HC chip in the interrupt module of each Processor.

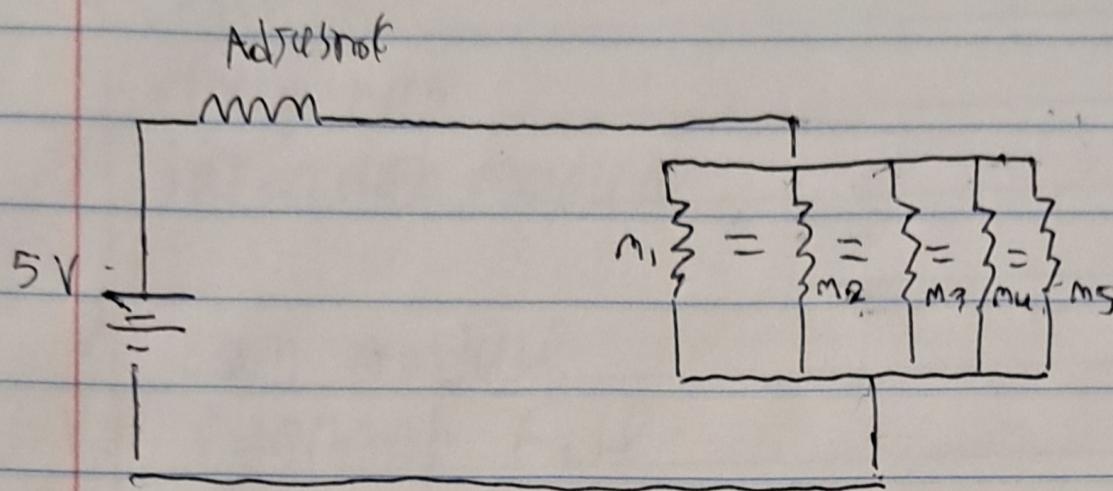


$$A) (R_1 \parallel R_2) = \frac{(R_1)(R_2)}{R_1 + R_2} / \text{for } R_1 = \emptyset / A_1 = \frac{(\emptyset)(P_2)}{\emptyset + P_2} /$$

$$B) = \frac{\emptyset}{P_2} / = \emptyset / r = \emptyset P / r = \emptyset(\emptyset) / \boxed{r = \emptyset} /$$

1)

[keeping equal current in modular circuit]

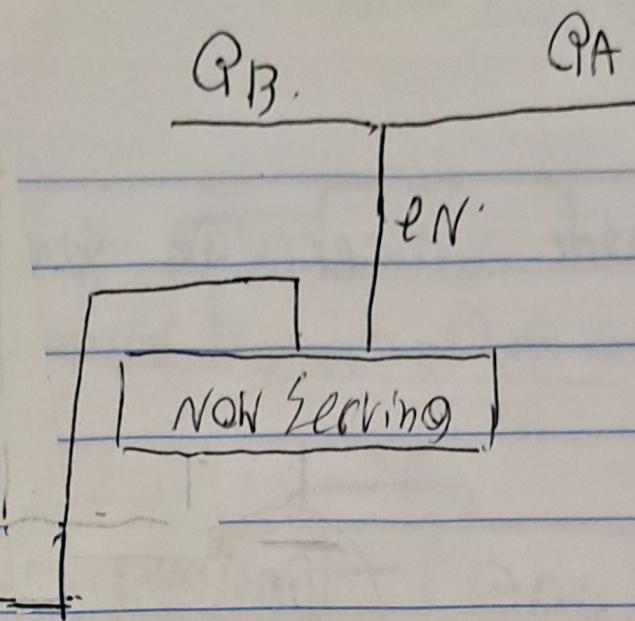
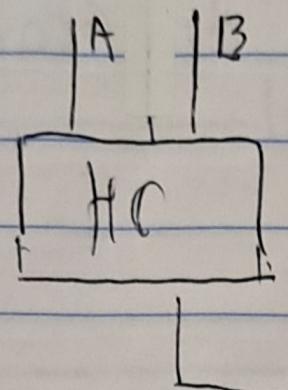


current in each module is equal as each module has equal resistance. The total current is equally divided about the modules. The Adjuster makes it so that the total resistance is great enough to provide a different current in the modules. It's possible for more than 5V will be needed.

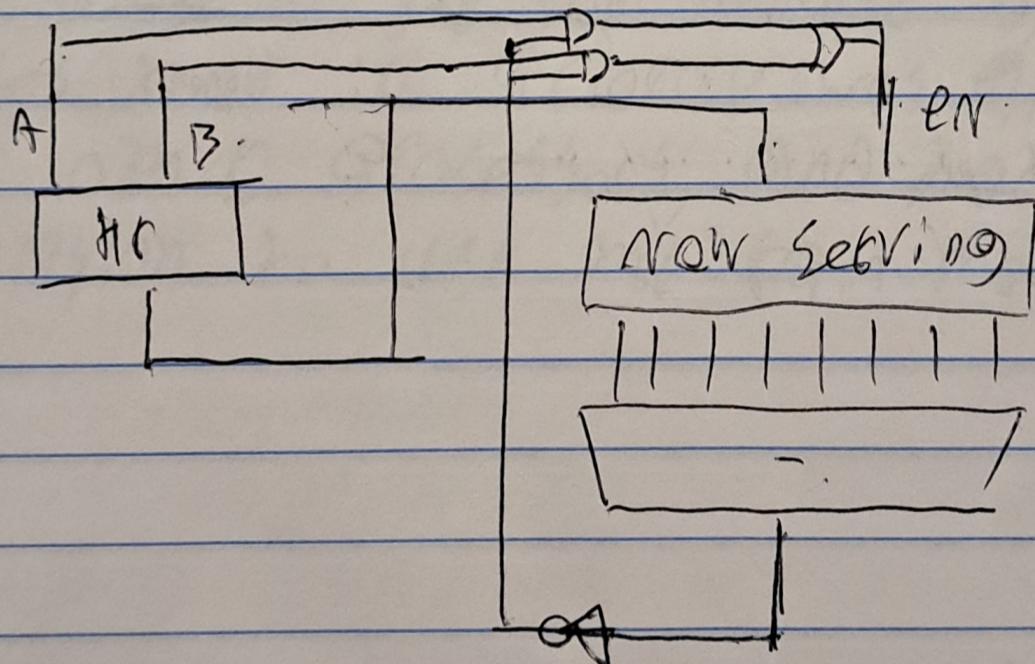
# Computer build checklist

- ✓ ALU
- ✓ data bus
- ✓ instruction module
- ✓ RAM module
- ✓ register file
- ✓ clock
- instruction reg
- program counter
- micro instruction counter
- data input module / control panel
- control module + instruction set
- Raspberry Pi Chip Programmer
- ✓ keyboard module

idea #1



Upon finishing the Prev Process (Ques),  
the Previous Process enables the new serving  
task to update the ~~task~~ next in line.



idea #2.

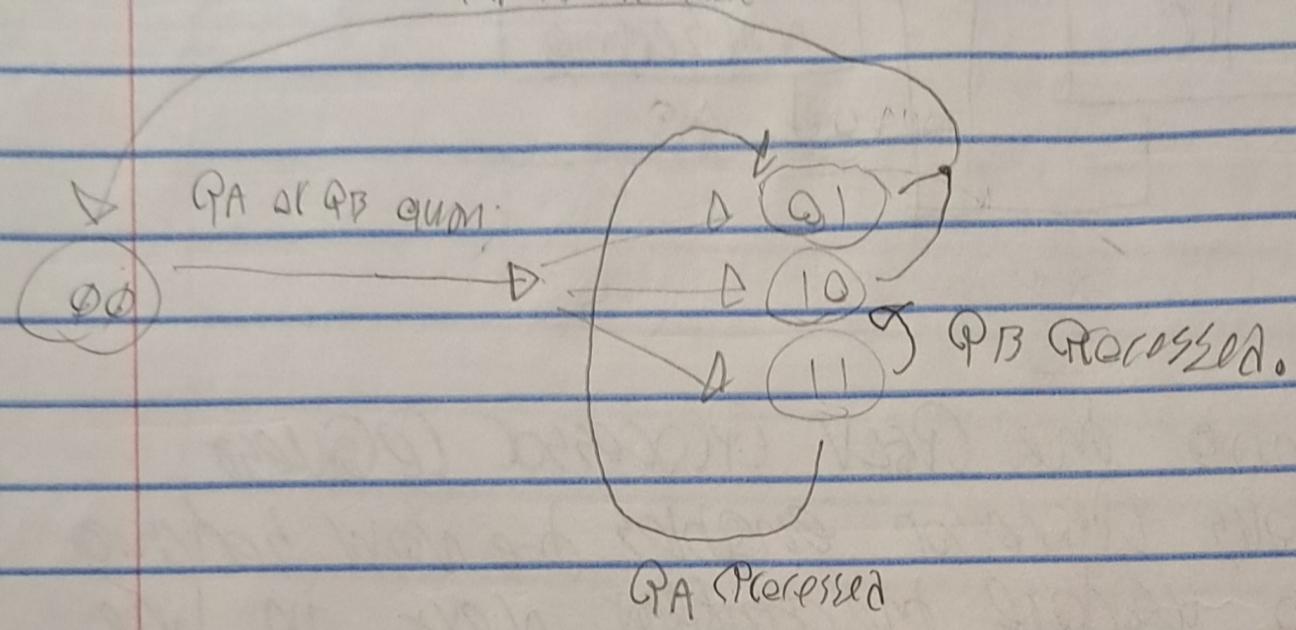
If the current chip being ~~is~~ served is not  $\emptyset$ , ~~is~~.  
~~is~~ was not active last quantum, A Process  
may request access and be served. Otherwise, the  
process is denied access.

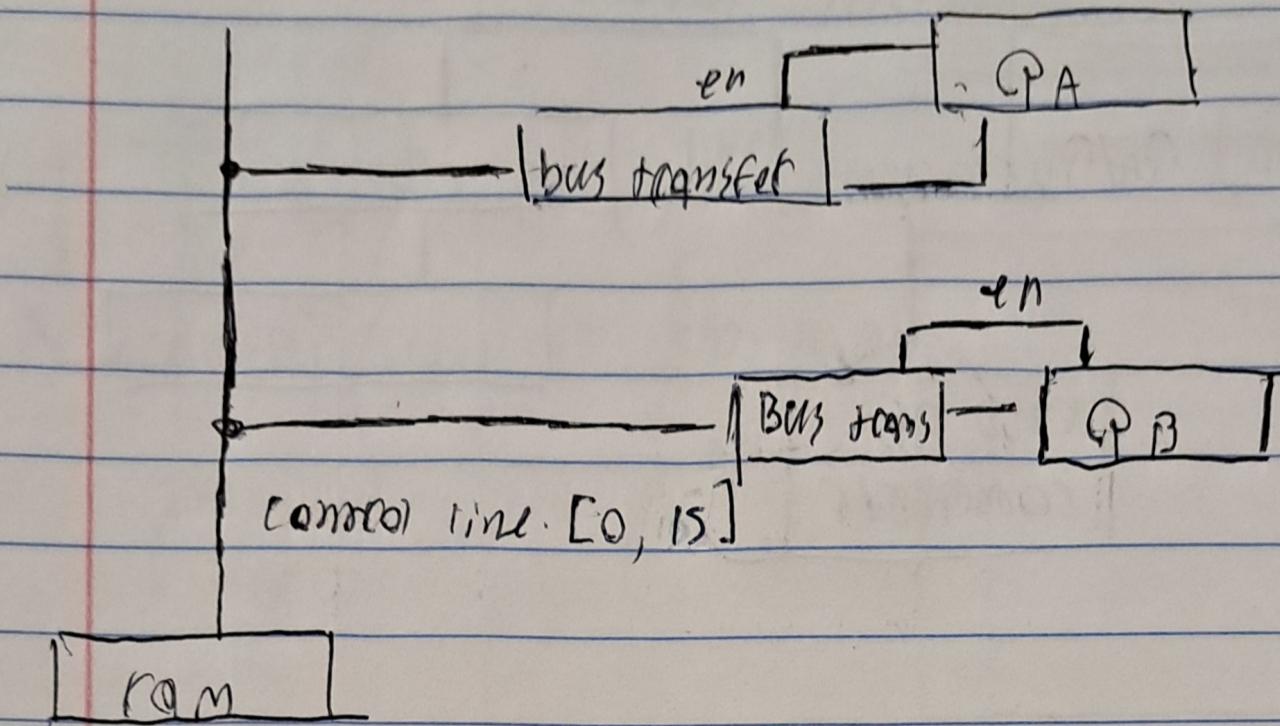
4

$Q_A, Q_B \leftarrow (stage)$   
 $\leftarrow (order)$

Current Processor interrupt stage dia 600m

$Q_A/Q_B$  process.





The usage of the ram module by any process implies that its priority was already evaluated over other processes, and that it has permission to use the ram module.

b

bus transfer

bus transfer

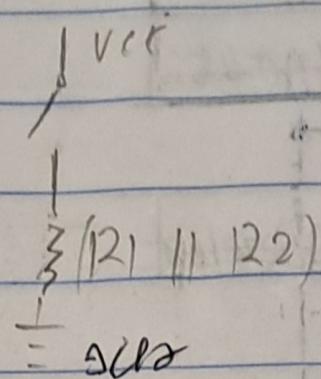
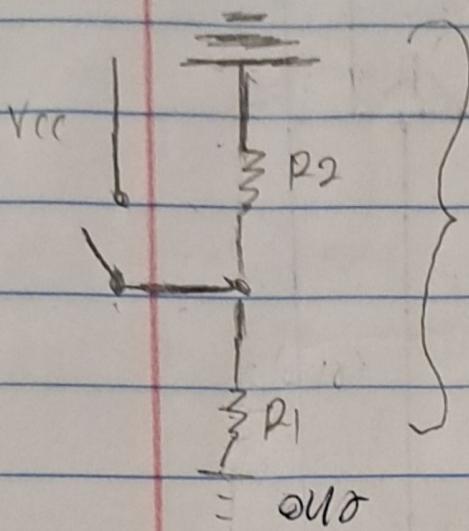
use C input

control

rest of  
computation

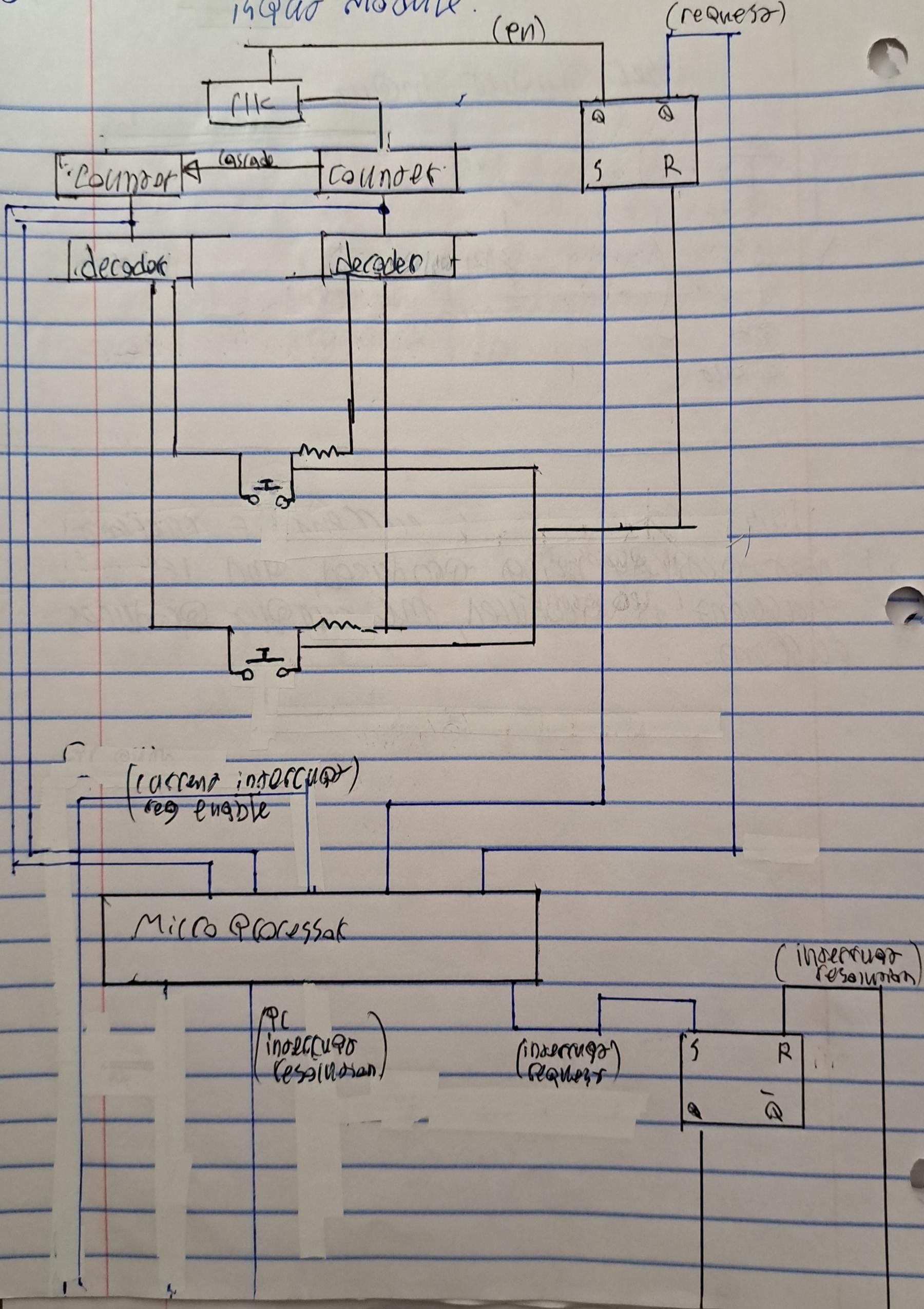
7

## Used control inputs

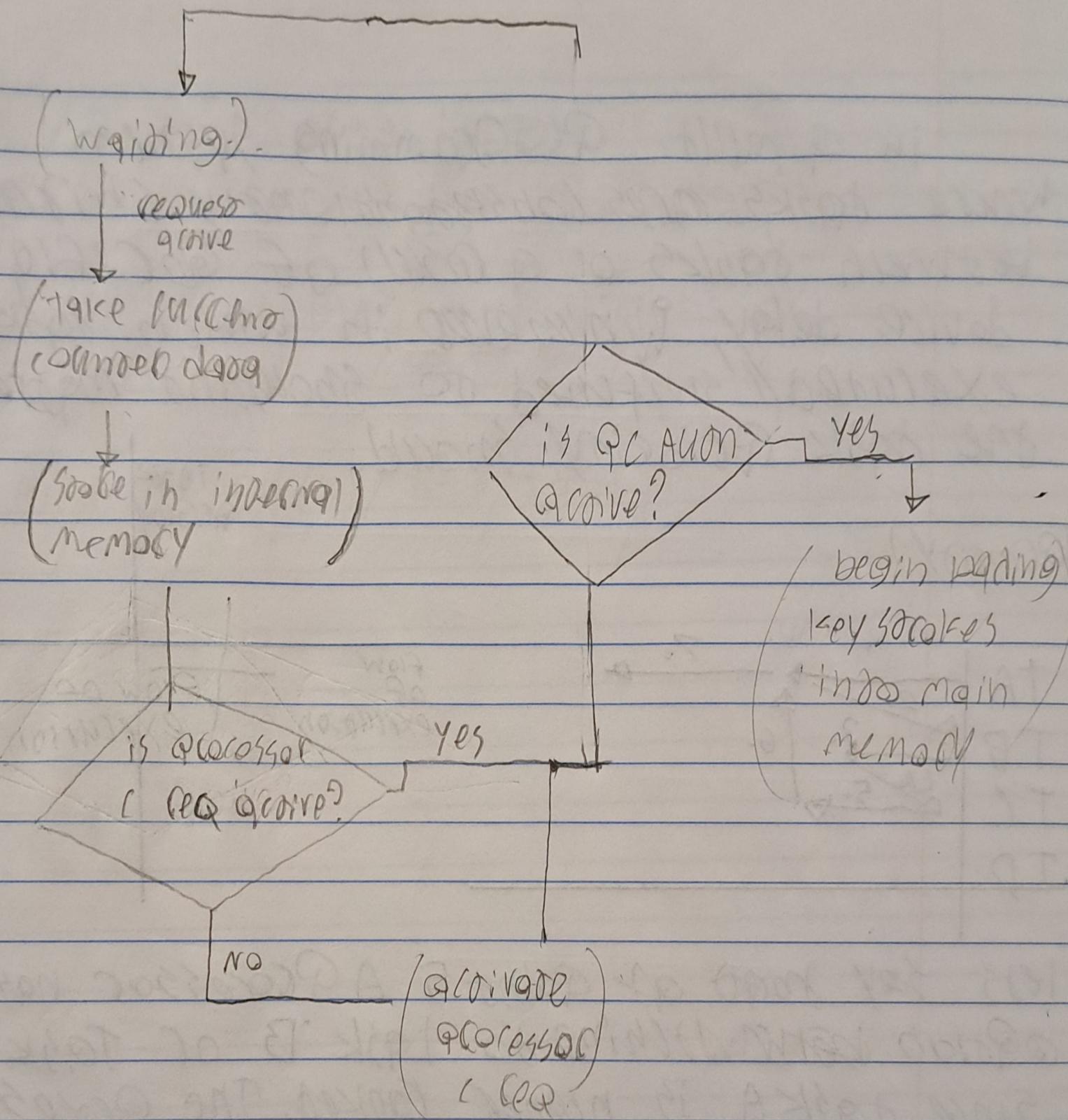


This system draws current if current is not actively being produced, and if current is produced, the output produces current.

## inquiry module.

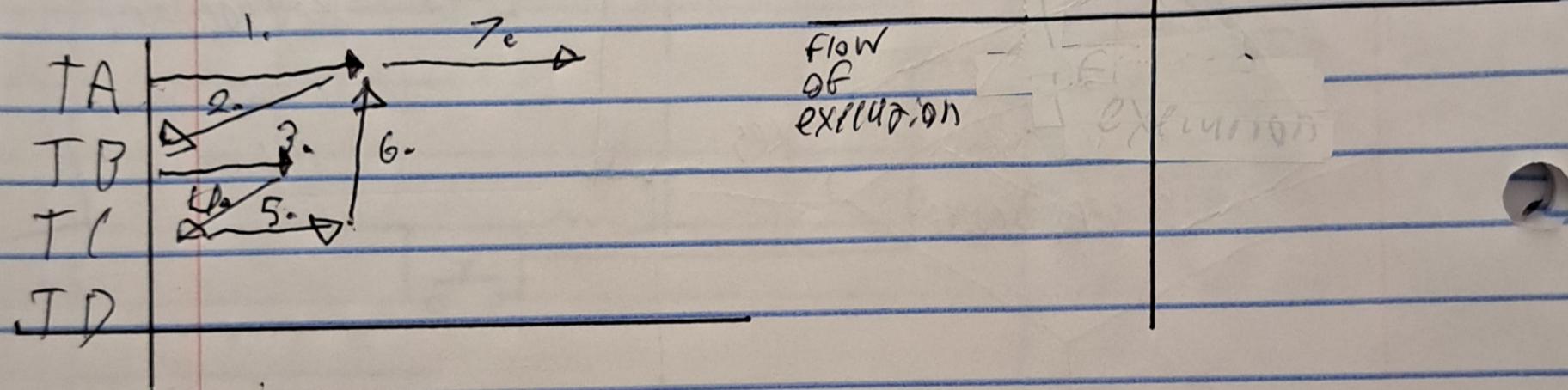


# Micro Processor State diagram



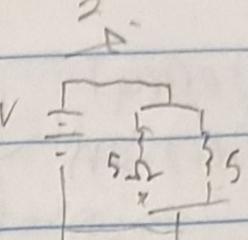
in a multi programming system  
where a processor constantly switches  
between tasks as a result of device I/O  
device delay, the order in which tasks first  
executed // switched to shouldn't matter, but  
the task priority should.

19312  
AT/06/04



Let's say now at point 5 a processor has the  
option of switching to task 13 or task A.  
since task A is higher priority, the processor  
will select to resume task A; regardless of  
the order in which the tasks were originally  
stacked.

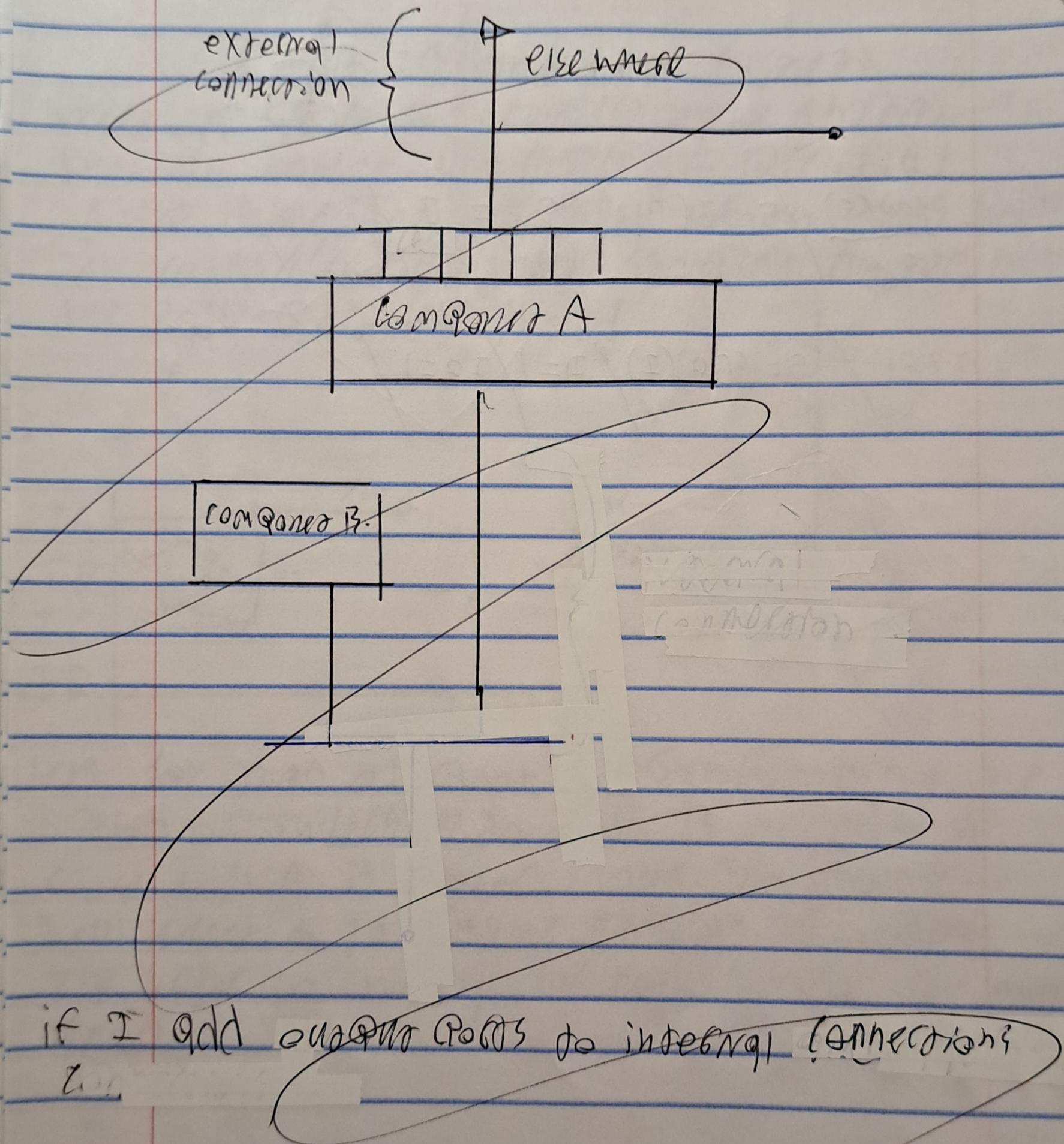
Random bullshit.

A)  $V = IR$  /  $5 = 2R$  /  $5 = (+\infty)(+\emptyset)$  /  $5V = \boxed{2}$  

B)  $R_{100} = \frac{(5)(5)}{5+5}$  /  $5 = I(B1A)$  /  $5 = I$  /  $I = 2$

C)  $V = I(R)$  /  $5 = (5\Omega)(I)$  /  $I = 1$  /  $I_2 = 1$

22

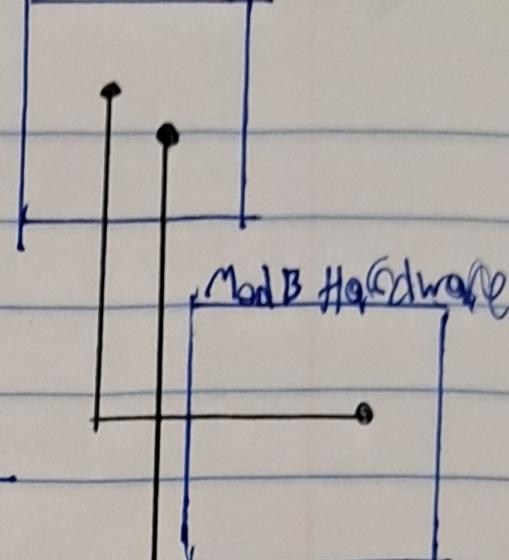


The hardware board houses all chips and passive/active components. All connections that would occur within a module are connected via an internal connection board.

All connections that would occur from one module to the next (i.e. buses) are implemented via inter modular PCBs.

There should be at least 10 available connection points for each signal on the module hardware board to ensure modules can access each other. At first, I'm not sure.

Mod B Intern



Module A Hardware board.

Component A

Component B

(x 10)

inter-modular  
connections @ C13.

Module A internal connection board.

14

Initial Score

Good start off strong

at development

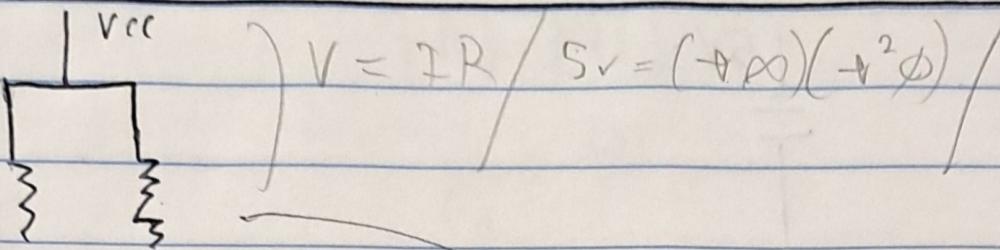
Initial - 20  
Good engineering

story

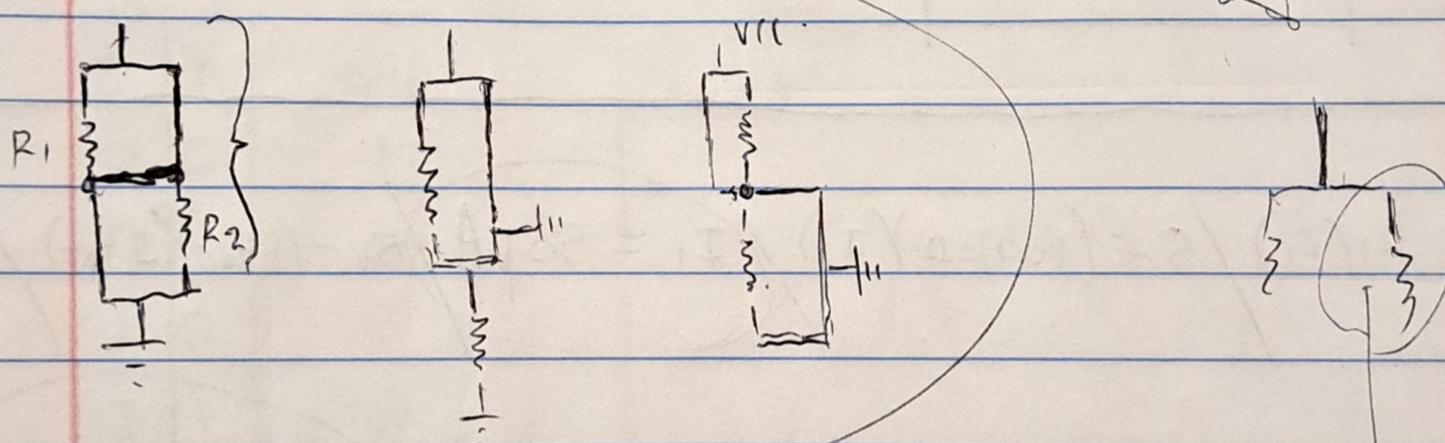
Good forward looking & unknown

15

ENSURING PROPER SUPPLY CURRENT & VOLTAGE  
INJES.



$$\left( \begin{array}{l} V_{cc} \\ \parallel \\ (R_1 \parallel R_2 \parallel S_V) \\ \parallel \\ (N_{DD}, R_{DD} -) \end{array} \right) \rightarrow I \uparrow \uparrow .$$



To make sure voltage  
is at the level  
from the adjustable resistor  
at the ground connection  
of the IC.

~~$$S = (N_{DD} + 1) R_2$$~~

$$S = (N_{DD} + 1) R_2$$

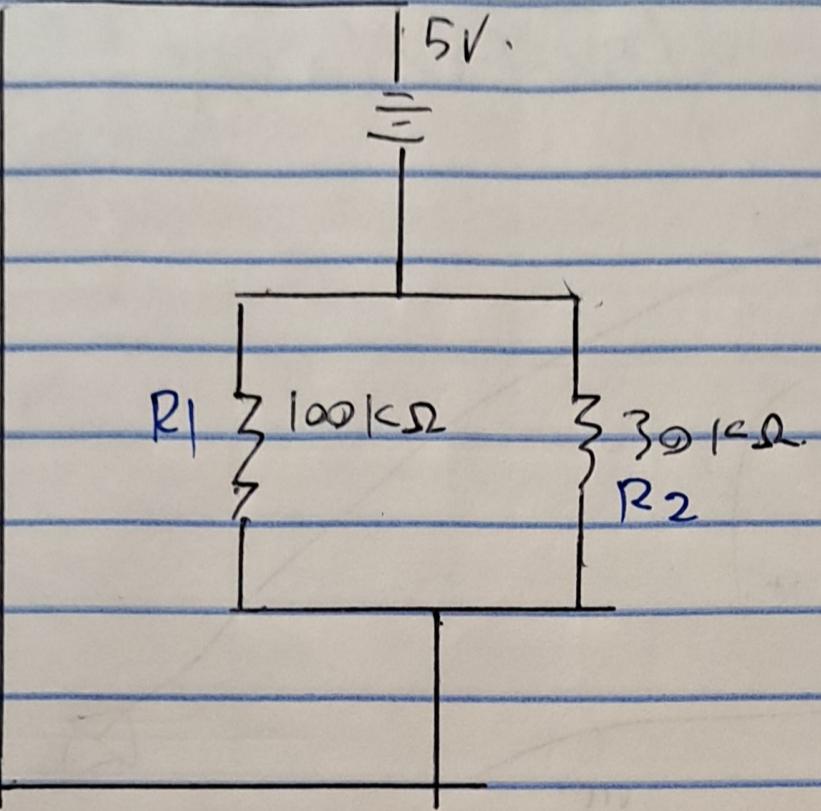
SV  
IC  
Adjustable

The II symbol  
symbolizes internal  
resistance on the  
Vcc pin.

A)  $S = IR / S = (\text{Supply current})(\text{internal resistance}) + (\text{Adjustable Res})$

B)

16



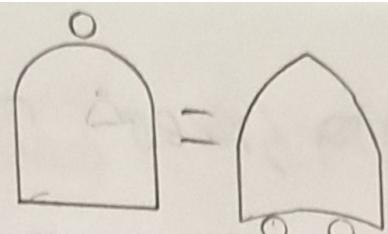
A)  $V_1 = R_1(I_1) / 5 = (100k\Omega)(I_1) / 5V = (R_{T00})(I_{T00})$

B)  $R_{T00} = (100k\Omega)(30k\Omega) / (100k\Omega + 30k\Omega) = 23.1k\Omega$   $I_{T00} = 217\mu A$

C)  $R_2 = R_2(I_2) / 5 = (30k\Omega)(I_2) / (I_2) = 167\mu A$

D)  $I_{T00} = I_1 + I_2 / I_{T00} = (50\mu A) + (167\mu A) / I_{T00} = 217\mu A$

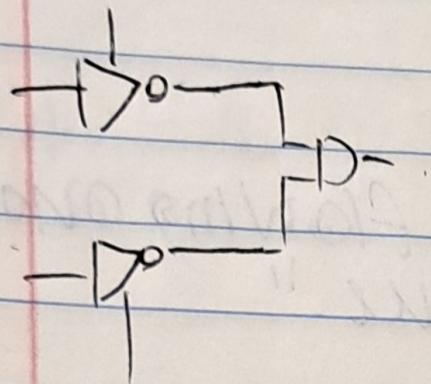
$$(\overline{AB} = \overline{A} + \overline{B}) \quad \{$$



17

$$[A] [B] ? \times$$

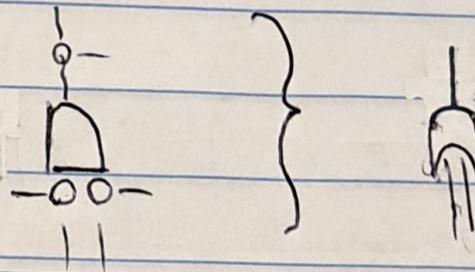
$$\overline{\overline{A}} = \overline{A}$$



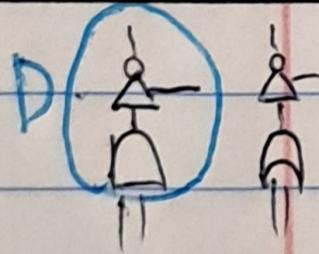
$$\overline{\overline{A}} = \overline{A}$$

[A]

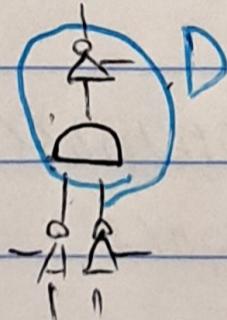
A	B	out
0	0	1
0	1	0
1	0	0
1	1	0



R  $\uparrow$



vs



} two chips  
in addition  
to (AND) and (inverter) chips,  
referred to as GROUP D.

Also depends on the cost

of inverters and the cost

of 20 gates

19

dangerous terms do look like

[recommended retarding condition version]

"The dangerous quantity coming flowing out of a device at a negative value."

$$V = IR/5 = (-s_0)(x)$$

outputs current: - The recommended amount of current you may draw from device, which changes based on resistance on the line. → basically, how much resistance you need on the line.

↳ recommended resistance to internal resistance.

I could

## All ICQY considerations

[Code 1091c]

A)  $Q_1 = +16 / Q_2 = +16 / \text{And} = +16 \cdot 7 = 48 \times 16 = 600$

[Zero flags]

B)  $\text{Total And} = 16. \text{Not} = 8. \text{And} 7 = 7$

[Code 1091c]

C) Octal bus transfer tot = 2 / 4 bit code tot = 1

[Addr/Sub]

D) 4 bit addr tot = 4 / inverter tot = 16 / Octal bus transfer tot = 2

[Multipli]

e)  $\text{Program tot} = 2 / 4 \text{ bit Addr tot} = \emptyset$

[Overflow flags]

f) Octal bus transfer tot = 2 / And tot = 4

g) Oct tot = 1 / inverter tot = 3

19  
Circuit of decommissioning  
[AIU]

X00, 24