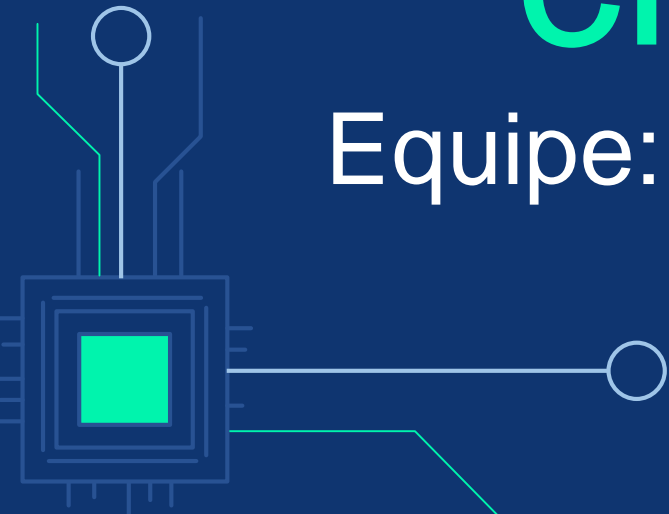


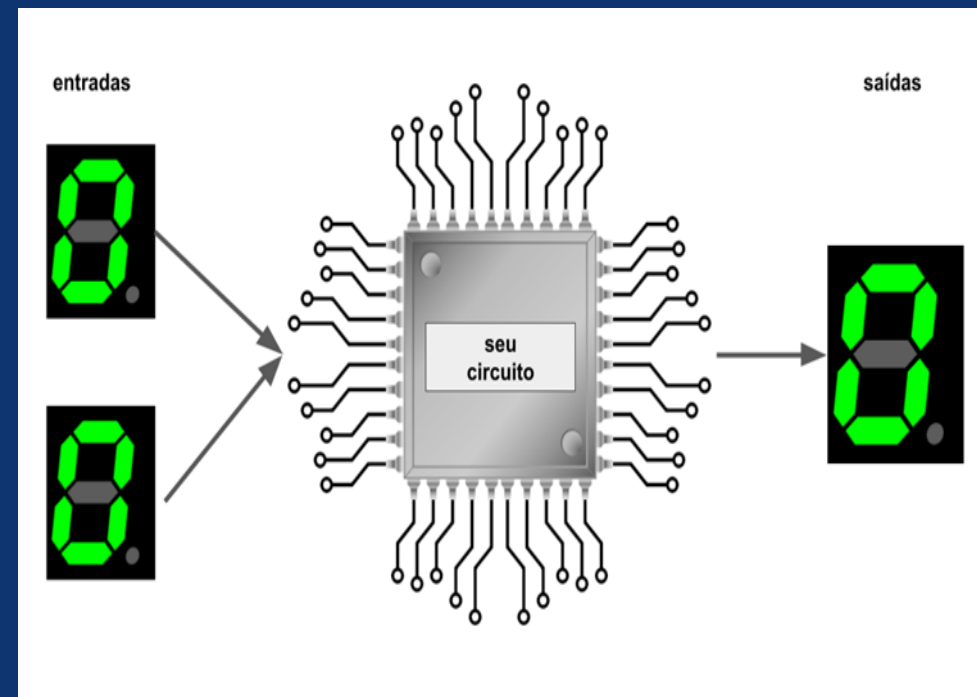
Trabalho de circuitos digitais

Equipe: Erick de Brito, Cicero Igor, João Marcelo



Objetivos

- Executar um projeto de soma de dois algarismos em base decimal.
- Aplicação de circuitos na linguagem C e na plataforma logic.ly.
- O usuário deve "desenhar" através de displays dois valores, que serão transformados em 4 bits e será mostrado um valor de 5 bits em decimal.



Início

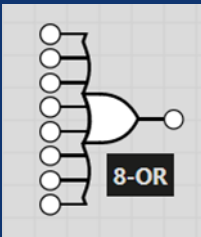
- Primeiramente vamos testar se é possível executar este projeto em C. Utilizando teclas do teclado para alternar os valores do input individualmente, podemos facilmente fazer uma ferramenta que faz a soma de dois números em binário. Mas como isso é feito em C? A linguagem tem ferramentas lógicas através do comando 'if' que pode fazer testagem com && (AND) e || (OR), diferente do C++ que possui o tipo de dado "bool", representando verdadeiro e falso, usaremos inteiros que trocam entre os valores 1 e 0, e usando constantes globais, vemos que: x_premissa = VERDADE;



```
int AND(int A, int B){  
  
    if(A == LIG && B == LIG) return LIG;  
    else return DES;  
}  
  
int OR(int A, int B){  
  
    if(A == LIG || B == LIG) return LIG;  
    else return DES;  
}  
  
int NOT(int A){  
  
    if(A == LIG) return DES;  
    else return LIG;  
}
```



```
int OCTOR(int A, int B, int C, int D, int E, int F, int G, int H){  
  
    if(A == 1 || B == 1 || D == 1 || E == 1 || C == 1 || F == 1 || G == 1 || H == 1) return 1;  
    else return 0;  
}
```



Parte 1

- Podemos observar que existe um padrão para a soma de 2 bits. Como as entradas devem ter 4, representaremos o resultado com 5, então podemos montar uma tabela para descobrir qual o padrão:
- SIM! Um padrão existe, podemos observar que quando as entradas são diferentes, a casa decimal em questão será verdade no resultado. Na lógica, já conhecemos a bicondicional, e sua negação é a disjunção exclusiva. Como descobrimos qual o circuito responsável pela D.E.? Conhecendo a porta AND e OR podemos usar a álgebra proposicional para descobrir.

$$\begin{array}{r} 0 \quad 0 \quad 1 \\ +0 \quad +1 \quad 1 \\ \hline 0 \quad 1 \quad 10 \end{array}$$

x	y	Unid.
0	0	0
0	1	1
1	0	1
1	1	0

$$\begin{aligned} P &\leftrightarrow q \\ (P \rightarrow q) \wedge (q \rightarrow P) \\ (\bar{P} \vee q) \wedge (\bar{q} \vee P) \\ (P \wedge \bar{q}) \vee (q \wedge \bar{P}) \end{aligned}$$

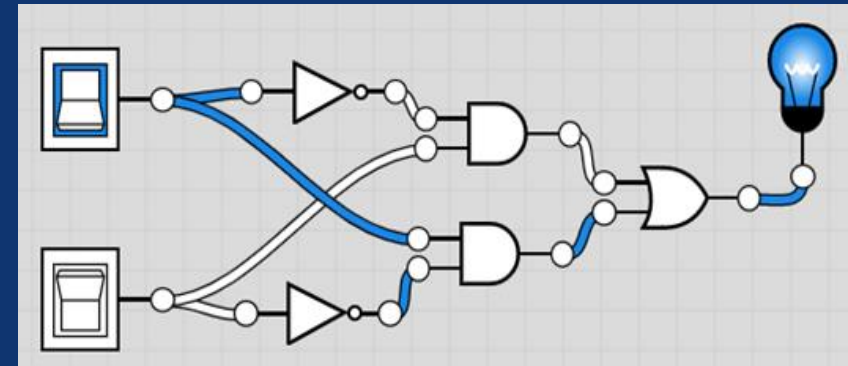
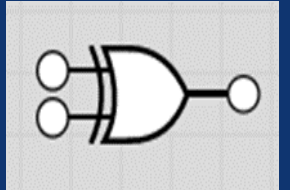
p	q	<u>p</u> <u>q</u>
V	V	F
V	F	V
F	V	V
F	F	F



Parte 1

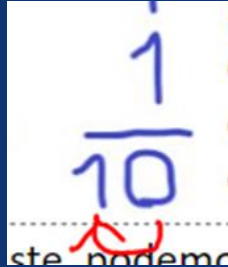
- Essa expressão comum motivou a criação, essa é a porta XOR, representada pelo símbolo \oplus

```
int XOR(int A, int B){  
    return OR(AND(A,NOT(B)),AND(NOT(A),B));  
}
```



Parte 1

- Agora só precisamos resolver o problema da casa decimal, que vimos anteriormente e ignoramos, como podemos representar isso? Bom, devemos prestar atenção que na verdade, o valor que sobra da casa anterior faz parte da soma dos dígitos da casa pra onde ela foi enviada, já que na verdade $1+1$ é o mesmo que $01+01$ e a na casa das unidades: $1+1 = 0$ e vai 1 para as dezenas, e na próxima casa teremos $0+0+1 = 1$. Espere.. Então no caso todas as somas devem precisar de 3 números, os 2 a serem somados e alguma possível sobrecarga da casa anterior.. Vamos anotar isso. Agora voltando, como podemos entender o problema da sobrecarga (em inglês overflow, ou seja, transbordar)?
- Podemos observar, que dos 3 valores que serão somados para nos dar o resultado daquela casa decimal, basta que PELO MENOS DOIS deles sejam 1. Podemos representar isso como um circuito? Sim, valeu Karnaugh!

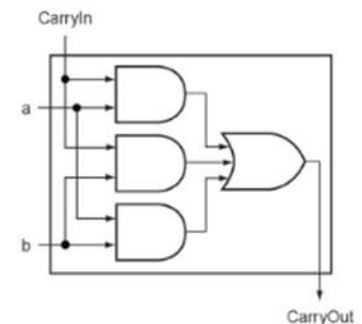


X	Y	CI	Overflow?
0	0	0	F
0	0	1	F
0	1	0	F
0	1	1	V
1	0	0	F
1	0	1	V
1	1	0	V
1	1	1	V

AB \ C	0	1	
00	0	1	$\rightarrow \overline{B}C$
01	1	0	$\rightarrow B\overline{C}$
11	1	0	
10	0	1	

Carryout bit

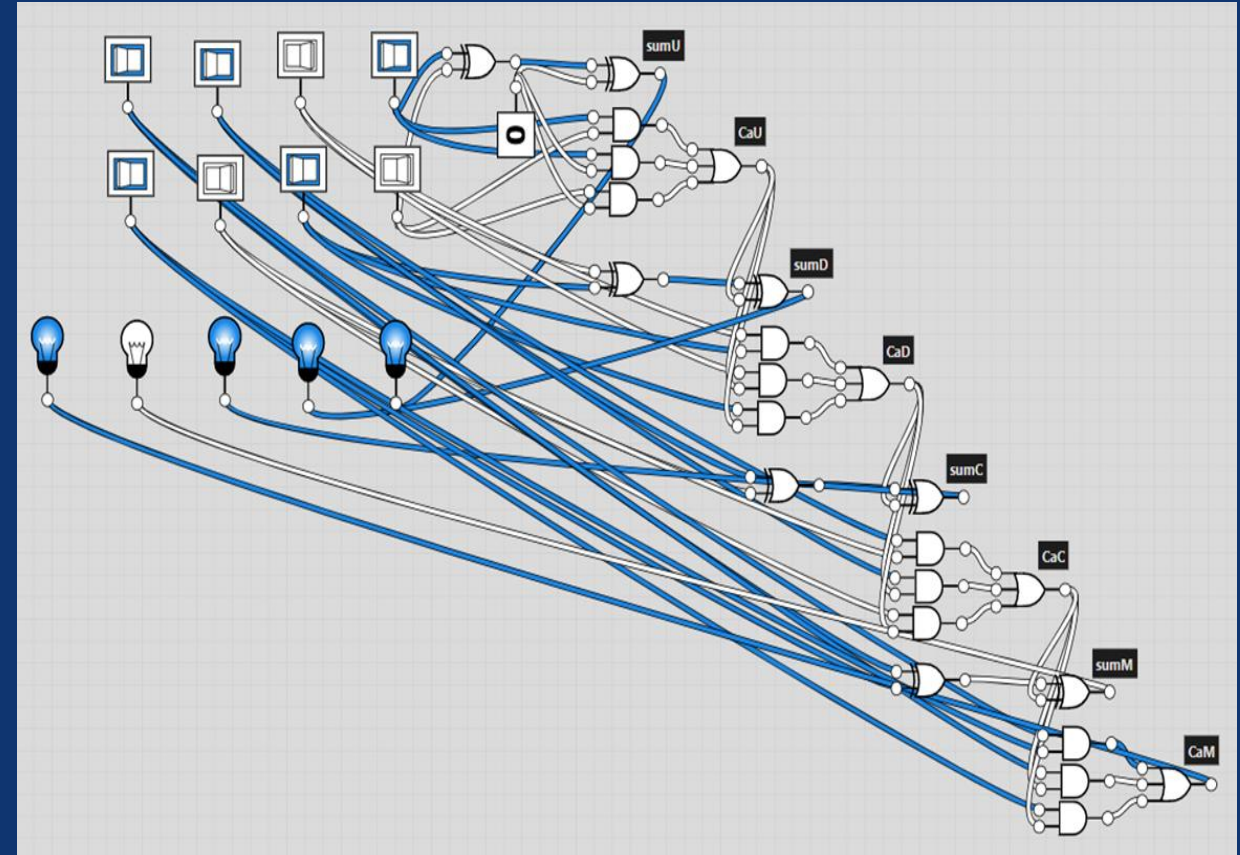
- Actually, it can be simpler $CO = (a \cdot b) + (b \cdot c) + (c \cdot a)$



Parte

1

- Lembra que todas as somas precisam de 3 entradas? Bom, nós já podemos começar a organizar a cadeia de somas e carries que montamos. Lembra que anotamos sobre as 3 entradas? Então no caso o resultado da disjunção exclusiva entre as duas entradas, deve ser somada ao bit que pode ter vindo da casa anterior, e como é feita a soma? Com a disjunção exclusiva.
- Também é óbvio perceber que nunca a casa das unidades recebera um carry, e vamos fazer o resultado com 1 bit a mais que as entradas, justamente para não perder o possível carry da maior casa. Teríamos enorme economia de portas se ignorássemos o carry das unidades, mas para deixar menos confuso, ele estará presente mas com uma constante BAIXA. Deu certo! Mas importante notar: a plataforma permite personalizar um circuito, onde qualquer sistema pode ser simplificado por uma caixinha com as entradas e saídas, então próxima vez que a soma aparecer, não será feita assim!



Finalizando parte 1

- Bom.. Isso foi.. Menos de 20% do projeto. Vamos lá para a parte mais difícil, e vamos representar o codificador, que transformará a resposta de 5 bits em valores decimais. Eu não sei nem por onde começar.. Mas vamos tentar. Vamos colocar as informações que temos num papel, e ver se conseguimos pensar numa solução.

Handwritten notes and a logic diagram on a spiral notebook page.

Logic diagram showing inputs A, B, C, D, E, F connected to a circuit with AND gates and OR gates. The output is labeled C.

ABC	DEF	Output	ABC	DEF	Output	C	F	to d
000	000	0000	100	100	1000	0	0	0
000	001	0001	100	101	1001	0	0	0
000	010	0010	100	110	1010	0	0	0
000	011	0011	100	111	1011	0	0	0
000	100	0100	101	101	1101	0	0	0
000	101	0101	101	110	1110	0	0	0
000	110	0110	101	111	1111	0	0	0
000	111	0111	110	101	1101	0	0	0
001	001	0010	110	110	1110	0	0	0
001	010	0011	110	111	1111	0	0	0

Handwritten notes and a logic diagram on a spiral notebook page.

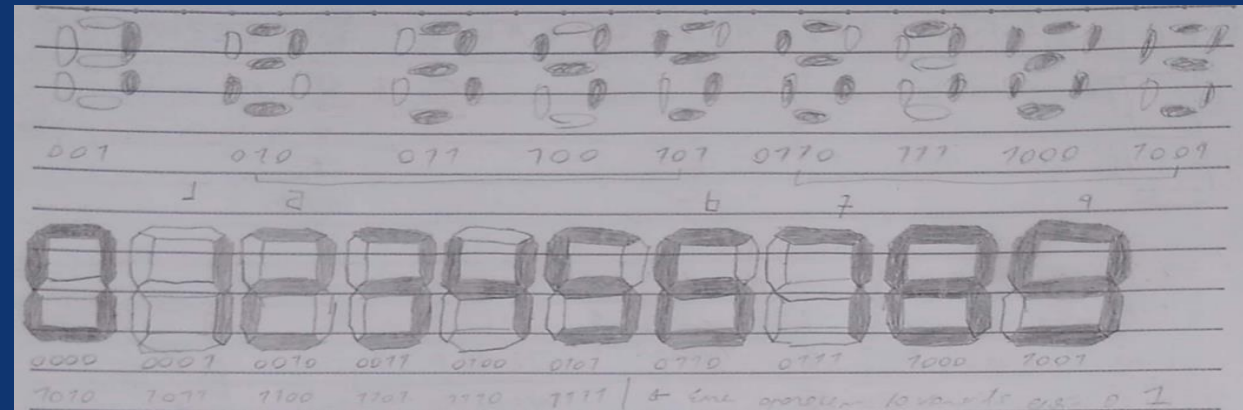
Logic diagram showing inputs A, B, C, D, E, F connected to a circuit with AND gates and OR gates. The output is labeled C.

A	B	C	D	E	F	Output	C	F	to d
0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0
0	0	0	1	1	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0
0	0	1	0	1	0	0	0	0	0
0	0	1	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0	0	0
0	0	1	1	0	1	0	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	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	0	0	0
0	1	0	1	0	1	0	0	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	0	0	0	0
0	1	1	0	0	1	0	0	0	0
0	1	1	0	1	0	0	0	0	0
0	1	1	0	1	1	0	0	0	0
0	1	1	1	0	0	0	0	0	0
0	1	1	1	0	1	0	0	0	0
0	1	1	1	1	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0
1	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	0	0	0
1	0	0	1	0	1	0	0	0	0
1	0	0	1	1	0	0	0	0	0
1	0	0	1	1	1	0	0	0	0
1	0	1	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0	0	0
1	0	1	0	1	0	0	0	0	0
1	0	1	0	1	1	0	0	0	0
1	0	1	1	0	0	0	0	0	0
1	0	1	1	0	1	0	0	0	0
1	0	1	1	1	0	0	0	0	0
1	0	1	1	1	1	0	0	0	0
1	1	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0	0
1	1	0	0	1	0	0	0	0	0
1	1	0	0	1	1	0	0	0	0
1	1	0	1	0	0	0	0	0	0
1	1	0	1	0	1	0	0	0	0
1	1	0	1	1	0	0	0	0	0
1	1	0	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0
1	1	1	0	1	0	0	0	0	0
1	1	1	0	1	1	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	0	1	0	0	0	0
1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0

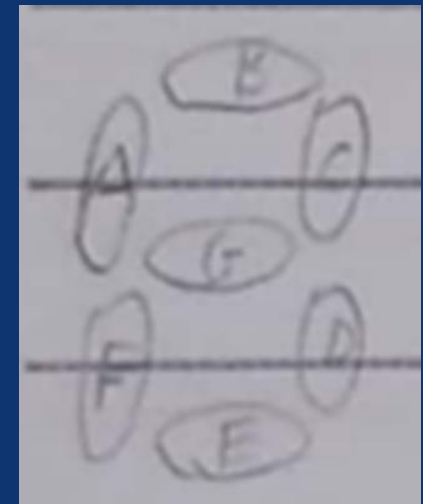


Parte 2

- Como fazemos pra descobrir o circuito que transforma um número em binário de 5 algarismos em um display de 7 segmentos que represente o número corretamente?
- Podemos perceber através das tabelas verdade que de alguma forma podemos ser capazes de organizar os circuitos..



U/W/E/D	A	observa tabela
0000	1	NUNWANEANR
0001	0	
0010	9	
0011	0	
0100	1	NUNWANEANR
0101	1	NUNWANEANR
0110	1	NUNWANEANR
0111	0	NUNWANEANR
1000	1	NUNWANEANR
1001	1	NUNWANEANR



Parte 2

- Usaremos as nomenclaturas dos displays desta forma: vertical/horizontal-esquerda/direita/nenhum-inferior/superior/central.
- Podemos perceber, que já que não há um padrão, as 5 entradas devem se organizar de uma forma que consigam representar as configurações de cada display, mas como observamos que todos se apresentam de forma diferente, teremos de fazer um circuito para cada um dos 7 displays.
- Relembrando o que foi explicado em sala, para isso devemos usar "a soma dos produtos" para descobrir as expressões que transformam as entradas nos padrões de cada display. Mas como isso funciona exatamente?

A	B	C	Q
0	0	0	0
0	1	0	1
1	0	0	1
1	1	0	0
0	0	1	1
0	1	1	0
1	0	1	0
1	1	1	1

To write down the Boolean expression that describes this truth table (and therefore the system that the truth table describes) we simply write down the Boolean equation for each line in the truth table where the output is 1.

The output for the first line is 0, so we ignore it.

The output for the second line is a 1. The Boolean equation for this line is $\bar{A}.B.\bar{C}$

The output for the third line is a 1. The Boolean equation for this line is $A.\bar{B}.\bar{C}$

The output for the fourth line is 0, so we ignore it.

The output for the fifth line is a 1. The Boolean equation for this line is $\bar{A}.\bar{B}.C$

The output for the sixth line is 0, so we ignore it.

The output for the seventh line is 0, so we ignore it.

The output for the eighth line is a 1. The Boolean equation for this line is $A.B.C$

We can now get the Boolean equation for the whole system simply by getting the equations where the output was 1 and ORing them together. This gives us the output Q:

$$\bar{A}.B.\bar{C} + A.\bar{B}.\bar{C} + \bar{A}.\bar{B}.C + A.B.C = Q$$

<http://theteacher.info/index.php/fundamentals-of-cs/2-logical-operations/topics/2642-deriving-boolean-expressions-from-truth-tables>



Parte 2

- Se Q é falso ignoramos, mas quando Q é verdadeiro, devemos guardar de lado A.B.C, sendo que os que forem 0 são barrados. No final, somar todos juntos.
- Exemplo, do lado: na primeira linha, '?' é verdade, P e Q são verdade, então temos P.Q, pulamos a 2º, na terceira teremos !P.Q e por ai vai. No final então, somamos tudo, obtendo PQ+!PQ+!P!Q agora vamos simplificar para reconhecer essa tabela verdade:
- Deu certo! Realmente era a condicional, Pois Vera Fischer é Falsa.

P	Q	?
V	V	V
V	F	F
F	V	V
F	F	V

P	Q	?
V	V	V
V	F	F
F	V	V
F	F	V

$$(P \wedge Q) \vee (\bar{P} \wedge Q) \vee (\bar{P} \wedge \bar{Q})$$

$$Q \vee (\bar{P} \wedge \bar{Q})$$

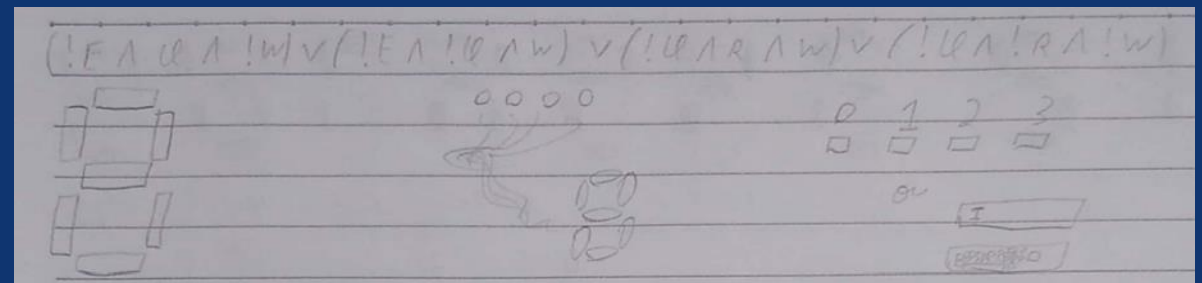
$$\bar{P} \vee Q = P \rightarrow Q$$


Parte 2

- Então esse é o método que usaremos para montar as expressões, a partir das tabelas verdade. Outra coisa que estamos adotando: como as entradas do usuário vão de 0 até 9, a maior soma possível é 18, então trabalharemos apenas com a casa das unidades, mas de 00 até 18, ou seja, os números de 00000 até 10010 serão representados por 0123456789012345678. É muito importante ter isso definido, pois a tabela verdade precisa estar completa para o método funcionar, por exemplo, mesmo que 11011 seja maior que 10010, ele obviamente não vai nunca ser um resultado mostrado pelo projeto (pois o máximo é 9+9, não há como obter o 27) mas mesmo assim precisa estar definido nos circuitos. Como são 5 bits, a tabela verdade terá 32 linhas, que serão: 01234567890123456789012345678901, de 00 até 31, mas sempre estaremos representando apenas as unidades.

Z	X	C	V	B	A	B	C	D	E	F	G
0	0	0	0	0	X	X	X	X	X	X	X
0	0	0	0	1			X	X			
0	0	0	1	0		X	X		X	X	X
0	0	0	1	1		X	X	X	X		X
0	0	1	0	0	X		X	X			X
0	0	1	0	1	X	X		X	X		X
0	0	1	1	0	X	X		X	X	X	X
0	0	1	1	1		X	X	X			
0	1	0	0	0	X	X	X	X	X	X	X
0	1	0	0	1	X	X	X	X	X		X
0	1	0	1	0	X	X	X	X	X	X	
0	1	0	1	1			X	X			

0	1	1	0	0		X	X		X	X	X
0	1	1	0	1		X	X	X	X		X
0	1	1	1	0	X		X	X			X
0	1	1	1	1	X	X	X	X	X	X	X
1	0	0	0	0	X	X		X	X	X	X
1	0	0	0	1		X	X	X			
1	0	0	1	0	X	X	X	X	X	X	X
1	0	0	1	1			X	X			



Parte 2

- Bom, agora vamos montar uma tabela mais bonita. Perceba que não é necessário seguir preenchendo após o 9, pois os números de 10 a 19 e de 20 a 29 tem sempre as mesmas unidades que os de 0 a 9. Então a sequência repete 3 vezes e falta apenas 30 e 31 no final. Agora nós pegamos os valores das linhas de cada coluna. Por exemplo, para o 'A', que representa a vertical esquerda superior (VES), nós pegamos esse valor 1000111011 e repetimos 3 vezes e adicionamos os 2 primeiros.
- Faremos isso novamente todas as vezes para cada uma das colunas, que representam os displays da saída. Então não precisaremos repetir esse passo para 'B', ou 'C' e por aí vai. Mostraremos apenas o resultado final, que é a única coisa que difere.
- Devemos também pensar num método para representar as dezenas, que apenas alternam entre 0 e 1 (00 a 18)

z	x	c	v	b		a	b	c	d	e	f	g
	0	0	0	0	0		1	1	1	1	1	1
	0	0	0	0	1	1	0	0	1	1	0	0
	0	0	0	1	0	2	0	1	1	0	1	1
	0	0	0	1	1	3	0	1	1	1	1	0
	0	0	1	0	0	4	1	0	1	1	0	0
	0	0	1	0	1	5	1	1	0	1	1	0
	0	0	1	1	0	6	1	1	0	1	1	1
	0	0	1	1	1	7	0	1	1	1	0	0
	0	1	0	0	0	8	1	1	1	1	1	1
	0	1	0	1	0	9	1	1	1	1	1	0
	0	1	0	1	0	10	1					
	0	1	0	1	1	11	0					
	0	1	1	0	0	12	0					
	0	1	1	0	1	13	0					
	0	1	1	1	0	14	1					
	0	1	1	1	1	15	1					
	1	0	0	0	0	16	1					
	1	0	0	0	1	17	0					
	1	0	0	1	0	18	1					
	1	0	0	1	1	19	1					
	1	0	1	0	0	20	1					
	1	0	1	0	1	21	0					



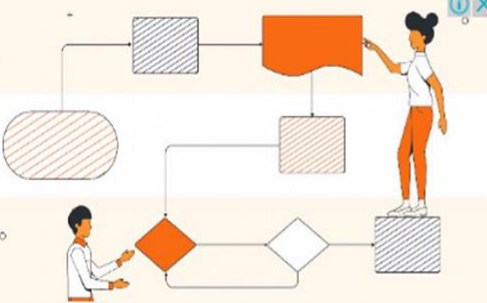
Parte 2

- Lembra do método que descobrimos para derivar uma tabela verdade de volta a expressão booleana? Bom, agora nós temos o resultado das tabelas verdade de cada display. Já que o professor liberou em sala, não precisaremos quebrar cabeça fazendo a álgebra e as simplificações, pegamos o valor de 0 a 9 do VES, que é 1000111011, repetimos 3 vezes e adicionamos ao final os 2 primeiros: 1000111011 1000111011 1000111011 10
- Agora apenas colamos isso em um site que simplifique por nós. Os resultados do cálculo do site, ali em marca-texto amarelo, representam a expressão proposicional, onde 'A' é o algarismo das dezenas de milhares, e o 'E' as unidades (em binário). Onde quer que vamos aplicar nosso circuito, precisamos completamente delas, que ao total serão 12.
- Fonte: <https://www.dcode.fr/boolean-truth-table>

Results

$f(a,b,c,d,e)=$

$(a \wedge b \wedge \neg d) \vee (a \wedge \neg b \wedge \neg c \wedge d) \vee (\neg a \wedge b \wedge c \wedge d) \vee (\neg a \wedge \neg b \wedge c \wedge \neg d) \vee (\neg a \wedge \neg b \wedge c \wedge \neg e) \vee (b \wedge \neg c \wedge \neg d) \vee (b \wedge d \wedge \neg e) \vee (\neg b \wedge \neg d \wedge \neg e)$




Online Flowchart Maker

Drag-and-drop interface & extensive shape library. Free 7-day trial.

Truth Table - dCode

Tag(s) : Symbolic Computation, Electronics

Share



TRUTH TABLE GENERATOR

★ LOGICAL/BOOLEAN EXPRESSION

a xor b

★ DISPLAY

☒ THE FULL TRUTH TABLE (INPUT VARIABLES + OUTPUT VALUES)

☐ THE TRUTH TABLE WITH THESE VARIABLES a

☐ ONLY OUTPUT VALUES (LIST OF 0 AND 1)

☐ THE ASSOCIATED MINTERMS $\Sigma m(\dots)$ (0-INDEXED)

☐ THE ASSOCIATED MAXTERMS $\Sigma M(\dots)$ (0-INDEXED)

▶ GENERATE

See also: [Boolean Expressions Calculator – Boolean Minterms and Maxterms](#)

FIND EQUATION FROM TRUTH TABLE

Indicate only the output values of the function (the last column from the boolean truth table)

★ OUTPUT VALUES (LIST OF 0 AND 1)

1000111011 1000111011 1000111011 1 0

★ TABLE ORDERED (INPUT VALUES) ☒ FROM 0,...,0 TO 1,...,1

☐ FROM 1,...,1 TO 0,...,0

★ BOOLEAN NOTATION ☐ LITERAL (AND, OR, NOT)

☒ LOGICAL (\wedge , \vee , \neg)

☐ PROGRAMMING (&&, ||, ~)

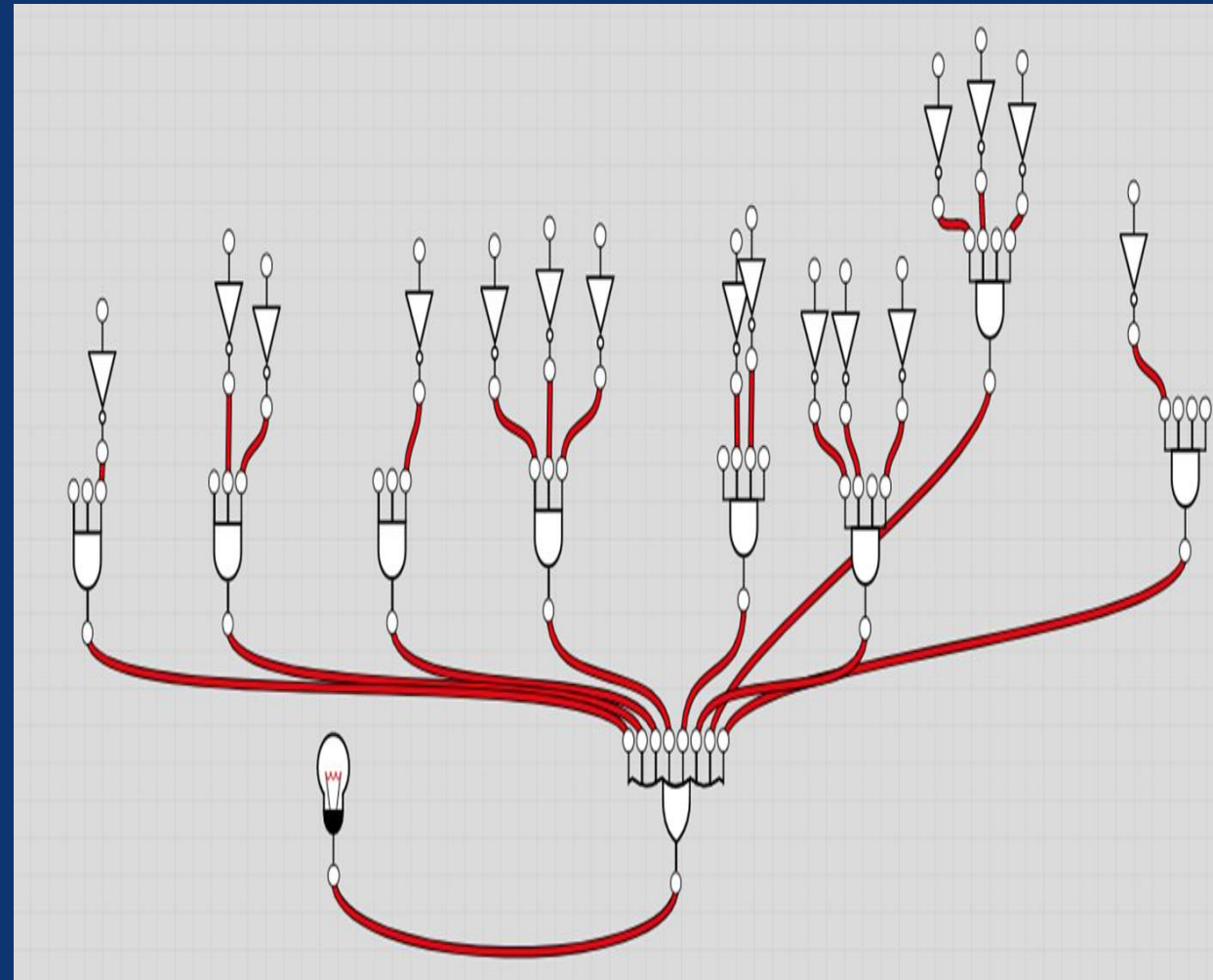
☐ ALGEBRAIC (*, +, !)

▶ CALCULATE



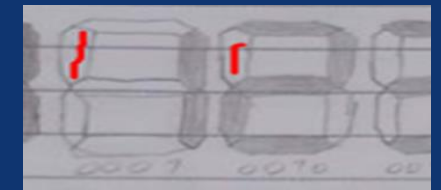
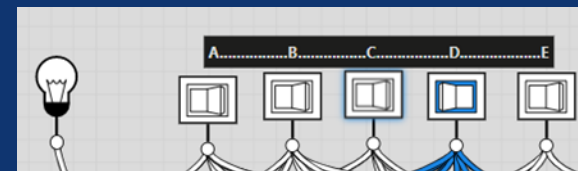
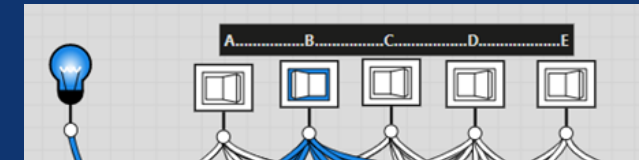
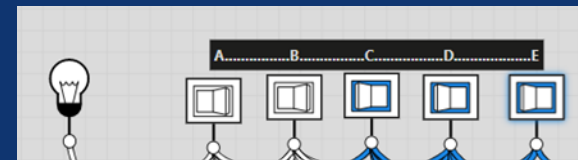
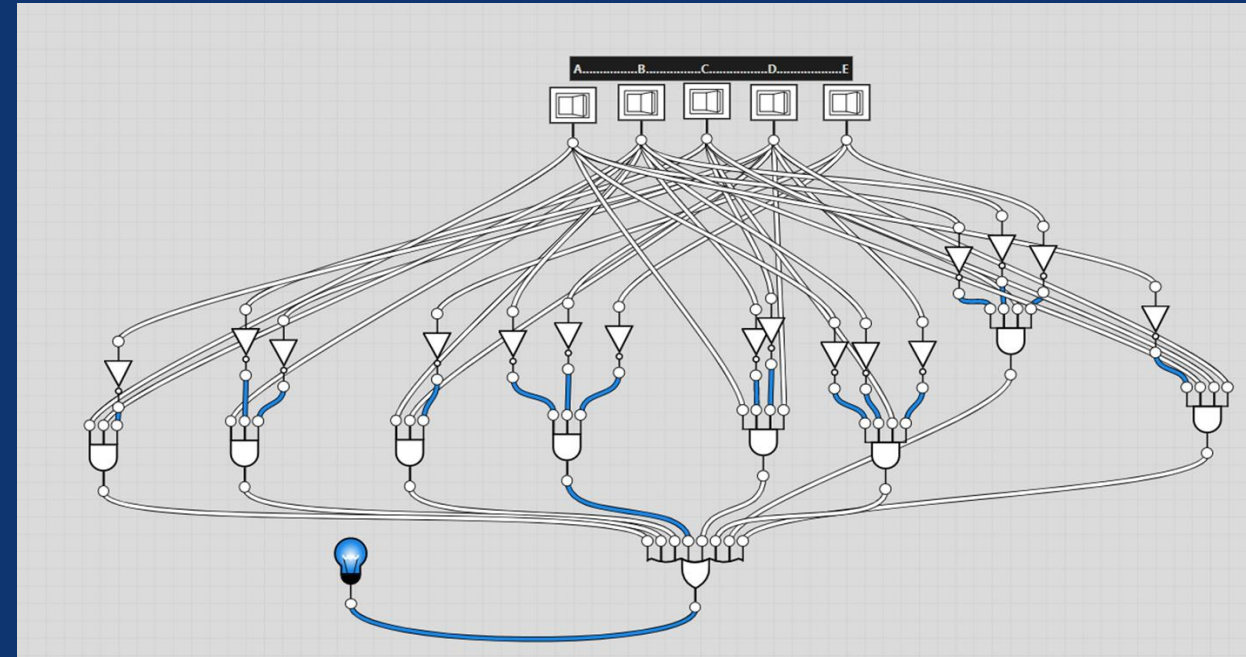
Finalizando parte 2

- Essa é a expressão do circuito que transforma os 5 bits do resultado da soma no display VES. Observamos que são 8 parênteses entre OU's, e dentro dos parênteses existem apenas E's. Sabendo que A, B, C, D e E estão em ordem, podemos também ir organizando os barrados, exemplo, são 8 E's como entrada de um único OU, o primeiro E tem apenas 3 entradas e a última é barrada, vamos organizar:
- $(a \wedge b \wedge \neg d) \vee (a \wedge \neg b \wedge \neg c \wedge d) \vee (\neg a \wedge b \wedge c \wedge d) \vee (\neg a \wedge \neg b \wedge c \wedge \neg d) \vee (\neg a \wedge \neg b \wedge c \wedge \neg e) \vee (b \wedge \neg c \wedge \neg d) \vee (b \wedge d \wedge \neg e) \vee (\neg b \wedge \neg d \wedge \neg e)$ vertical esquerda superior (A).



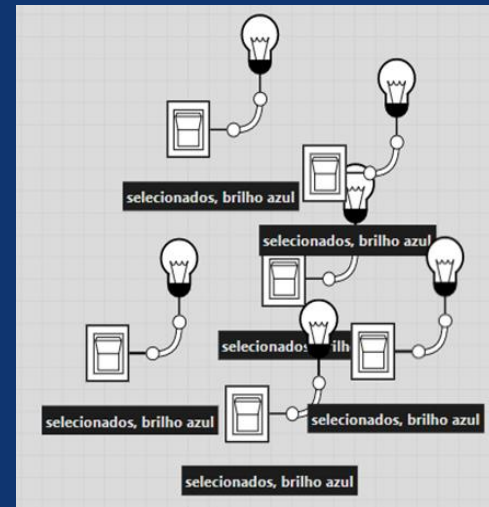
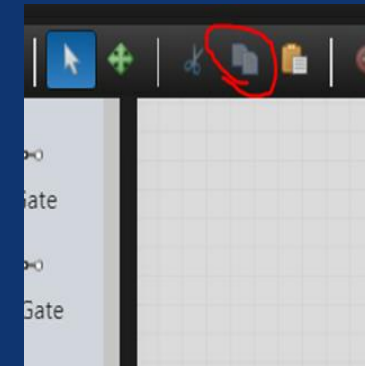
Parte 3

- Após organizar corretamente em ordem temos aqui o circuito funcionando. Mas como sabemos que está correto? É bom testar um por um, pois as vezes os erros são discretos. Por exemplo, o 2 e o 7 não tem, mas o 8 tem os display VES aceso.



Parte 3

- No site logic.ly/demo que é muito útil para a prática, podemos perceber que ao selecionar um circuito inteiro segurando e arrastando com o mouse, é possível copiar com ctrl+c, que trará um código bizarro para a área de transferência. Esse texto, quando colado de volta no site, replica totalmente o circuito como ele estava, até com os fios conectados.
- Mas por que estou explicando isso? O que tem a ver? Bom, é que eu salvei o código de todos os circuitos que fiz, então se alguém quiser estudar, analisar, melhorar ou até roubar, bom, estou compartilhando com boas e ingênuas intenções. Qualquer pessoa poderá copiar esse código e colar no site, criar um "custom" (versão simplificada do circuito) de cada um dos displays e ver o resultado final na prática (eu também salvei o resultado final). Bom, agora eu vou mostrar apenas as fotos e códigos.



<http://ns.logic.ly/1.6;base64,jdLNbtwgEAfwV7E4BzwMw1cVRz3kkgfoOQKMs66ovYqkstunr+tNcnAuuTlz4sd/uC/zy5jKtbn8KdPSsVOt5x9tOy1iL4hybaUw7OF+jr9zqk29nnPHyvyqs9xLfHnRxtr1rHvmHEQIGXgKQ09J5cjwwqJJ0Dpkld90Jo1l44hCq/BgkRptDVGE2uuHZPaCFDeW6PAASmNrHmda6jjPHUMWHuQLG9jTaejgmK0mQxw7aXkJO3AvTaKy0gDupSHIHFXbEJB1jkrwYOSUiuZM1CRIAQ0IMkq8ObAaOa1nte6BTaEsuQvrBJiLkeVDSAjaMezU4pTsobHSJprTcE7RNQ03FQeBDnlAD2g27K5oRRYYQGV81J5jWQOppovdUskl5y2o9DPy10TX8dymvpwdy3/kWmepk25ld9f8Ovp8btX3Qaepj5f9uvG6WP+e0vf+z/HN0z7/vce/gE=>



Parte 3

data:http://ns.logic.ly/1.6;base64,vVrRbuM4EvyVwO/UNNnNznOxs7iHe9kPuOcDKYk7OeSSwY4HN/P3V7l0wGlzFmhYSYLkIbblcnd1VTWVX59e/ngcn74/fPvw0/OXj6dP5/PnXz58eP4yXB4Ynr5/8lOefvv1pf5nHs8P5++f54+nL/97P1+f/vHjKaeHr4/Tx1NgphySOYt5dpJLdjXGyRm1VK1ZSb6eHr59PLHy4MU4iekayPLp4TteTkOkZMLasqeU/enhz5dzOT++PH88ZTo9vHw9f/56BshWnr7Mpw9doCZfdMw8uuJTcxJ8cnVu3tVMIGoutZpeQAnbgPfgU6uKMvEFIE9DAklyUfHJoh4CKmqZZxmLCzGyEx9nZ4nN5clPU+DSfA4XUNHTIJo5Ja/EOUZZUenAHGM0b1E8m8RDYBFZm3QaXa6lApZGZ1aaa+b9bDxbqX6FZTYJJZQks2R8iDeslU5FpjazizFkJzWoK4WTi36q2mjKyY8XUGjbEDh4VIU4pHwlpv18/R1Qy1RHy+LyqABkpbhiai6EiWpplUKIF0AuDjJwzhWgYiCA5UbpgeKJTvc8+KEcxL3/H9Pi8QuJXcJ5fzq8l3kYLuamzaVInrQJOqN7xWJLV5seobYWjQ1TC+wFPyQxrGOX/BCzcGBTBcPCNTS+Bw1rDZny6KZ5KU4cxVWailPO0goVs3HtFseByOM7RA8i6VqaYDKQJAKpNOXM/Vh+0qgiLc3TmF2BDDkh7x3GrrpaEk3THFEvumDxKQ+UJRigBCO1lc9o3iALbyz7GCjqfY2iGZ+rTMHnqRilbNVIMXZcdAqUJpmDXeAEQhFCAg4jrx5jvzVKBw+9lOgFW+alZhdUh70m0NWchyw7kTa5HAo7xeyT5yal6tapMIRIMaFxlI5K04YUBRcJAYoUKBxk3VTccqkFpZqFAG/itMypCeQqy2gJT5XuMVaHEuDZzEUKBuHHH4Uh5dhi5i3jFbFq63qYjFNQUopGKlaUJsSmjPgckGk5iPcJPqy1kZsgH95L3xpTdIVDQwugdnJUB2LMd/Qqp/AgQZXg1+4UKs4vJV35ith2PFFNME10+phxPBQ9oAknFCdbao0DfBV1AzTz3RLCx5CHKqJ65QmylzhWWDWJrvA0uwxVg0gjudVcHxM4AjYGqIm8pnipn8yZEsSiJiCvjavm5hTZS6QutHJFBLkuBWKDiGHnzEY+yLOZYMjinn2i+BAk2P2G5MJPYS9KiYcchxe5Y6bqlOTzDFAdGCWeRkrdbGhLEyqSMomutK5Jgi7CqnyB61kR9gFnqrZPQQyUghydfQSE9x0l0wEzLG2BLMoRjMYUkb48xxzjSjBqtXYWIGMVu0DlPkW/pZvCHgF9lYHuPrneqicUvoBVwGyYlggFybg/4Ex+RngnVitKetNH7wHg2MSRi6s4Ye9GxIAjIzJ5Qs8X3uYKlkxvSAtwLe8Bwhx3NwEbRuWZth1i5oEuVBA4hjPgSOIDcSB+hQkLTKMMC0u/o0za14VNhRQEGkxYoQDf2ZK/JOWrTFukZnFRpQwmWSF9Okh7YJa8iSgVohgkp3yh+XQhgiJzOChPgWkZkrEgZVAbUbGrWhAYcJcovEZZFf8a/vq0RHuDCtPltLbwLjarOMWOEUMBUbtHY5QijqvDLWutlcYsUirNisWV4Qo5lft9qE4ek2SBEGCiVq43qYk20mQsjY40oEPKNoiqzqJu8iFWCXW0mjsA5XALyggRE3sAkLDnMumRBLEvxPtbM2FaMY0PYAiIJC8cZ3leStao1Ta1jcJpEek4GIAGLB06ZeMFIrRHTVJEm+6LfvPcKmhZweGEpQb5EsqXBNKTaUoVQ59WYrYHRY6CS+iUD06fbShOGjKlKW AUCUL7TMIPBGIUo4LtORk9UiCE3wUqPGvBRujzCgYdQfaFO0FpPPO2YCVYIGQEa2QPGYvfQOGfsGY0rAAxwsAR2yDDCKHmJ4SKqpC3wrG0tTSL6g+mALTIMMcNDitokyRrBPvRxavK1yc2AdJLa8vjB GqSH6MDDhKgQw3RTRdNwYE8EX5sF6iFnoR3IXDgvZxQj+xroM910n8ujYvf74Kfgw191KdjrocGIREQJTcphHyCM+0GtcYinSFGmB/x34FIdwODZCEB7QyC0i97DZXsdgrLE+Pf3w6/7t+faqvhrxga2htxNbil7yFSS+QfshOZEWRMF1rwgFL0A4gw36HFkHrt7CSY7e5LFMoHaLhXzHRgmN8eX4GEPxh2z//9fs/eze59QW/P0/zt+Vq6we8vL5vMb08f3u5vw6mb5E7EsxOZfoY3nUwfZG1E0zfHncdTN9a2gmmb2+6DqZvC/wrmLBtMa616UgwOwTuW5qOBLPTpr4cfh1M31bR2aa+HH4kmJ029R0aHznaO5XpOyw+EsxOZfpOY3fa1C WanQS+tzK328Fom/qOz6+D6TPaXju4s019RvtObbrdm/bs4M7K3A5mZ5r6biPsjHaX679Xm7pc/50l3Of6nWDubVOf6/e69p3edLtr8ttxi+CdFbmXgXuYDvRODb2/SGnOk7CXwnnek7CHynNvWdA74TgtfOj3u9qeuTHQlmr01dPT8SzF4g75qGI8HsKPDdo9117tY72lOxuw6m70jynbzpdjBvuB3cDuYNI7jwbwexwpu9Q/DqYvpsfnWD6bhccCWAHM313Uo4EszNN9xL4djBvuDf1ce6dwlfm98pXPUNQCeYpP3Y2Si7bsP0hqsuAh4JZu8cuCsCHAImR2f6Vo0jwezoTN8J95FGZOdGRtdhz5Fg4s4Jeddp5ZFgdCdCdB0qHwkm7bSp62l73tR1a/WVznzY/k39t/8D



Link



Parte 3

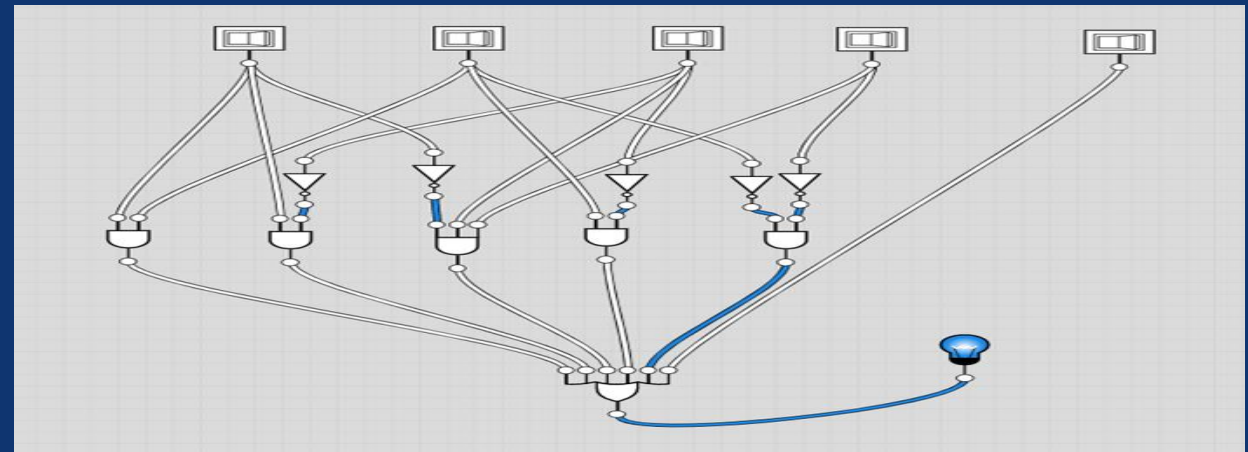
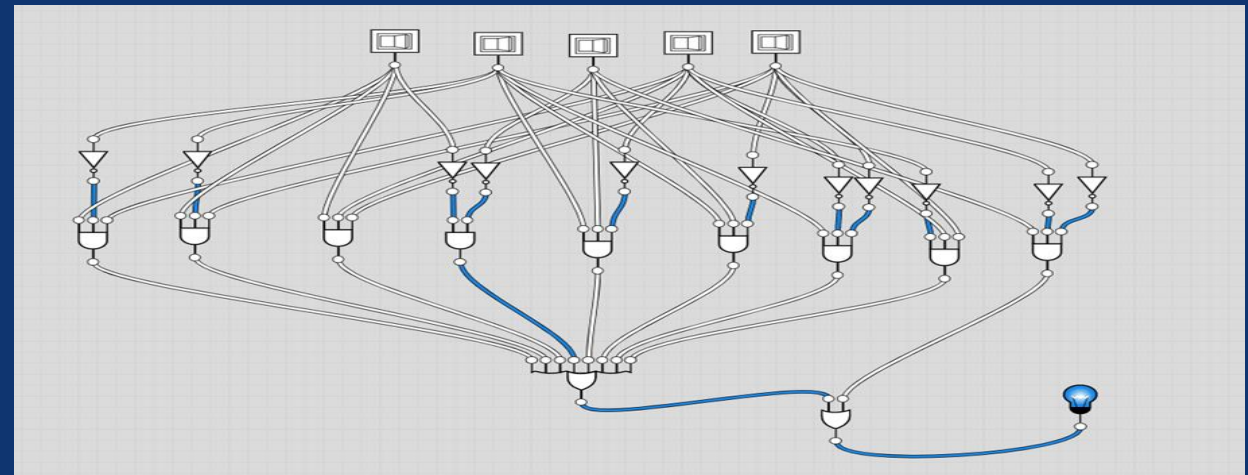
- Agora, 'F' ou display Vertical esquerdo Inferior (VEI): 1011011011 1011011011 1011011011 10
 $(a \wedge \neg b \wedge \neg e) \vee (a \wedge d \wedge \neg e) \vee (\neg a \wedge \neg c \wedge \neg e) \vee (b \wedge c \wedge \neg d \wedge \neg e) \vee (\neg b \wedge d \wedge \neg e)$

-



Parte 3

- Vertical direita superior ou 'C', VDS:
1111100111 1111100111 1111100111 11
 $(a \wedge \neg b \wedge d) \vee (a \wedge \neg b \wedge e) \vee (a \wedge d \wedge e) \vee (\neg a \wedge \neg c) \vee$
 $(b \wedge c \wedge \neg d) \vee (b \wedge c \wedge \neg e) \vee (b \wedge \neg d \wedge \neg e) \vee (\neg b \wedge d \wedge$
 $e) \vee (c \wedge \neg d \wedge \neg e)$
- Vertical direita superior ou 'D', VDI:
1101111111 1101111111 1101111111 11
 $(a \wedge b) \vee (a \wedge \neg c) \vee (\neg a \wedge c \wedge d) \vee (b \wedge \neg c) \vee (\neg b \wedge \neg d) \vee$
 e



Parte 3

data:http://ns.logic.ly/1.6;base64,7V3LrhvHdv0V4YxPtev9CK6CDBwgnmQJSJNOgntcKZMmwjhM7X5V9724PHpJ1d6ubzB1QhmXBksjFV/pr79rV/Mvnr3/9ID//+eGPXz5/+fbx4eenp1//4YcYrnybDr8xf17zBzH2h3/8S/7929PXXz58ibUjw/8m8PH57+/BW/4soZkaJllbMfCnY2qceVmD9Fhrovfrp7L1/RfNT8tr/3ttz495Z/6fmPPHz4/Vp5+JC0qUIZ5y2nGmdPlsxcNaszSaLmGyMDx//+PiguX/48OFHB+EEpVz29Sk+fr65eND4A8f6h+/v3t6V8PH+/V/Hz58/f3p19+/AkvFz9/qww8kGMHfImzQcNUUUmTbGsVhIZNa11rjnLil5gGGMkuQBIJxExPeVDIH98Z3AvNFWG1ZTElyrnV1h0UnBpFRF20KNjKAZb0Lm3dgTPgjaI7bEbL8nci089E5p1kNjYEtBwFBA4R0jU61Usq/IDMeFWCTK5B9frixEXqR3gRVZE9mISQRW0yZlZjWpTmXDSiCeEVm11BLfXNZF/JODKaM0y16hoCHK0U1QOOqly0GxHw2AH2AJWYpDng0px3PCXh/19qk9tZkSHksBVIxPMmoUA4TLTEvPvqZUXurxm3U52G8nfYCCQOJ07MTAqICsdFjNYOTFvnmFehMZPxn1C417UdEhJvP/iMQPMdEdQsJC1VMNQ2HT2iAsJp0QIRMC75GxnL17agNtYrJsfqob3UohG3xhWinU1wglELmpvTOWu5LLUGvtaA7+hjDbiVkuWU1tJrYzpojkiUSF1rkmxW1FjsUnTfSyT2wrcbgsJD4MgG1rwAgoy8DUUX1gy3yajMTJZLvnT6NRLFJ4wrrUWuGfYVncsl2hwgZuEh1sbMHIOReECm9VQjk56ddkVxAniD48vXpPYKmlT4zctCGALJtYVnwgfEJYNOVvrlouyDQM/dU7Sb79v0vAhAUAMIK1Ss3i3NB0cvwqcsarRZUoOeewmCD0Sb2jrM5ZLCO0cSH4SDc4KpiXPbiHRk4N3CVDatJc4fGmqPj2Yyb5l3AeMn/RaM2YGT2Q0UspZBD1SEH5MID22YsQ0EitvC9kfw6NAiPiZ5qk7O8KRaNgaNKCcwoC00AxxkDVCWpsOQm9wZM0I4dU+4bRUAz6OpNeHtue/VlwcUT4g38H5lVXlrvZrF7fnwkYpdZzHehMkA8ka7WQbuB47bWEXYxgtiQcVlxGRtMX9JDM8Ytx5HFIVZAOARpggaJBF0LQU5ShRIKk1k520zXp0dGnYNSuQdySqsqBbRN3CnGD0Y6h1Ircu3yNvEq1qr5Vkd0kdt5Oy9Ri4ztIzy20S3JsJYJopCC2CNa2UOG2/49U3jXJTgGMAHbqig1fMC9eLL8L755JYvIsUEmfz2x3F8ATfTP3pPmQTgSPeJG6VNTHCld1y8hqrCvVay/DEjRhEsdg8M+u/Pkk3dCRCSz+BMvnmT3/9+ek0/+H00n9A2tp4QNLzfe+CPdhUNWcTQ11ddibZ1LjnRH7z5jcihERZxVnYBGyws+euXLwCDP7DI6X/6UocEcX/4acvpf5xePXDhzz8fZrOPvz55a+Ly2BoHlknGH4ZDI2U0OhqS4iGBoPXXQZDE6FEN946T1YmiZCOZeBkMwMwBpFClbGSHuCGbiJxtV7ghkEiM0CL0hddNEy9DY8TtYwMNNAtYm7kMhZbES1DmxDuWYEH2USBew6KHonbiG6izQYvg6FxG5kOtrpPtCn3LTVmuvBDaKYN04cZBOJ9W/kpvWsfU030SQl0TK0efaelrmim2h66E21Zr24umlZQdzRTfRIOKN3ERTiRWZaObaEqRCGYra68XV1cseuvBDLJpa9GjydYbuWm9lB+4aatlaN3FjbJpWwWu6CZad3Ej17lTqozAKfQqNy2Bog8C3YNygt1yZp2D3B2MFmJYS09gRjLoOhSYA9wegBGNJY08w6jYGrfsCWWY0uSiVt3BjGQngU7sCWZUgUkvNih6pOH+CZjPX/NhmV+HxHd81o8ODocF34gv+P4VPf28glUf11IZ5HixpLDGI4l08aF+WNbh8KI5re7det1//PPLfh3swb0tkQXKE9ouf4KEd0DDKq0T0hL992vg9gCBfPvAkQ9SHCR8Y5NlzyTeYQcyn6sgqSaD+cfxkK7fITx5ZLzWlkZBHdBXGPF5rohoc1C5xwzLQQRHum2CpcDh0xBT8ss6B8BDBfArrdVW9Z0Uw0bg5AAyXQx6ODga+zhliWYQ6TkyaqW6NYHk8055UCqclJIVsliWlgoJPEDJkOWG5C5b0GaSdILGElEAFcrrk+PLPXbYOLrYgsEYV09ZK1h3seDBEAaqurRFDNKrybHqWIA74T0st+53beXhKvPvZBUIJHEHQATE40oyT2eghPLRgPvK1RSWGCZSRWQYyuNb0W0EObqkZfVigDfJ01qOAFJ5QqLhqvD24cys+ORW1IHx93LzP4GeQtpIdFeXfvON1T0rfeOKAgQ7PM17GyplfEkwaPzqjYsL7lwx+XLMi7A7LHDVwzmDXz6+1kspbi3DctomiLNPvDtleajeOAtnsjnoG4QDhy6P3eXaeMhPnAv9lIx8mdjaN9Hr2nGEnHUhrrDg1P+OoiNahj7ZQkVjmgSBoKCIthXkQ50FNPdwpALXw9aekCRntlnwzOm/TMW8Jlbw2R/GKZ4VCzqgDAODQW9rdH0q2dKhKXVd1jshlXGkAw7P+wg+mMVRdhloLzFRfmf7Mk+8SRkEGdwtUz8DBKGW6gOBKv56gMKXimwamP+QkDTc0glWbYUR04wCIZ7SjQryuJdJGMA3nvZbdGBfIAOC2gMEk0AU3CJSikEM+YUumobSGHcGRMshnd4iebA5fUaqs21CLyU4xlogQJp6NsYHYnmYyuCQMmMmLePgSQCVQohFYH17h8lQ1c6RDZsMeaBnCCledW6hCVIEGf4YsAeMfQSB7KSCDWTc1VICA3bLepUDHSGCSGV7WwBdLkugLUDzh+xyVf5sxyz6HJqV56bELPbQZl0luUyge2FoloCpmPR0mJEOmIFoZCc4XIXFIEDvNOSc8WIPWW6VZGTdl35cBuhyEMXECFg0D73y99eSWuUMDRFvksesFBTTZBLJLM4UmwjYrJ6Wokig6NPS0U5y37JBWYxqfcl5bSHmJGQaigjlyBV9kDn9tUBOP1TsjJ9BoinhGdVh5BEIuDuQQIT/BgzTJVZkaelUNQZCgXkzxa+17rUlv80pW6N0J8BGGH31OXPUXbNnPWbVGuoQJAKRyUQGOgPZlqLykGdS01nlixKwQcafYmPBZlCFnk9MIJq8ctObUnbWlGaf7EHcasSNagbluLZC0smhfflfl6Ag+TlQcL94jGB6WUaoQ1K7LoA02yU6ktCllYcQdmD6GByQyhrUn5ogfxyzaS0L0LXcSbsmdxJkz7Jggh6ADUFKXQ8o3T8XlkURMvWwglZqB35QGDxphHaDoxY0iGG0GcsB5o3ScfCRgwsDfEueW+u3ATGove0AKSkAuH0VmqLbGvQJlpoSe2yySwikHDeaDOCEYIMBn25jJWXZTUIChnR7oKfLUGF9VvNBYGQacvRGwSkVjWITWPHQ1U1LiQ/TQ753AjkbYXdy6SUV7oQIZfZfJUVnY5XpkF9X5X2j40q5wJVLlGW2qLLSHseiE2nQCsROdhF6ncalZwV90E27aJv1qbQlhmRDD/awK+hKVDy10hmAuLiH3p3VxrQPRIKGdF18BBYaXkvOAUryOAglobKRMF9FzJR8YV33gkAVUJA0/kzyqig4fBXZemfTWclQv2AmVrYi1dGMOFKUDhD0HwaLRwjmH1GTPV0AY0DgkG0owxChXhSimRR3qlysc2m6VV/8haAehLWZoGjLMLG6WAnoh9evFj5aMvGovRyuEF1INE8QfkaPMCSl8pws5Cmc8cAA4d4H3pR2MleCu7hkgHcpzx8ic4f0XM5iCm7rVGHmgudmM22zx8SJKDol5ycVAzn+mrMGQp1BRug90dh5qS45B+ghCe4MMgEd98LmJZidsaT0WlIG1iK63kOkRpR9xylgVyG7ZowDKIE7gA9HVSgnb2fH729zaGzgukgDQd7m32c3CTzrBgccpE6V00CdkXZ7ghmeq1NE32AXg6RhiWBojd1IMLQ+HQIG1khdBKnRc4kTeloffSeYUQowqYvaE8zATTRhPthslRUZ1H1kjdIE8zATbSB9J6pPbAMbfK8J5iBZWJD29E2HKVoEGh4q2XW08HATbTh+2CZIES0vDy6CYa0d7T5eUaUQHGY2zHswgm2jHDIPIJH+rdxYy0bBTcN9YlgtqJxvpU1t7TeuJcrCLsTVmaBKEuoC2sQLTJMiNan9m64YM7TR4l3qdG0y6CM30QadNwpg2jZyk2kT7YnmJGbsD7IE8xklJoYUWbwgq8ObVJgzhqapNebLcpTZpR3oib1o05YnewHswVm7j1YEY3XEIT8sGelOk0hAiGdn6wJ5hBzNCOVvYEM8imrQG8HswV+YzaZn11XNHcfCNxRUAlhhanRh0IKRzGa4lGxgnmBGc2CSBNgtZKDO0fQNPcEMegxtw0nmMGG89qWZ08wgxshTgnlnmBGd2VIQ+U9wXuDfEfbMBNpLPWkzoZuGpAecHRVQMkpxzSav0jhbcdYlWkUS6lKSGP75cNXc6epY0ljtXQBgiBepda5WxKphq56sG2hZfY+Xum7m6BGMGRik5mkpUzU7LJ9LMSGSuXHW49ZuTYClbuJCx+sddabvgF1jcsGfbP0kE4RrgdX4acEKcWU/vSWcx+3tUIfY0lFOUZnG8heUJ8sa6zbZ60soZbS9Ekarqdxsl3goWRChafC4estrUsXkz6HUk3Lcnhx43tX+YMVDSUllRtVhEoVHWK7GcyRtrPOFSbN02saPm8h1J05i3Kk722Pa5uAxAidZua7XU65ZUpAGrcGULSubzLuAvvyU5QhTlpsyr99c+Pcc3kzrBAITJhRWmFH0GC/FmRRA5SXXYblYXQl2EkaZV4QLmtnS4dnFISRDGZiyuotOfTor7nfj3y8AgAYnTlIKkZZSNCoaU4ZeerLklY4N+NdiyQGQOQFq+DxoVu0/3kphQpi8WDqivDuod0JKKwakw7yrzdURmOX6A7JYvWvZ7amxSmf0BDQUA3o3b2Or3OC6m6QirY0+baAi3gtHyzokRpij9f84huCUIGFHbO0eJoaAYoSd6N9BMQUypl7UzxLB4g0Yse+Ny8ruh1AJV9yzYvvdNgVSGWQzNWoXUm5xcBf0kycmkb7PR1lloleQF4KZdEFuNBgqAKCbSUjgkHoL8lguYlZRLdyu9oGhjE8p0467BNNghP0KAoGxGAdSel7Lgr8Z2PYTYtmIv7gbQYjg1qDgkic+dpXlAPi0Dx0VZJG9E25bhcEuPH1eZssbm8rUh6CnKpVISNOEmmmqX0uBXrUfCdm+DA1hNuq3YvMMMP+cZvGktDKoqq45gNB5snAUMipq6DCIlfii8WgS8nP0xNE5DvoN+dtFavhXbk3ZFMAjabBiPMJCWujds6JEMFHduNQuWeAYPtK3cGY0SDT7F53G0W0bVktTfSjVUj9gTswWgbxyYFgp+51atZQbp8+geW+bbY+T9YpnH3glCnSyyWUtKzoU6x3XENIGUD2Dh55Kn0li0h2abbaKwDSKwUaDI4Cgy4AnCkXoMLVXKHZlNfnTWfHgzLdwm34K8mrcGHfdQ3HrQHISQlKGceSDGXpC39+oL842bXwEFJ1YxY+X08nzA5pZmH3GatEbvNLg0qZ25YGCwx3F1y18RHNxWYXCRVXQxOoX38KhjclSS+dvmosYfM8+Fctoi+g3Ec30xTt7/JDfFbZ3xhllvShVbDW+as/QCfa5SoGpJY+Yxp7fGZbZlZpy7oOm0S6l2KNqdw5ijRpYrWM17YrWsdh414LSVHMbYifF8ujnpMvsyhyY0m5l2KEseezX5DzuRVZK7dL0lAX3rDubBl0j1MQBEI8ujh+5gHNCavfBhRGKJHdGvmdia/G8dvN3CqG82dbXaEtp2KzYkYN25Sr1h0oBpm5SXC2MrTWaNDNFASurjMPLROH+YfFigs0PyZqtSk3KspXhbrhQ1ag84hdo7VN71YvxtMRO4uBeQGp96HOfkBPNPiZU0vDAkhhgAlek0lHhaadkTzGjgEGQlwgknqdgWgHPkS+GAw4SR1GkQwNOleUnUxJhJizq9gQzsaXNIQ60WUMjJNMvFyH9YnDl4CSSMWOxtcXcYeYntqH8J50MHPmaM2k+OdrNo3Abdf+BNNLd0zJXdN6oryim2ist7QM7XsgT8tcU02wRN1ZZZjAK8HMzCh21uWg/mim6kUIUIGNpxx6DokZQlFczG1F6v9E21ZqNl1oMZBPDWBkJP6Bu5ab0Gvqk1o05Yp2hdRc3ctNQX5FOqC1OjdbVqrcyM30vqdG8NO9X3TFbNPzPgrEu6vumK2UTrSG+UTbSOlDoslgZCL4OhDw+prE3y+Z5gRpMkUc4kBAkYXBWl9255gBiuUNHW/J5jBcildLonmNHaLUnD7gmdPGBPNT2BD0owDRxtCqYobieRHR7ghl+4Qolal36JspR1YodYmILDneACQ37l4UfHyhm8/UgRH8kOULAAt7peQdy+2ZzJhYtXyXVRiQXewweiGR2SRGDdd4AH08BCGHu0BPYmkkh72C0DhZfg43aby+8KsifdW9su4PD1d/PCrDXNJtBoB3ONUPwU/m/aG+lmf5SEP9nwbD06ggWjokBzvlvVcf4hErpYVYQ31fsQl30p9ysilPz88FdnCpJ50SOWuDGZ5dl4qJHalGRmYsPXUjQy+NuxRkQovLjNemNG2glOys5PNR6Q5AF0PSvrmWBbpqSUq1kuoa+P81087u7strmYb33M5h1R/qnAr84ejYmYrNIOXRBS5Xl6Z0nSKQpqrjne10iRfU4d08zwo09T7YcCayStaP+od7pfHdWKGzKfduUbdRyBlAIElIaFPvNxyvCXVvxaR1SaMDH0qzCHV5JXOX2XSspv9dR53b9Cxp1SLN+L6d/kYzr32JHxuxT684Vo7H5YvV5WTRUUhVuxc870+RZiCkYfYbXoyFPX01OkY0Rk0tSO5jEYBujGAI5gg0iSAeLnxj03PCuVycrJr8hnB8lB02R8gzULmov7ri00RlVbsu3UdWvblpVof7ZdacqJRmjYtkK+T9S53eoFkeE/sPn+iuifi0UBka7HOFCTKLpScv58JyQEEKraUSwThXh2u2AN8QaNO2ebFUt51b9szAel82gcQDQAUMmf4QnYI3TGGpDs8a7XGOSTha9Q5tKw38TNePl+gvjBhL7N4GXGUt/owN5ok0nitVoeRJFq7kVWoptG3kax95oLk+h2Ftt7ZBJT72Oqm9WBG02dSal4GQyMRatEjpebg+1ElSR52daYWW+ZLcxsXXpYbZkHu2vBzNKBvIFHzymkzQZEQYtg60T0hybo3Sagge4K55gbGajcNLEOruHkq8hXca1O7AlmcdXCS809wYyOT0g+3xPMaGuHAI7ghH3Q4l1htW36f19en+3t+n9/fp/X16f5/e36f39+n9fXp/n97fp/36f19en+3t+n9/fp/X16f5/e36f39+n9fXp/q+n9y5j6wA=



VDI-Link

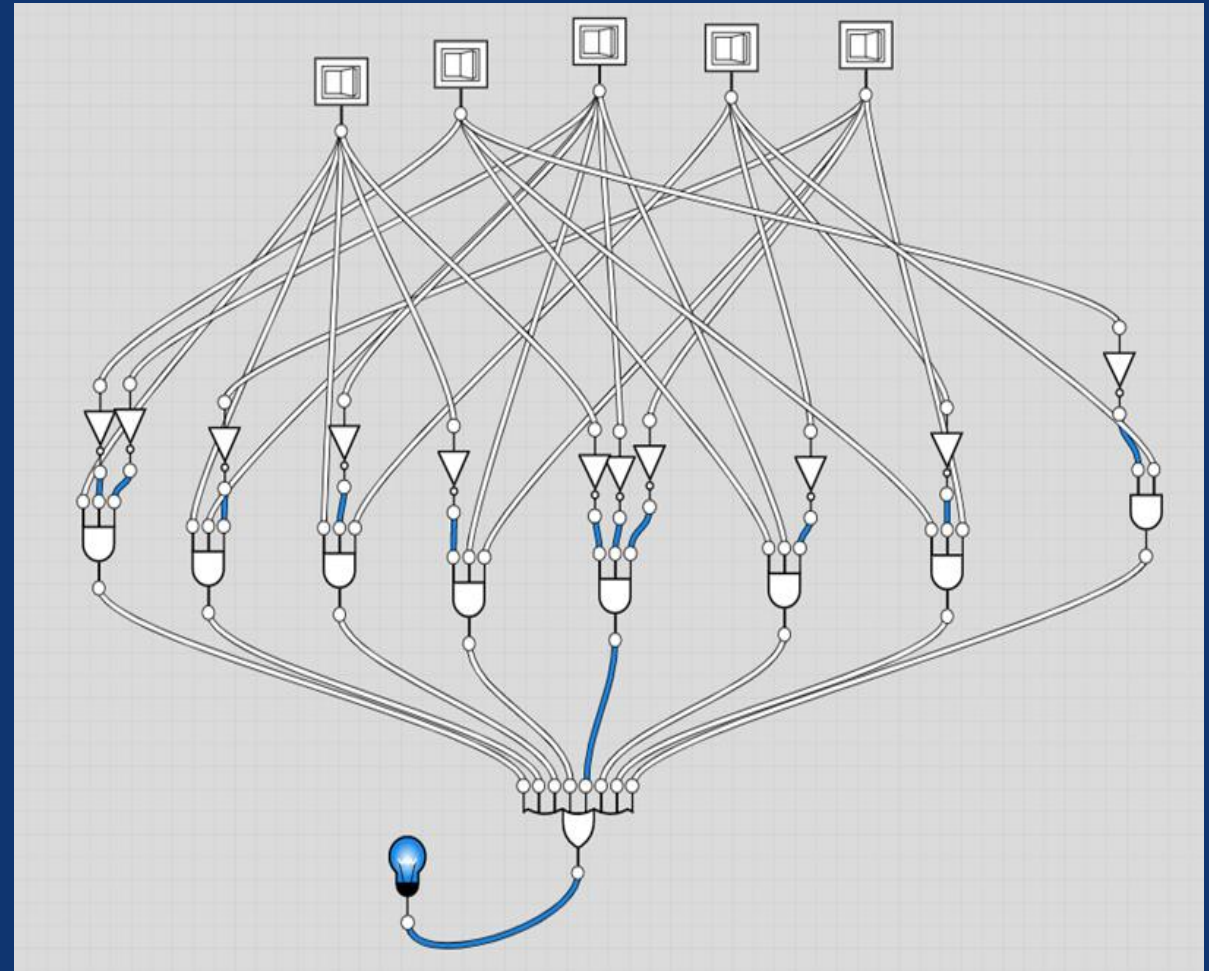


Parte 3

data:http://ns.logic.ly/1.6:base64,v3V3LjhzHdwwYtBtMUj5PZnqXhnc24LVxxc3Rj6rJfaAcORzall/ekV01wx12d91TrOqmlGogznRHn0dEnKxTxb98evzb/Lpz3ffv0+cuHh1+fn7/h19+ +x1Ov3B9OnPX9RED//4IL1y9PjP+8+p9/ah4d/feh009/c6ypPFO5UQJL+EDabjg/sFOC3qkKyZrw0910e83+39rS89pf/fhUfz2n5295ePf1Y/3wkK1yrggyCUprM1BpBSi6ETfFZUypT7wtuHByvDw7s/Pzwo/Duj8en9PTx8IOHhygf3VvVz+8fRvp4/yfwVHr8+771Cb86+VSlPfZCghF96pmiFb7JLKxzXgSmkyDfe5dB+rmz0CYbTZltnHPS8fUvtrws208CC86SdbqJLJUV/vggktdKaG2gheocpXICRc0Q8wFmqPAGUjgl0hgz8JDLrO/LaW9Gic8JqgJual9lpgSvZuspnnpD5YfG6QS3/ucnkbbkqw4+qgbFIZjJOLuVuRSZL66du/zCVIU35G5Lcy30WWPtezzBnnbU9BoOCRSJtUcEAUtak8RNv8c0J9gkYqTddcVspJOvnyS/H6PnGYXhIPBN6ZKKFbElpCz0LElJk4wpeHfTkt2DkygKSwlU2CdwchIBMrpVDQWDYKS96LYGIXruA/ScpgWz8hCGZ88BmBIQciaDU6qk0LV0fatgTUuZUSKie41KtoKtLRN9YN1ExyHoe8ByBID0qgDgGf8TEPhkg3NN+lrqwjX0PQbhwBhI0jkn0FtrOasrUQlarQuauw7qZYqLaSzXpXEN9hCKqMuIalbR8KCDr6NFZbRxeSsjNFujTvvb2eyWqA7PgPW5SCVUrw69QF2AYIMliY2UenR67gW/DNLCK8n+70ZryPOZwg+Pz79IKBbwmGD1KMENsOxY0qZJRFN8GUVpSIBYgdtU9bP9hr978KQHEAKFJmJmJIMyG3vkpSbBAb1WUEpCQRDz21+wdhONH4UHCiA2E4EbfJb6KFO2SUB3czDfZ92IVC/ShR745F4r7wlmTPY1GLdJiaPmjSES8AMN5eE1ysM64ajQKtfdHSH8MVC640KtFukuxHJOSSgR+ilXgCKAZYK9Zs4KvcIwXwKi3XumbZYOBJLxakvL8NgjxJBnjBvOPZym5ZeBzP6LRLZWOmedvHdhc06AksayNYN28DVIYLLk4JqIHk5Izyc2EHt9g/Blc/ZZZS4wHkWam0MjwdtCFcfncvYgdHFHBO9d0O5FXcw5GHNoEdismvFO2y9Od2a3BzoFpVzrC3WbZ1ZLVFH9QgaEpJOSLXnzKBfS3SxazhSOhasEIQM73OpdNcPL2ofE+awGmBeAGx9glWjAjkDJfFeYZXk4lSULqYn7NzW8ITSPI5wNH9qGFAv2IYCDycbBERGN0e+tmCDjkrXEm94YnDucvJVKF5iCWdYpN38269P/5Wf5pn/Ad1s1pEKXsYc5IEuyPrpYRFpolajNKP9m3eIMxKQORvO+8fAY+OEwL4+TPA4BsW/O/ff/1nndMMP/DxZ7V907366UOelp7ns0/fvY4ug6GpSFHgpHxwBE6ToYnutiguGJ0nUwBPBKTBNPK66D4Xny12D0dTA820BwxpRmGB4nOkmJUI8bT6SDArBcyTgOlgeNM0MzI8dbwOhjdYmYPDM0yug+HNvzI8E4lj2TgIW7nROukB5L25hp4p0NXgdD0za2HOxL03ZVWknT3shsB7NswLzjyJVuYqn+ndK0XbVvmCaeBWF6GhneefWRkbpgrmnH+6E89sN1c37KbYV6Y6YJp5TVFOaeE6RO6rsTBPPKTLB7FXI7ebghS3H6xKN+0IP25tvVOathryITTIQXuvrhTN22PzA3TXIsu7gTa20eVNTCsUeM6GN5B4GswfmXWZnnYI8HOypkeyxwdCcZdB80zAEeCsSigWfEcaR4lx18HwtOVIMGsnVyzSOHLMmu1k8sRYYNymPVikGTH0wA/PpsZxO89+NCwR/EsHp4sFX5gy+OMfG95OU4Rv9r3jeco+Ve45cyGeJLJ7UL86HL6a1+paWIK7z3952a9DPCSgmKREPwnryfleAcMrJa8UwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNldGBztJ6YVKjnYMG66A0hUwSPHAdm5DolyCCCTC1MeEHDXRZD459ARBjVfVtkJgX+kh20jIsV8X5KQP1pC0SvqobrF8V1WIEkOZNONk0MqL3LoSOUrkNKacw7KOMq4AiIhplkmXOK6bQROHOB8TqBjwTYg8PZYNOL0io5zCpKTRf8CZAUpe0bVJxvKvOHKcWThEKslmOmndkDOZ45wLcglWgTLBnly+P2MmTmPYVVRiqnEJkHmJMFQbuaFAU0M8pgphA8AhaNfthM2yP5995K9beZPQY0eyHABKaeLNaqZuzRq2WjkeykJValmITtax+0O2pNrOcJ0waWYUAD85ApEABBFJIBGfJvVzCF6rid5IK7gNYSnnpX3b2LRmBaTL0bnV1veF8T+nSakOHgdyRKID6zuYL+msHcnJh9xVetrZAmahEm8H/D4KMGBowf7d3aGmOCExp3yaKIYzEra9hEcB3XOEnwG4wDjK1MIZU6ecZOUC9p91BmNld

Parte 3

- Horizontal superior, 'B' ou HS:
1011011111 1011011111 1011011111 10
 $(a \wedge \neg b \wedge \neg c) \vee (a \wedge c \wedge \neg e) \vee (a \wedge \neg c \wedge d) \vee$
 $\neg a \wedge c \wedge e) \vee (\neg a \wedge \neg c \wedge \neg e) \vee (b \wedge c \wedge \neg d) \vee$
 $(b \wedge \neg d \wedge e) \vee (\neg b \wedge d)$



Parte 3

- data:http://ns.logic.ly/1.6;base64,vZrbhbpJEIzfBXHf4z5WdUdxtBd7kwfY61UfY69Yi GKsjd9+a5iJDcEzKsRgmbmxhoD/q8P9VLX/e7L495s3L6ue/m+3T/fphv//+6e5u+9Qd/tBtXu5UB+svn3fnp5r3q/3L93q/fvrvcZ8f/vj1kvXq+bHcr5N1VRnnBVqQwtrkRYxBiga QXVYxQYzr1c/7tZV+vXq5Xytr37s9nH/uNver4Ncr3bP++/Pe+JocfNU13escwPGliB YgTVFYZ1DEauOARc1Jr3EZPTThXKdNpw8n606F44dbhsQ7C9bpKmJSWtiMRUT USmhtioLiHcr8IAEHFNWeRCh/wul9LIJi0UdEtKIGg8JqKCJYCylkWCKapIKVBxT 05hVF3wTFRVm0xyCKrkiYlbTwulmRcpbN6tIQ0wElqDcUdyVK3JazkBiHtkUvqEop O6o6QvBatBYDFWp2RHngUBA67Q4gVspOOvn60L9jPG4HCsMhwKp0jtmKUCMI xbcKfAUrjMI0uqnZRTtEwkPnRwJtO49ulQlwoUD0mZrDBmEBUXgTmnCZfoQiva3t QOBN/8UHAisXJKgIOChVC1caqUT2VAiakhKNUoFOyRmGLAC8xcC6DpYjaHQO GEP17ouwxkjhlxFgtN65KrHkMuofVMXALxgDCTqISBJVam3CBiupEjU1J7ikG6ga C4xKqV4rsRevxQiKDEFSN4jmFRE0asxQbBHNSUjOZOnCoNUa7VslqgWz4LBgyl IJ1YqjXoAmSDRTHxlbILbg9NALSuEvWTCoo/vWjNME+oxgu9v/TtAs0HemHoQQ MsUAswjKJyqLZrzJNSsLI4EdDEtb7OD4/EkAxQFQoEyySCIOgnT4LUohqyc1KDn nMIYg2M4cy+EQDh2AV5QsGkmmGbAalXUvTslGQd5ZhMmutSSloilZ5dF17tg9Rx jf2WMydxWNg4rGAajy9ErlgZrKwzrhoFGDxCrbaKXav0djyMuWC02yBUE2l6JzIKg SUAQDZPPkGKRaoSQTxn6VHb4Do05b98zMLqLxMrpaEx0vfa8eOZB40sRFzh5S 1RK1N4N+SXivblwpx31sr8lhP1GUI6Dxi0zdeqspLk4JKEkHE6Mz0Q2CHk5NfQyO PIVWcxUMUpGm6qiRaBilsYdyFkooQmeVjUd0XrtXdzHnMGbRIm7JVIPktkmioboh eSOdl6lVufYjapJpmEzB6ndpYMHQ5Oy9pV4WSnnqbx1CNGBUMXQ3A4OWxnK xjt5+9AgRk0eQ2NAIo2x1RvhFfbis+Xpc8km5HEUuu/2N57U8AWH2f04WyA0quCp XrQtQDocKWf9ZHR1gKV663UYiyZ06nQafOmHedmpl1eYZPFnLJvHbw/7v9PzJp 3pH7mTVT6l1Hy/TElqd4XUVZJ2i+prAaiD5Gg8OX1gQhKiE82T/dl5t93S4fTEOD7/ 9fVPrvAPb/i6LfVn/2nDlZq8nzdXH14/vl1Nw/A8cUmYmcjwTGgahjdlMWF4JjQNwxs 6mWniedA0DG8GP4bR0zA8mZ2BYa0kTBieAy0JM5MmnjcvCTNTwDzJn4bhbc/M yPDccBqGt0gzl8PbRKZheLSUMzK8a7wIFXimm3h3ezOix/I2Zpp413vTMDxvY9vBd Wm63Jtm0nRtZC6HmSlg3n3jTDexXP+D0nS5a98wTbwRhBkZ3g31kpG5YZp489A H6czlw9UNu+lymBumiTcplCaeJMid1W5Mk28SZEJc61rXz5c3VD0LoeZ6aZrRY83 tn5Qmi4fyGfSdG1keNvFB3XT5ZG5YZp428UHufblq8ocDGvVmlbhXfwdw+DMrs2a YZeEgZk7PdZwtCSMm4bhjQBLwtgZGNaVxplwZhqG5y1LwszdXLFEa0mYubGTP RNLwswpMOvDZkSPdZI/BnM3/nPWl/8B

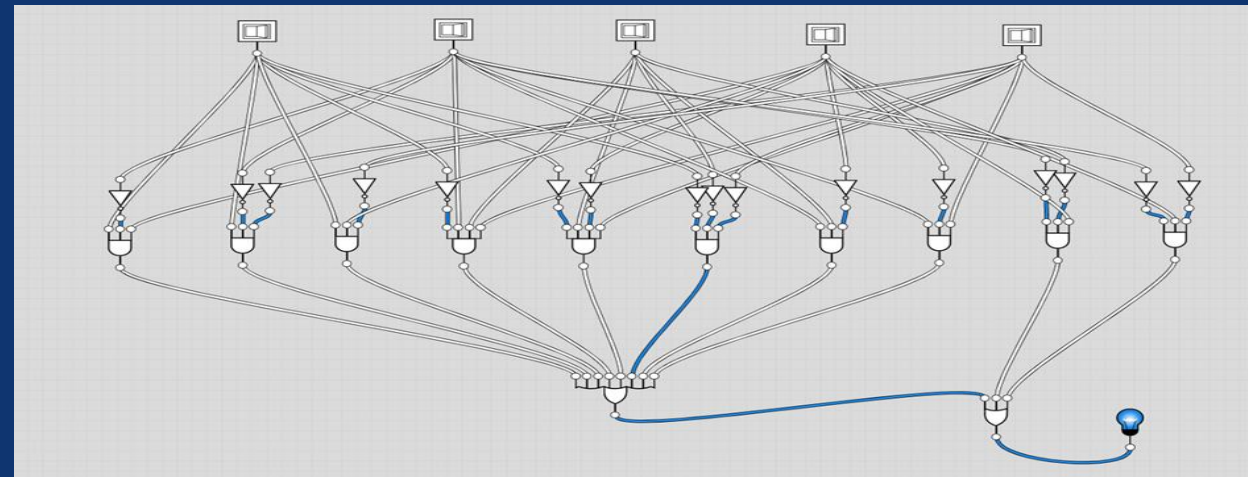
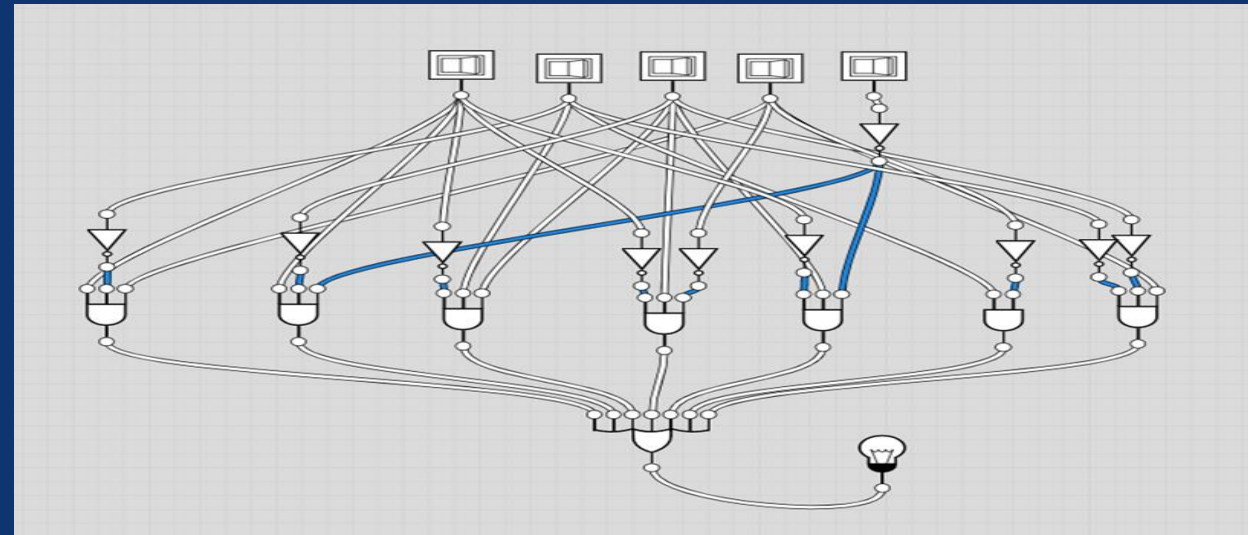


Link



Parte 3

- Horizontal central, 'G' ou HC:
0011111011 0011111011 0011111011 00
 $(a \wedge \neg b \wedge d) \vee (a \wedge \neg c \wedge \neg e) \vee (\neg a \wedge b \wedge c) \vee (\neg a \wedge c \wedge \neg d) \vee (\neg a \wedge c \wedge \neg e) \vee (b \wedge \neg d) \vee (\neg b \wedge \neg c \wedge d)$
- Horizontal inferior, 'E' ou HI:
1011011011 1011011011 1011011011 10
 $(a \wedge \neg b \wedge d) \vee (a \wedge \neg b \wedge \neg e) \vee (a \wedge d \wedge \neg e) \vee (\neg a \wedge b \wedge c \wedge e) \vee (\neg a \wedge c \wedge \neg d \wedge e) \vee (\neg a \wedge \neg c \wedge \neg e) \vee (b \wedge c \wedge \neg d) \vee (b \wedge \neg d \wedge e) \vee (\neg b \wedge \neg c \wedge d) \vee (\neg b \wedge d \wedge \neg e)$



Parte 3

data:http://ns.logic.ly/1.6;base64,vZpLbXvJFYX/isC1q13vRzAaDJAeIDfZJdugnmMFMmIYVCLn1+c0u2U3S
bHnUt2SFoJgS+TH+zj3VN3+5X73+12+/37z9OV++3C7+bzff/3Tx4/bh+7wH93994+is5tff8mPD/vdl5tt/FJvN//
89NfNzf77V/zlg1ZB8cRc04bpFjyLSlimchZGJ9WEEVhz6tvs0r9r3o+v/fDfu33+/Nvzr2xuHu/K7aZIUazyleGtG
9MyNeabtaWGn2OJOqVsnJdPtupZMenX2pz8/12o0QXpl9uc/Ntt4/7u932dhP45qY+fd192//98En/t7nZPe6/P
u4B3eL9Q918pEFy70Ufh+YqZ9oWxVLhgTKRtbDWSuncAdIYe4wzMKpOOv3zH+UcY3olY86J18Qjsykkpn0
fSB8kC16ZGF3mrskDo3amU+eMttNTRJHH+J9XMLapa006Mm24Ylppy5BhzXSUWbSCt23pwKiC78yOOZ
XMqaal28c1Z8sCjKYwny2gvmiEeHkUvZDHFEP+X6mjg+/SFj3JZTQJ9C86VFJqUJ6BgfmOeysJKrNkZFG
bl4APpwGc/AkrV+UkbnQHebQcgRYHh1SVlikcr2IKMWsBE7pnlRmZvjdyMh2CsDR2fxMX/gEE9/oTRi2iK1
yaG2phUXCI0QRBDGCMFs7xLjFzPoiJtp2d0DyHxqPAftKc1ddVMDWbzMETAoVoQkoJF0c4054G5P3yp
ehkLw4Cs1zsbvOTOzCEY0G6sJmaH9kKcGjmCbZSmXVJxRuoShqo1WnTvPk5Zd+MnSa+LMPoMZrvb
n8l0x2MVibJcDNLEi2QxucCCtEr7KmwldahgfawC+gAjuTuaBOJM859pBIUGICajMlgLFalkHYrGZo88Vc5R
Q6ppO4QGE+hclA6INM2TWURJRAY+YR6KCBcd0Oc+J4zHlkLxohkbh1FjNepjIBD6mOBidkgE0ZccFWoD+
oJBliCFmMgQ6SY1prVdgJ7rVnZylJCO2Ma0jBhuHS+BqZAQA58rSwZdXbzMAQMjaC7GYn1pTAihjtnWT
ykyZA+PrMUBelR4QACksO0zk2pgL4KQ30o4ztxH8SwjelsXSa3bez6aQhUhySb3k/nXzgLAoYpxa18cmqa
uQIE+yx3g69Y/TFZJFnb39/9/vn/b/S4306Mxu6ylCrZhyOB4YooUaK4ownHoTmtuk6GiLhuzDp1aFeDBpbX5
pBjzOydcAy7vtFmT4jXFK/uPTX6hKMvzBp22pT4eXP3ziw9/TZunh98c/F5dhaC7xMgyt7Kcw/DIMrWovw9
Cm1xRGXoahScqaMDNpoinSRia5SFghJZ8ZmqGNNepBUw6i63ZTTORofnwywA0aSB2E60114zMDMzS
NNFcMjFNtGPnmjAzBbw0TdfDvGGaaApOhKGdvC/D0BT8nbp+nEwA7NUZ65XYPV2aboe5g1bmzbb3qlm
ro/MG9YMzYIQYZam6Xo/84YKfD3MTGRoU/cyDO2wNYXRM+OANFvWwJnTGZKCrgkz54FJrbkmzEwB0
wpwTZg5nSG92GUY2jH9DOZ+lw9n8Zv+qP+tvwQ4HPsfiC94+gr3tf14AYop+kA5SH+gyN4Hjyz9QFJzfKiP4
9oLPw37M9l+wReRRE6BZaHR95E7FlufSFeU5twaZYfLTiVN5+XkymZccIXOTy9kz+5vXrfllyDY4DVXMaHd
8cFVZLL0UBG2FM9XrNl4zGn5ENV/w5atGJ6Y312U35q6AkWizk6liTDola3JAtw1mp1kARsuRFD7dKKnRy
GqoBSndqCnV2Y/46KB9Ss1BrgeJGHUXOvABj0apklyMXlg57tSA7PY2UfsNQQb/huVH4tVRUdxNoRAMqt
ERtRWQTzXBJC6v+Uv4WhuryzWCwlbvsl+MS33QSDm1ZYfp4c4WXWNW4hrRCdk4e7Tf6m0EEUV7aep
zcDP7tz6TnHw9Ba4xcKwqKCocmFjBqmSrRWTBa64bLZC2Pb2/HNYwzR+syuWxDfarzIYblam4QHFQ1S
9JG5qKp0AfjqxvUAPOM8+dLIX+8L1u2oUI2MGCKYK1K3ht0A501huUgVMY6NZ3jKE2ys2YkEEeX68sldJl
Zl04xVvtviirKWFeBMrYtleoiN0PHe6NeCofzR5uPZTCuGV2LdUyWjGM2Jg2LGDVMMNpgC3WJ1bthliqCOF3
bmR6mslxyYMutq30NceNRt01Do3kk3xYN3laschro9XR8+J2rNZWZUvfr+RlhkMBqbxXxfgFSQOUmXnRm
7yPWbwvMu8sdL54swkrKRQZ/oUJNksByIT/RoaetgEICzTjkHmx3HHZk53TMc9kO2s1OYZRu7GjL0c8kyV
2AWdSyGRZzhUM049koueGt83Gaq542dOyFYBJAV5pGEncoeaguTl1hymAUS9QKipkQs41MVXHTJctUb
WheTABJsm+mNjYehA4CMaJzD0Qa+J7je7gyiZvhJcYw04aRxiVE4QghDBxX1TbDGeKbr/ASPBpvtMc3M
9aqeXGh6zWtcUgwmisVbZ49wKFy7CpselnEaJ0OPTDgA6iotzxUznjZtUSu5i2XC7JVGfJwxhDZBLGYEjw
D4pbJebIYkuDpnjx4irTS5rcmC4KQ2OQDCjHE5aKXmWYs4Mlp6+ABgFqalPANE5V1ctVsyw00fmoK0oolt4
Gp8RCwxmHO1VD4c2750e8vH0pN0jo9fJUZPDCysJ4gTfQCIMgSZxiKlfbGDRY7XfaTwdzc1w1z/qjlnbOV9
Ccr5yzylJyyEtRQSAy8C1JwZ8njKdYssXZYXx0pJcXfZan/pm4qQhfhEzB2HSceXyQZg2MYinclpLz1xFkh
wpeWdHKZXLMDTTQ4ShTakZ/QvJkFiXiKSpNFMzJANGrBma2K0JmXmZmtZdhqEdYogwtGk9s5kiOVNim
mgDciZnPMMMTI0Y7kmzEx0r66x1hS9uc0U6fZjJlKbac+aUO6tVozMnMrj4Vpun4czKSJdnU2Mw5Is406t
UkWMY3lvGGaalOWukBc2E3XT+0ZnVlcM1fDvGGaaFOfqiJMLi3O9hZhL00LRO5mzd0oTzQ+9U5quN1cza
VoqetfDvKHOXO/03rCAabb1nSJDs63vVMA02/pO5up6DzxnIUgVnJO1STcpRBhaN6wJM9NNNJ1YE2bGdtl
G3ZowM0/a0PZImjAzyDRPOyaMGbmsR/SuWdNGDujwKQXmxmUpLX3Wwv/eKTK/w==



HC-Link



[illegible]

HI-Link



Parte 4

- Agora que temos cada um dos displays, só precisamos montá-los em ordem. Pegamos cada circuito e criamos uma caixinha personalizada. Assim todas as portas lógicas são transformadas em uma só, que recebe sinais invés de interruptores e envia sinais ao invés de lâmpadas. É importante perceber que como fizemos só as unidades, temos que pensar em algum meio de representar as dezenas (que estão presentes a partir de 10 e até 18), podemos ver assim, que temos os valores 01010 e 10010. Como bolamos essa lógica? Podemos perceber que é fácil identificar se o número está entre 16 e 18, esses obviamente terão a dezena. Mas os valores entre 10 e 15, como percebemos isso? Vemos que o 10 é apenas o terceiro valor a ter o algarismo de milhar, se pudéssemos ignorar o 8 e o 9.. se o número tem a dezena de milhar é maior que 9, se o número tem o milhar é entre 8 e o máximo. Como fazemos para ignorar esse 8 e 9? É possível depois de perceber que o 8 só tem zeros e o 9 apenas um 1 (unidade), vamos seguir a lógica:

BINARIOS	DECIMAL	PALAVRA
00000	0	
00001	1	A
00010	2	B
00011	3	C
00100	4	D
00101	5	E
00110	6	F
00111	7	G
01000	8	H
01001	9	I
01010	10	J
01011	11	K
01100	12	L
01101	13	M
01110	14	N
01111	15	O

BINARIOS	DECIMAL	PALAVRA
10000	16	P
10001	17	Q
10010	18	R
10011	19	S
10100	20	T
10101	21	U
10110	22	V
10111	23	W
11000	24	X
11001	25	Y
11010	26	Z
11011	27	?
11100	28	!
11101	29	.
11110	30	,
11111	31	;



Parte 4

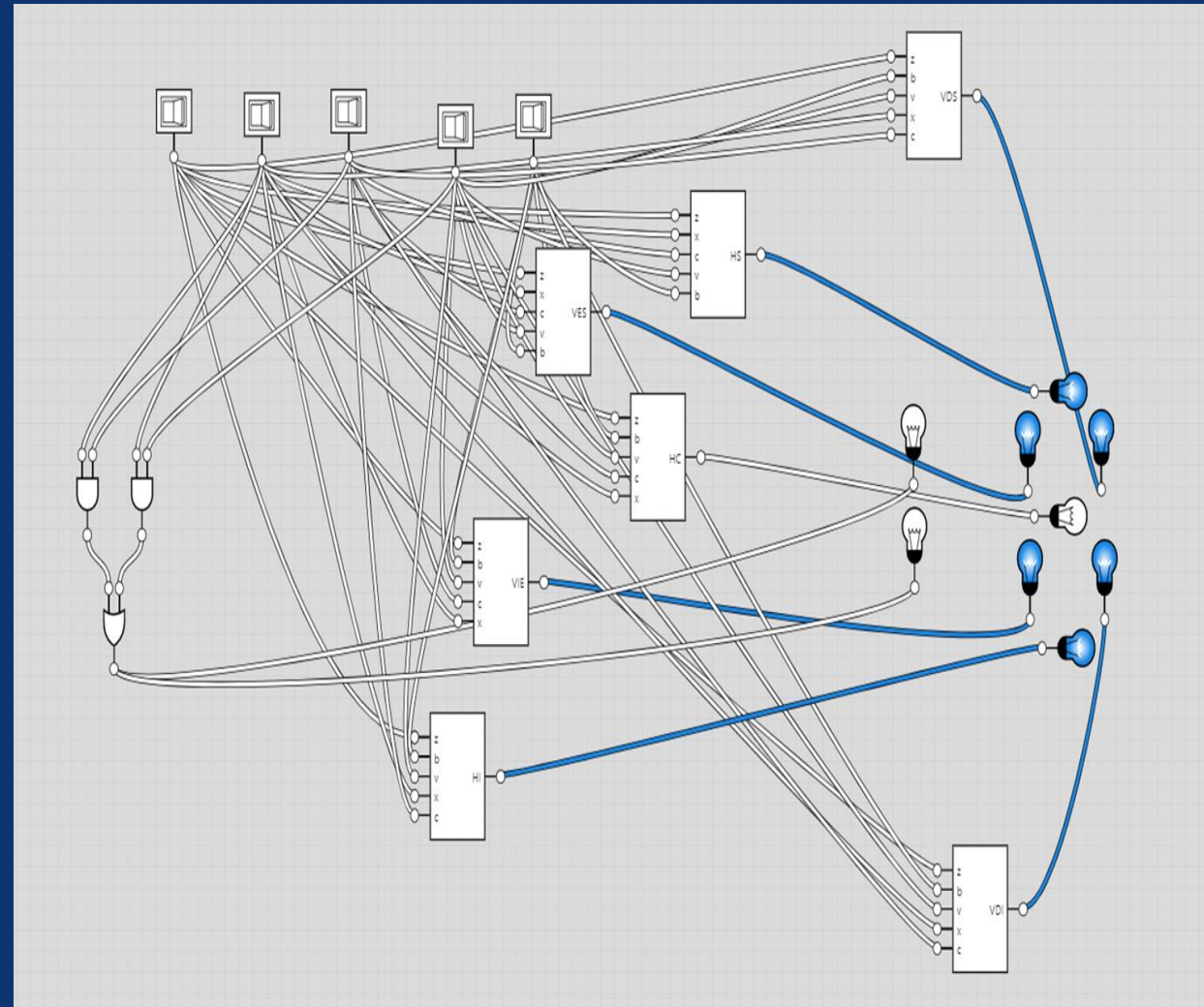
- Eu estou pensando em um número binário. A maior casa dele é a milhar, que é 1. Se ele não tiver a casa das centenas como 1, a das dezenas será ALTA, e vice-versa. Perceba, que esse número obrigatoriamente está entre 10 e 15, pois o 8 e o 9 não podem ter nem a dezena nem a centena como 1, e meu número precisa ter ou um ou o outro.

00111	7	G
01000	8	H
01001	9	I
01010	10	J
01011	11	K
01100	12	L
01101	13	M
01110	14	N
01111	15	O



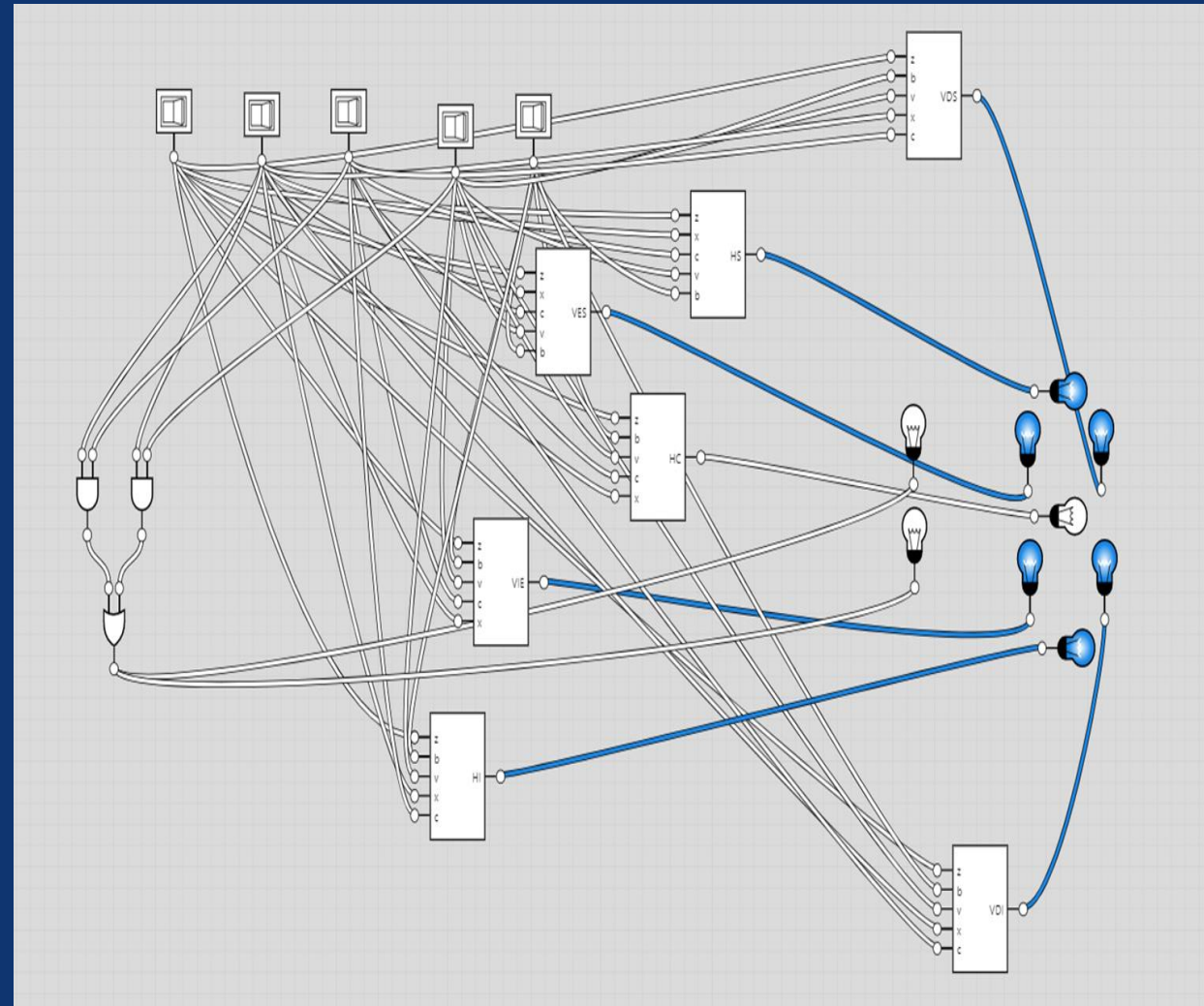
Parte 4

- Isso foi simples né? Agora está tudo completo, vamos ver o resultado: (o sistema das dezenas não está perfeito, pois nesse caso não há limites para entrada, é possível inserir um número maior que 18)



Parte 4

- Para o decoder, é simples a lógica que devemos usar. Mas antes vamos definir algumas coisas, primeiro: O usuário deve desenhar o dígito nos displays e isso deve ser traduzido para um número binário de 4 dígitos, entretanto, faremos com que desenhos que não existem darão o valor 0000. Segundo: apesar de um pouco grande e bagunçado, faremos usando a mesma ideia do encoder. Vamos lá!
- Como vai funcionar? Bom, a resposta é óbvia, com uma tabela verdade gigante. Deixe-me explicar, lembra daquela ordem que usamos ABCDEFG nos dígitos? Vamos pegar o valor 9 por exemplo, o 9 não tem o display vertical esquerdo inferior (VEI ou VIE), que sabemos que é o F, então se o desenho do número 9 fosse ser um número, seria 1111101, certo? Ou VVVVFV. O 9 é 1001 em binário, assim como eram 7 displays, também são 4 dígitos, então teremos 4 circuitos, um para cada algarismo. Então como são 7 variáveis, teremos uma tabela de 128 linhas, com os displays acesos pelo usuário representando um número entre 0 e 127, onde apenas 9 dessas combinações entre os displays representa um número real em binário, e os outros 119 representam o número 0 (apesar de 118 não existirem).



Parte 4

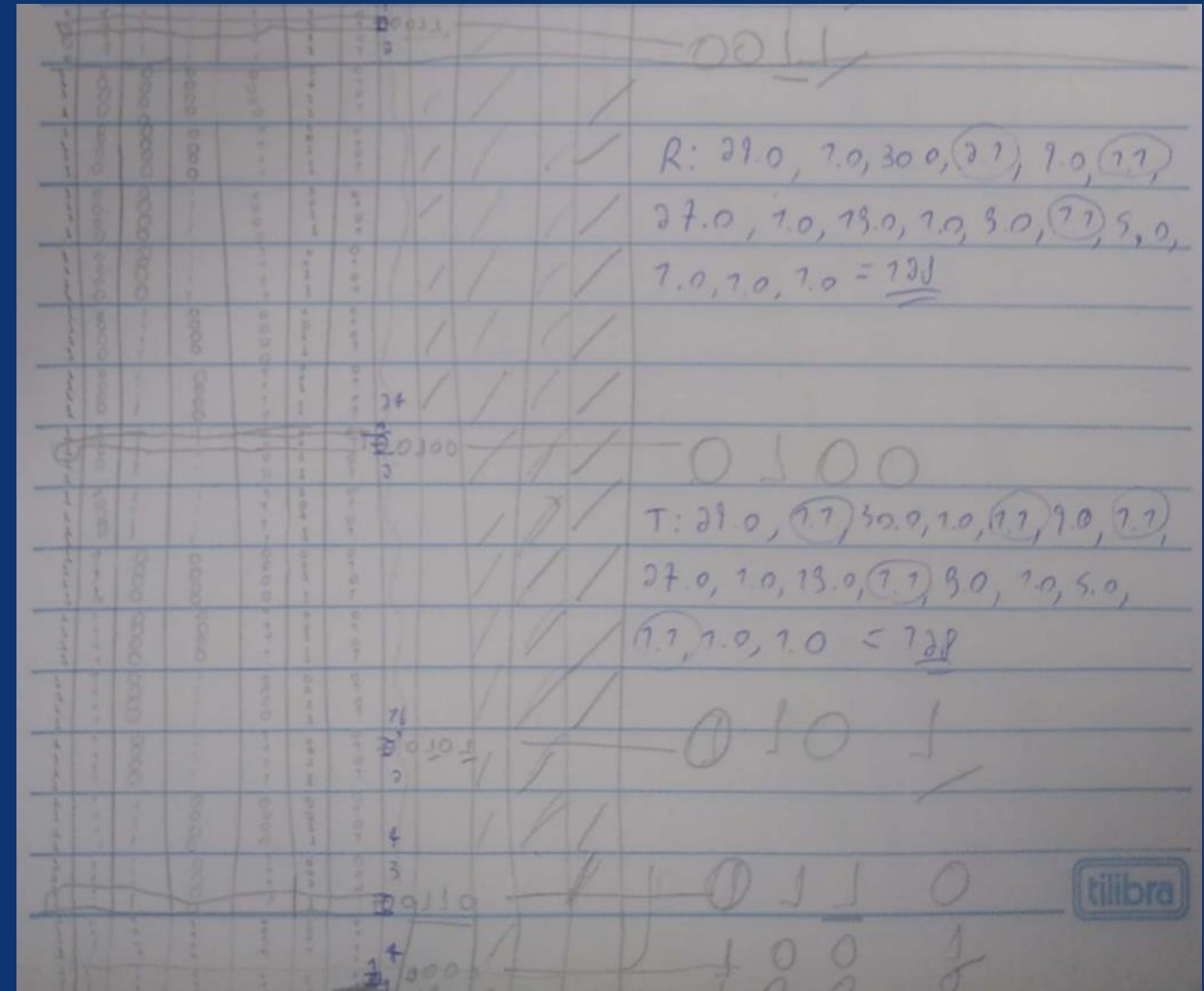
- Lembra que precisávamos da tabela verdade por escrito para colar no site? Bom, aqui estamos montando ela, já que cada uma tem no mínimo 119 zeros, temos que descobrir em que posição os outros valores ficam:

Handwritten notes on lined paper, including a date stamp "05/09/22" and a table with columns labeled A, B, C, D, E, F, G, W, E, A, T. The table contains various numbers and symbols, including "W: 0000, 0000, 0000, 0000", "E: 21.0, 2.0, 30.0, 2.0, 17, 9.0", and "W: 21.0, 2.0, 27.0, 2.0, 2.0, 9.0". There are also some calculations like "5.0, 2.0, 2.0, 2.0 = 12.0" and "10001".



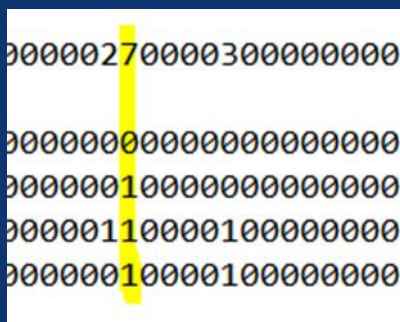
Parte 4

- Continuando:



Parte 4

- É bem simples de visualizar o que está acontecendo aqui, a primeira linha cheia de zeros representa apenas onde fica a posição de cada número, sendo que o zero é representado por um 'O'. Aqui por exemplo, podemos ver que já que cada saída (W,E,R e T) representa um algarismo do valor representado pelo display, na posição onde fica o 7 é formado o valor de 7 em binário. Agora que temos as tabelas verdades, podemos obter as expressões e fazer os circuitos.



ABCDEFGH --- WERT

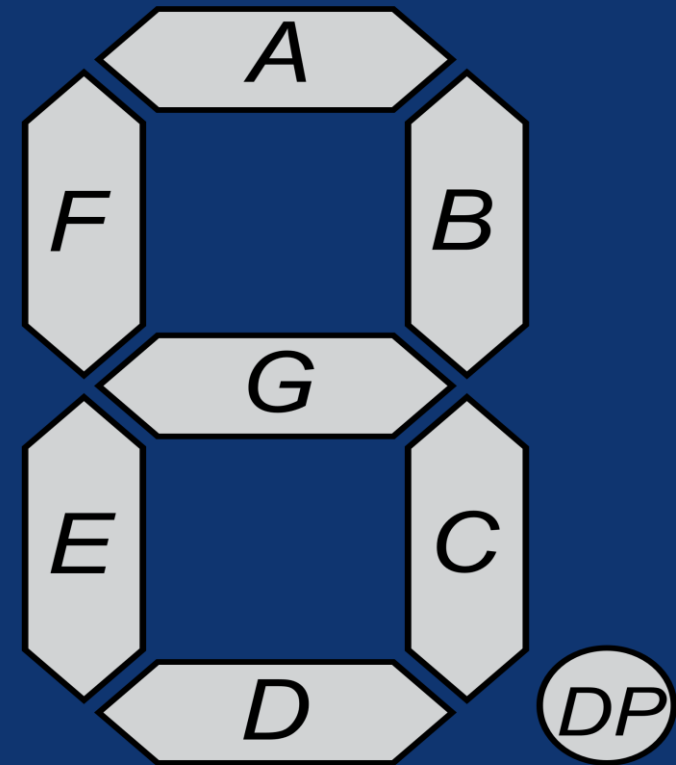
0:	1111110	126 ou N 127 na tabela e 128 na linha
1:	0011000	24 ou N 25 e 26 na linha
2:	0110111	55 ou N 56 e 57 na linha
3:	0111101	61 ou N 62 e 63 na linha
4:	1011001	89 ou N 90 e 91 na linha
5:	1101101	109 ou N 110 e 111 na linha
6:	1101111	111 ou N 112 e 113 na linha
7:	0111000	56 ou N 57 e 58 na linha
8:	1111111	127 ou N 128 e 129 na linha
9:	1111101	125 ou N 126 e 127 na linha

[illegible][illegible][illegible]

as colunas em `pe` representam os numeros formados por cada configuração dos displays

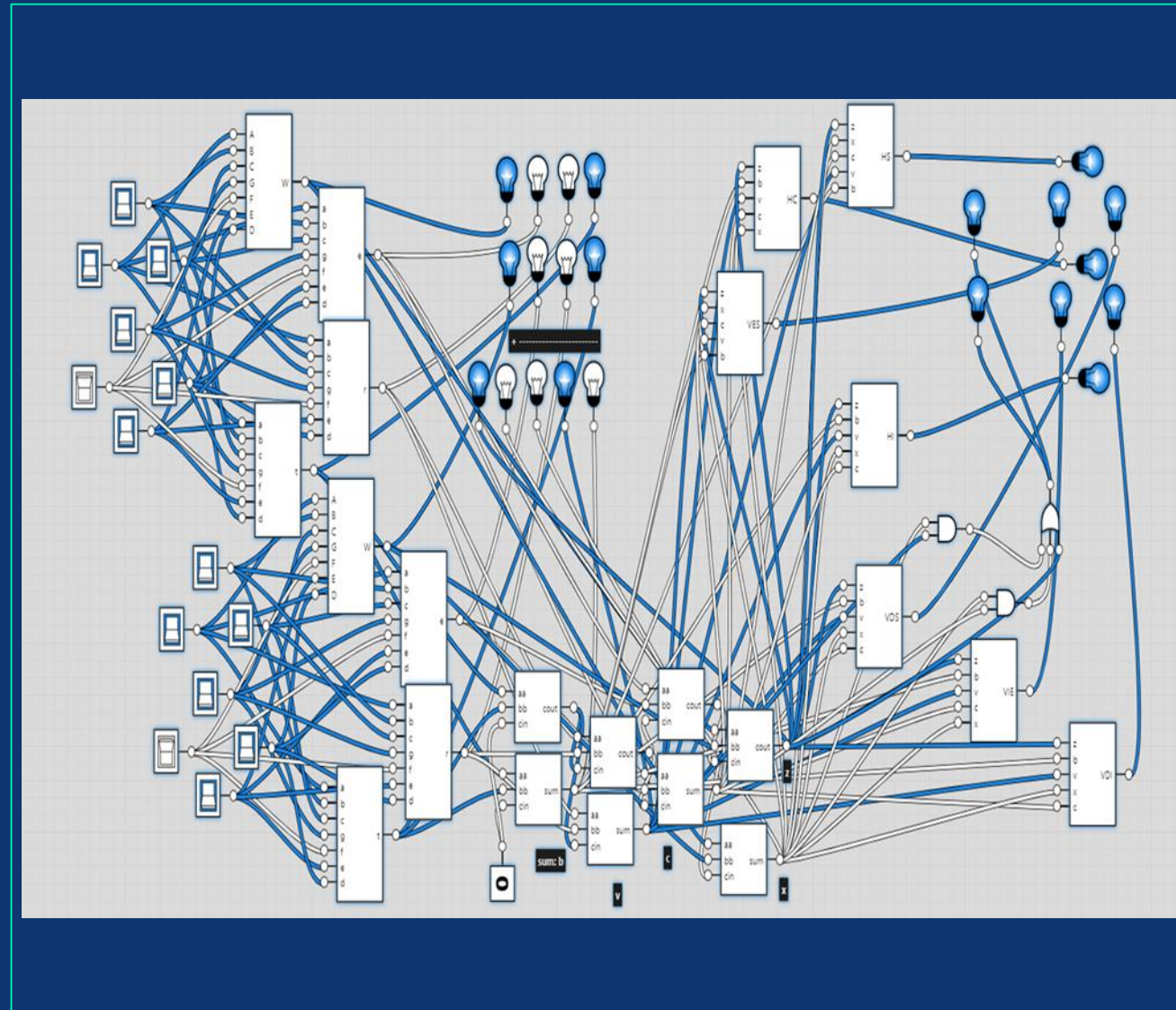
Parte 4

- Para o dígito W (milhar), a expressão lógica será $a \wedge b \wedge c \wedge d \wedge e \wedge g$
- Para o E: $(a \wedge b \wedge \neg c \wedge d \wedge e \wedge g) \vee (a \wedge \neg b \wedge c \wedge d \wedge \neg e \wedge \neg f \wedge g) \vee (\neg a \wedge b \wedge c \wedge d \wedge \neg e \wedge \neg f \wedge \neg g)$
- Para o R: $(a \wedge b \wedge \neg c \wedge d \wedge e \wedge f \wedge g) \vee (\neg a \wedge b \wedge c \wedge d \wedge e \wedge \neg f \wedge g) \vee (\neg a \wedge b \wedge c \wedge d \wedge \neg e \wedge \neg f \wedge \neg g) \vee (\neg a \wedge b \wedge c \wedge \neg d \wedge e \wedge f \wedge g)$
- Para o T: $(a \wedge b \wedge d \wedge e \wedge \neg f \wedge g) \vee (\neg a \wedge c \wedge d \wedge \neg e \wedge \neg f \wedge \neg g) \vee (b \wedge c \wedge d \wedge e \wedge \neg f \wedge g)$
- Assim já podemos montar os circuitos. Como não são tão complicados, eu não salvei as imagens, e já imediatamente montei o resultado final. Uni o decoder, o encoder, a soma e o overflow, juntos, de forma organizada temos o resultado final.
- Link:
<https://www.mediafire.com/file/wrz5lkyhlmtkkcq/compartilhar.txt/file>

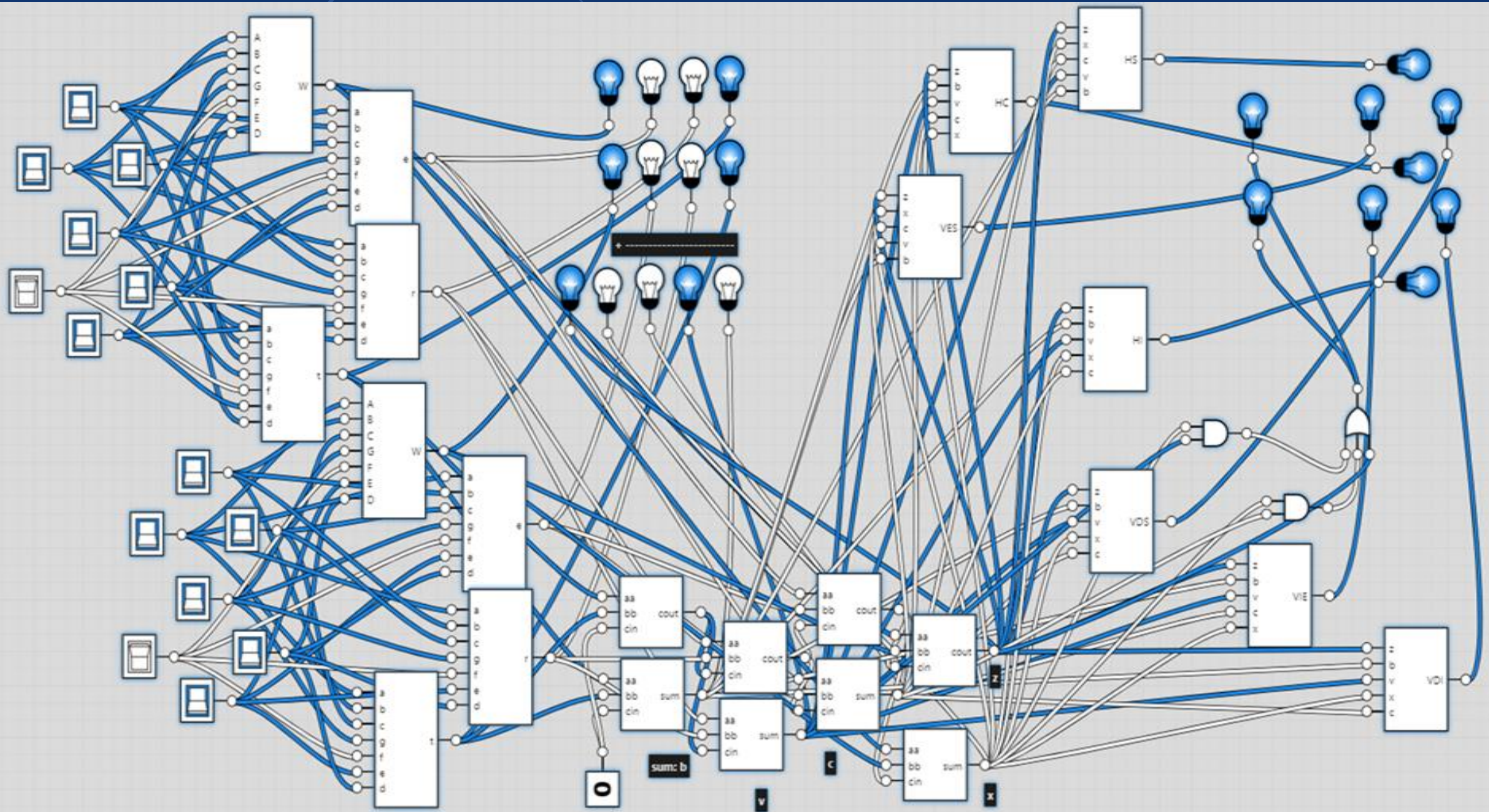


Parte 4

- Esse projeto pode ser dividido em 4 partes, na esquerda está o decoder, na direita o encoder junto ao overflow. Na parte inferior central vemos a soma, e na superior central vemos a demonstração, onde vemos $9 + 9 = 18$ corretamente em binário. Resultado:



Parte 4



Final

- Agora que finalizamos, já é possível passar o projeto para a linguagem C.

THE C PROGRAMMING LANGUAGE



Final

- Primeiro declarar as funções básicas, como vimos na página 2, a de soma, de carry, e iniciar a Main. Inicializar as variáveis, os 23 displays e 13 dígitos, assim como os 5 carries e 2 mecânicas ou gráficas. Num arquivo separado, criar as extras, como TRIOR (3 or)

```
//ESSAS FUNÇÕES DESENHAM OS DISPLAYS NA TELA, APENAS A PARTE GRÁFICA

if(clear <= 4){

    interfac(W,E,R,T,S,D,F,G,Z,X,C,V,B);
    digital(HS,HC,HI,VES,VEI,VDS,VDI,overflow);
    digital1(HS1,HC1,HI1,VES1,VEI1,VDS1,VDI1);
    digital2(HS2,HC2,HI2,VES2,VEI2,VDS2,VDI2);
    partes();
}
```



Final

- Para a parte gráfica, simples, caso um display seja verdadeiro, imprimir um espaço em branco onde ele ficaria. Os displays de entrada geram os valores a ser somados.

```
//ESSAS FUNÇÕES DESENHAM OS DISPLAYS NA TELA, APENAS A PARTE GRÁFICA
```

```
if(clear <= 4){
```

```
    interfac(W,E,R,T,S,D,F,G,Z,X,C,V,B);  
    digital(HS,HC,HI,VES,VEI,VDS,VDI,overflow);  
    digital1(HS1,HC1,HI1,VES1,VEI1,VDS1,VDI1);  
    digital2(HS2,HC2,HI2,VES2,VEI2,VDS2,VDI2);  
    partes();
```

```
}
```

```
//AQUI OS DIGITOS DE CADA ENTRADA SERÃO CALCULADOS DEPENDENDO DO DESENHO FEITO PELO USUÁRIO
```

```
W = HEXAND(VES1,HS1,VDS1,VDI1,HI1,HC1);
```

```
E = TRIOR(HEXAND(VES1,HS1,N(VDS1),VDI1,HI1,HC1),HEPTAND(VES1,N(HS1),VDS1,VDI1,N(HI1),N(VEI1),HC1),HEPTAND(N(VES1),HS1,VDS1,VDI1,N(HI1),N(VEI1),N(HC1)));
```

```
R = TETRAOR(HEPTAND(VES1,HS1,N(VDS1),VDI1,HI1,VEI1,HC1),HEPTAND(N(VES1),HS1,VDS1,VDI1,HI1,N(VEI1),HC1),  
HEPTAND(N(VES1),HS1,VDS1,VDI1,N(HI1),N(VEI1),N(HC1)),HEPTAND(N(VES1),HS1,VDS1,N(VDI1),HI1,VEI1,HC1));
```

```
T = TRIOR(HEXAND(VES1,HS1,VDI1,HI1,N(VEI1),HC1),HEXAND(N(VES1),VDS1,VDI1,N(HI1),N(VEI1),N(HC1)),HEXAND(HS1,VDS1,VDI1,HI1,N(VEI1),HC1));
```

```
S = HEXAND(VES2,HS2,VDS2,VDI2,HI2,HC2);
```

```
D = TRIOR(HEXAND(VES2,HS2,N(VDS2),VDI2,HI2,HC2),HEPTAND(VES2,N(HS2),VDS2,VDI2,N(HI2),N(VEI2),HC2),HEPTAND(N(VES2),HS2,VDS2,VDI2,N(HI2),N(VEI2),N(HC2)));
```

```
F = TETRAOR(HEPTAND(VES2,HS2,N(VDS2),VDI2,HI2,VEI2,HC2),HEPTAND(N(VES2),HS2,VDS2,VDI2,HI2,N(VEI2),HC2),  
HEPTAND(N(VES2),HS2,VDS2,VDI2,N(HI2),N(VEI2),N(HC2)),HEPTAND(N(VES2),HS2,VDS2,N(VDI2),HI2,VEI2,HC2));
```

```
G = TRIOR(HEXAND(VES2,HS2,VDI2,HI2,N(VEI2),HC2),HEXAND(N(VES2),VDS2,VDI2,N(HI2),N(VEI2),N(HC2)),HEXAND(HS2,VDS2,VDI2,HI2,N(VEI2),HC2));
```



Final

- Após a soma, o valor obtido gera os displays da saída.

//AGORA QUE SE TEM OS VALORES DAS DUAS ENTRADAS, A OPERAÇÃO SERÁ REALIZADA

```
B = SUM(T,G,RIU);
RID = carry_out(T,G,RIU);
V = SUM(R,F,RID);
RIC = carry_out(R,F,RID);
C = SUM(E,D,RIC);
RIM = carry_out(E,D,RIC);
X = SUM(W,S,RIM);
RIDM = carry_out(W,S,RIM);
Z = RIDM;
```

//AQUI OS VALORES DO DISPLAY DO RESULTADO SÃO CONVERTIDO PARA VERIFICAR SE ACENDERÃO OU NÃO

```
VES = OCTOR(TRIAND(Z,X,N(V)),TRIAND(X,N(C),N(V)),TRIAND(X,V,N(B)),TRIAND(N(X),N(V),N(B)),TETRAND(Z,N(X),N(C),V),TETRAND(N(Z),X,C,V),TETRAND(N(Z),N(X),C,N(V)),TETRAND(N(Z),N(X),C,N(B))));
```

```
VEI = PENTAOR(TRIAND(Z,N(X),N(B)),TRIAND(Z,V,N(B)),TRIAND(N(Z),N(C),N(B)),TRIAND(N(X),V,N(B)),TETRAND(X,C,N(V),N(B))));
```

```
VDS = ENEAOR(TRIAND(Z,N(X),V),TRIAND(Z,N(X),B),TRIAND(Z,V,B),TRIAND(X,C,N(V)),TRIAND(X,C,N(B)),TRIAND(X,N(V),N(B)),TRIAND(N(X),V,B),TRIAND(C,N(V),N(B)),AND(N(Z),N(C))));
```

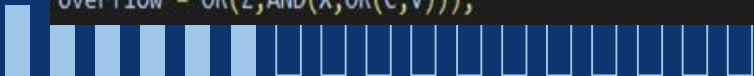
```
VDI = HEXAOR(AND(Z,X),AND(X,N(C)),AND(Z,N(C)),AND(N(X),N(V)),TRIAND(N(Z),C,V),B);
```

```
HC = HEPTAOR(TRIAND(Z,N(X),V),TRIAND(Z,N(C),N(B)),TRIAND(N(Z),X,C),TRIAND(N(Z),C,N(V)),TRIAND(N(Z),C,N(B)),AND(X,N(V)),TRIAND(N(X),N(C),V));
```

```
HS = OCTOR(TRIAND(Z,N(X),N(C)),TRIAND(Z,C,N(B)),TRIAND(Z,N(C),V),TRIAND(N(Z),C,B),TRIAND(N(Z),N(C),N(B)),TRIAND(X,C,N(V)),TRIAND(X,N(V),B),AND(N(X),V));
```

```
HI = DECAOR(TRIAND(Z,N(X),V),TRIAND(Z,N(X),N(B)),TRIAND(Z,V,N(B)),TRIAND(N(Z),N(C),N(B)),TRIAND(X,C,N(V)),  
TRIAND(X,N(V),B),TRIAND(N(X),N(C),V),TRIAND(N(X),V,N(B)),TETRAND(N(Z),X,C,B),TETRAND(N(Z),C,N(V),B));
```

```
overflow = OR(Z,AND(X,OR(C,V)));
```



Final

- O programa funciona de forma que ao pressionar a tecla que representa cada display, a situação atual desse interruptor será alterada, fazendo ele acender ou apagar, usando a função kbhit que verifica se alguma tecla foi pressionada, e depois é só verificar qual foi. E é isso, muito obrigado a todos que acompanharam! :)
- Link para testar o programa: <https://www.mediafire.com/file/hscas9r6wc258bs/Projeto1.exe/file>

```
00111
```

```
+01000
```

```
-----
```

```
01111
```

As teclas mostradas abaixo, como A, Z, ou W, representam os segmentos de cada display, pressione!

