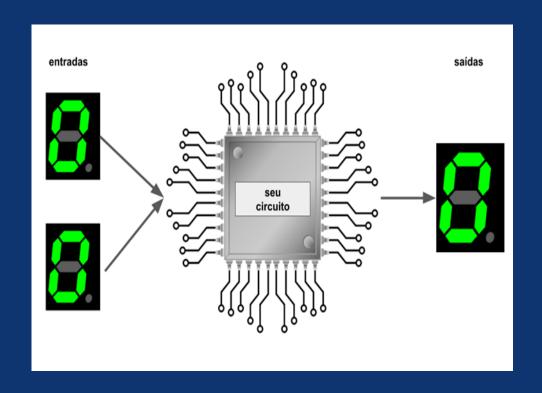


Objetivos

- 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;

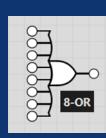
```
int AND(int A, int B){
    if(A == LIG && B == LIG) return LIG;
    else return DES;
}
int OR(int A, int B){
    if(A == LIG || B == LIG) return LIG;
    else return DES;
}
int NOT(int A){
    if(A == LIG) return DES;
    else return LIG;
}
```





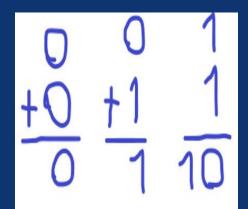


```
int OCTOR(int A, int B, int C, int D, int E, int F, int G, int H){
    if(A == 1 || B == 1 || D == 1 || E == 1 || C == 1 || F == 1 || G == 1 || H == 1) return 1;
    else return 0;
}
```

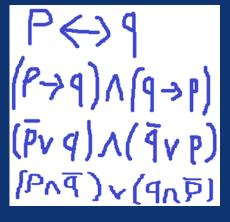




- Podemos observar que existe um padrão para a soma de 2 bits. Como as entradas devem ter 4, representaremos o resultado com 5, então podemos montar uma tabela para descobrir qual o padrão:
- SIM! Um padrão existe, podemos observar que quando as entradas são diferentes, a casa decimal em questão será 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.



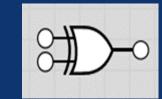
X	Y	Uhid.
0	Ō	۵
0	1	1
1	0	1
1	1	0

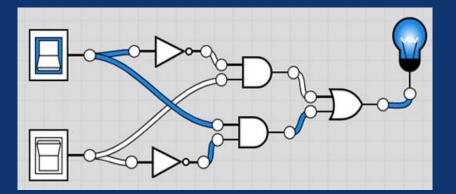


р	q	P_q
V	٧	F
V	F	V
F	V	V
F	F	F

 Essa expressão comum motivou a criação, essa é a porta XOR, representada pelo símbolo ⊕

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



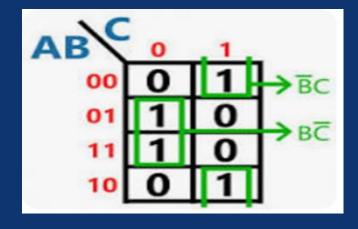


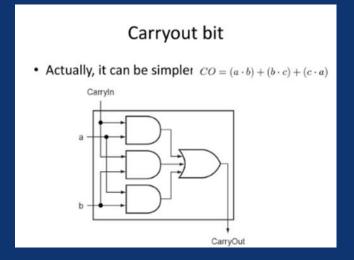


- Agora só precisamos resolver o problema da casa decimal, que vimos anteriormente e ignoramos, como podemos representar isso? Bom, devemos prestar atenção que na verdade, o valor que sobra da casa anterior faz parte da soma dos dígitos da casa pra onde ela foi enviada, já que na verdade 1+1 é o mesmo 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!



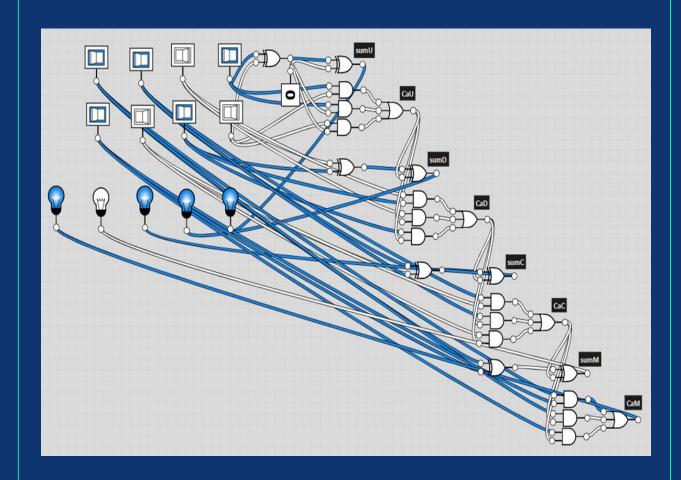






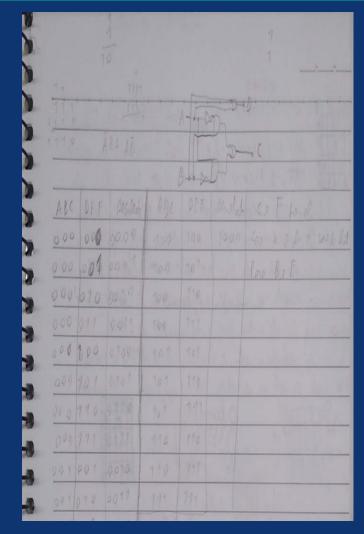


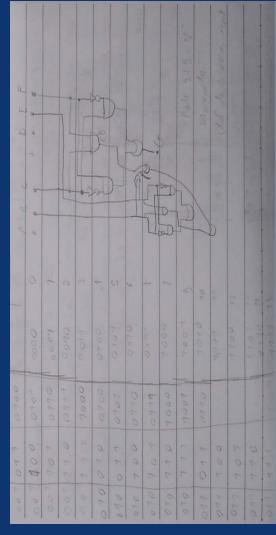
- Lembra que todas as somas precisam de 3 entradas? Bom, nós já podemos começar a organizar a cadeia de somas e carrysque montamos. Lembra que anotamos sobre as 3 entradas? Então no caso o resultado da disjunção exclusiva entre as duas entradas, deve ser somada ao bit que pode ter vindo da casa anterior, 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 ignorassemos 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 simplificada por uma caixinha com as entradas e saídas, então próxima vez que a soma aparecer, não será feia assim!



Finalizando parte 1

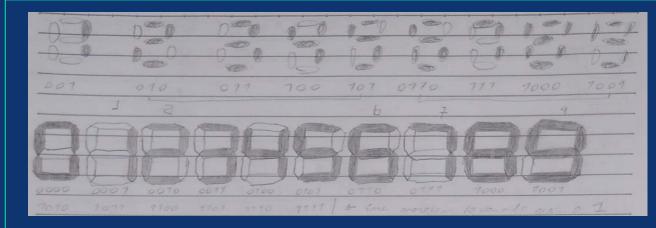
• Bom.. Isso foi.. Menos de 20% do projeto. Vamos lá para a parte mais difícil, e vamos representar o codificador, que transformará a resposta de 5 bits em valores decimais. Eu não sei nem por onde começar.. Mas vamos tentar. Vamos colocar as informações que temos num papel, e ver se conseguimos pensar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



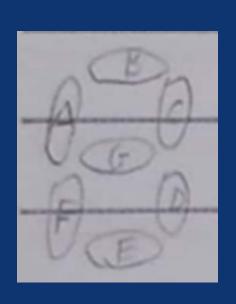




- Como fazemos pra descobrir o circuito que transforma um número em binário de 5 algarismos em um display de 7 segmentos que represente o número corretamente?
- Podemos perceber através das tabelas verdade que de alguma forma podemos ser capazes de organizar os circuitos..



UNE P	A	derive totale
6000	18.	- NUNWINEAUR
0007	0	
0070	9	
0077	0	
0700	1 -	- PNONWANE ANR
0797	7	-D NO AWAVE AR
0 710	1	PNUNWAENNR
0 777	0	- NUNWAEAR
1900	7	- UNNNANDINA
100 1	7	- CLANWANEAR





- Usaremos as nomeclaturas dos displays desta forma: vertical/horizontalesquerda/direita/nenhuminferior/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?

Α	В	С	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 A.B.C

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

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 $\overline{A}.\overline{B}.C$

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:

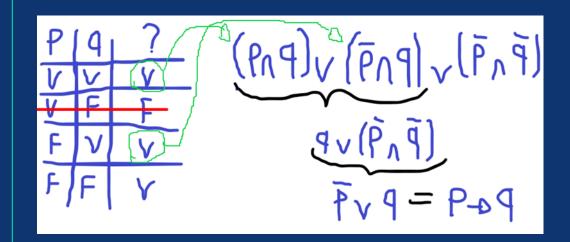
$$\overline{A}.B.\overline{C} + A.\overline{B}.\overline{C} + \overline{A}.\overline{B}.C + A.B.C = Q$$

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



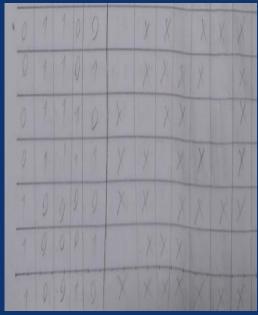
- Se Q é falso ignoramos, mas quando Q é verdadeiro, devemos guardar de lado A.B.C, sendo que os que forem 0 são barrados. No final, somar todos juntos.
- Exemplo, do lado: na primeira linha, '?' é
 verdade, P e Q são verdade, então temos
 P.Q, pulamos a 2º, na terceira teremos !P.Q e
 por ai vai. No 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.

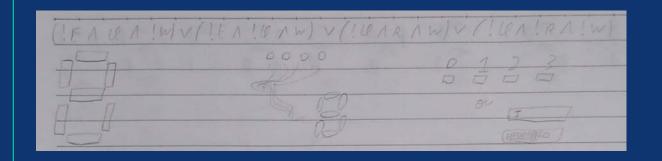




Então esse é o método que usaremos para montar as expressões, a partir das tabelas verdade. Outra coisa que estamos adotando: como as entradas do usuário vão de 0 até 9, a maior soma possível é 18, então trabalharemos apenas com a casa das unidades, mas de 00 até 18, ou seja, os números de 00000 até 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.





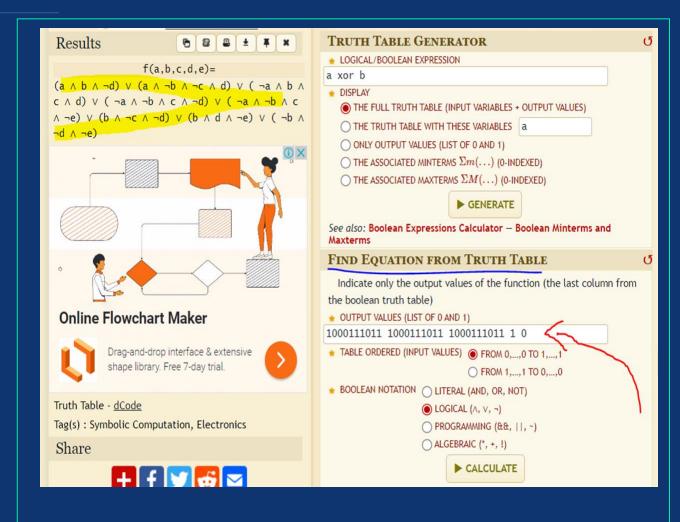


- Bom, agora vamos montar uma tabela mais bonita. Perceba que não é necessário seguir preenchendo após o 9, pois os números de 10 a 19 e de 20 a 29 tem sempre as mesmas unidades que os de 0 a 9. Então a sequência repete 3 vezes e falta apenas 30 e 31 no final. Agora nós pegamos os valores das linhas 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	Х	С	V	b			a	b	С	d	е	f	g
	0	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	0	1	1	1	7	0			1			
	0	1	0	0	0	8	1			1			
	0	1	0	0	1	9	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						

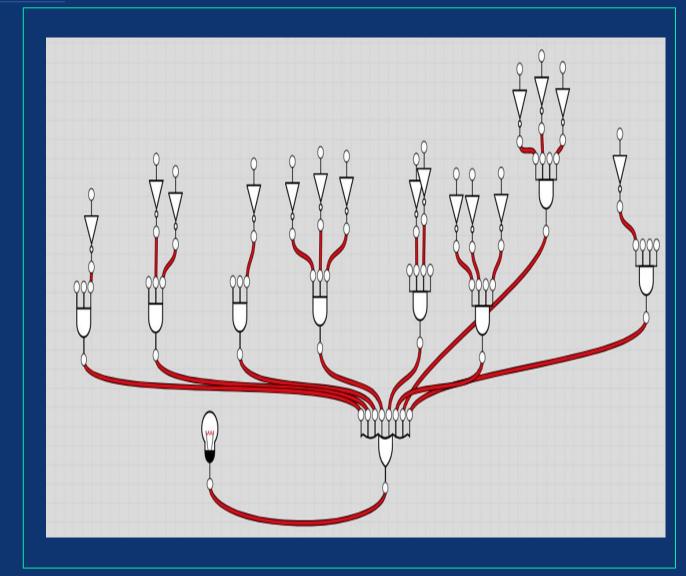


- Lembra do método que descobrimos para derivar uma tabela verdade de volta a expressão booleana? Bom, agora nós temos o resultado das tabelas verdade de cada display. Já que o professor liberou em sala, não precisaremos quebrar cabeça fazendo a álgebra e as simplificações, pegamos o valor de 0 a 9 do 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

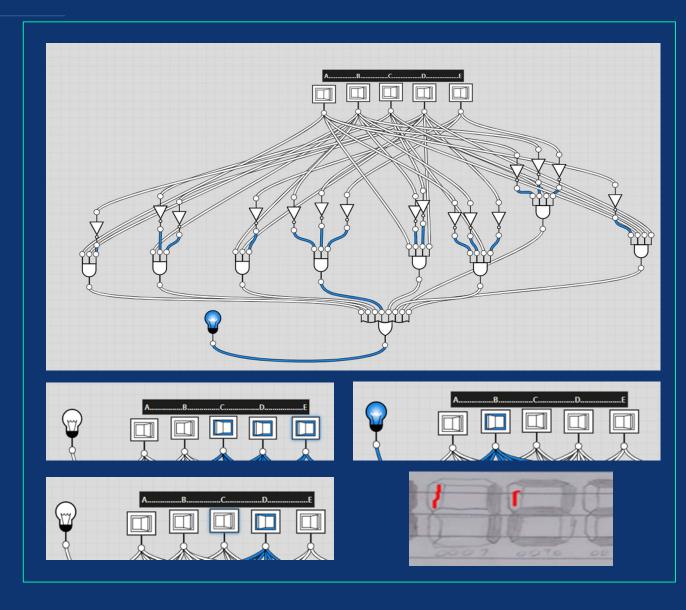


Finalizando parte 2

- 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 ∧ b ∧ ¬d) ∨ (a ∧ ¬b ∧ ¬c ∧ d) ∨ (¬a ∧ b ∧ c ∧ d) ∨ (¬a ∧ ¬b ∧ c ∧ ¬e) ∨ (b ∧ ¬c ∧ ¬d) ∨ (¬a ∧ ¬b ∧ c ∧ ¬e) ∨ (b ∧ ¬c ∧ ¬d) ∨ (b ∧ d ∧ ¬e) ∨ (¬b ∧ ¬d ∧ ¬e) vertical esquerda superior (A).

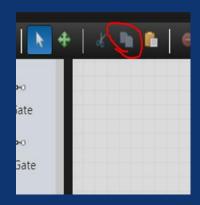


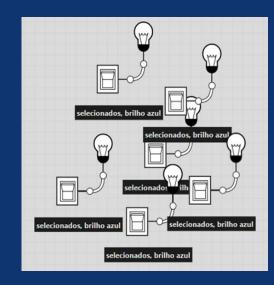
 Após organizar corretamente em ordem temos aqui o circuito funcionando. Mas como sabemos que está correto? É bom testar um por um, pois as vezes os erros são discretos. Por exemplo, o 2 e o 7 não tem, mas o 8 tem os display VES aceso.



- 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.

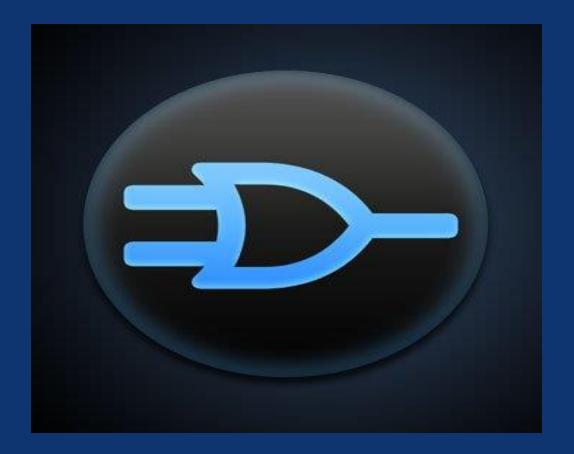






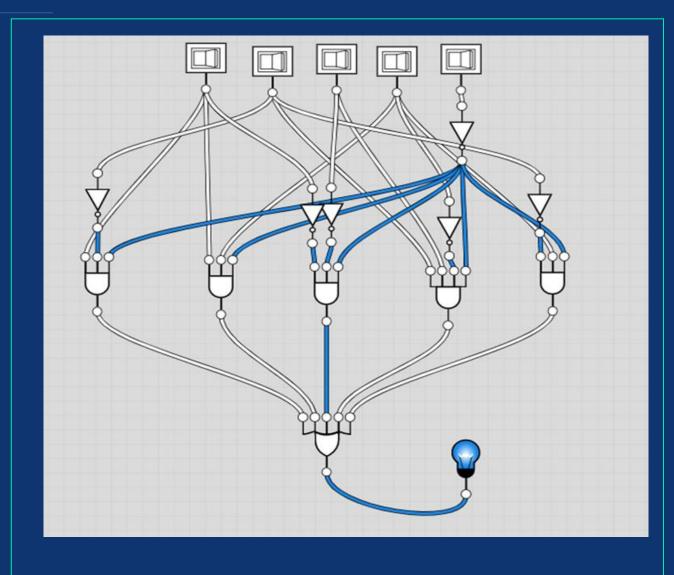
http://ns.logic.ly/1.6;base64,jdLNbtwgEAfwV7E4Bz wMw1cVRz3kkgfoOQKMs66ovYqxstunr+tNcnAuu Tlz4sd/uC/zy5jKtbn8KdPSsVOt5x9tOy1iL4hybaU w7OF+jr9zqk29nnPHyvhyqs9xLfHnRxtr1rHvmHE QIGXgKQ09J5cjjwqJJ0Dpkld90Jo1l44hCq/BgkRpt DVGE2uuHZPaCFDeW6PAASmNrHmda6jjPHUM WHuQLG9jTaejgmKOmQxw7aXkJO3AvTaKy0gD upSHIHFXbEJB1jkrwYOSUiuzM1CRIAQ0IMkq8O bAaOa1nte6BTaEsuQvrBJiLkeVDSAjaMezU4pTs obHSJprTcE7RNQ03FQeBDnlAD2g27K5oRRYY QGV81J5jWQOppovdUskl5y2o9DPy10TX8dympv wdy3/kWmepk25ld9f8Ovp8btx3Qaepj5f9uvG6WP+e0vf+z/HN0z7/vce/qE=

data:http://ns.logic.ly/1.6;base64,vVrRbuM4EvyVwO/UNNnNZnOxs7iHe9kPuOcDKYk7OeSSwY4HN/P 3V7I0wGlzFmhYSYLklbblcnd1VTWVX59e/ngcn74/fPvv0/OXj6dP5/PnXz58eP4yXB4Ynr5/8IOefvv1pf5 nHs8P5++f54+nL/97PI+f/vHjKaeHr4/Tx1NgphySOYt5dpJLdjXGyRm1VK1ZSb6eHr59PLHy4MUi4ekay PLp4TteTkOkZMlasqeU/enhz5dzOT++PH88ZTo9vHw9f/56BshWnr7Mpw9doCZfdMw8uuJTcxJ8cnVu3 tVMlGoutZpeQAnbgPfGu6uKMvEFIE9DAklyUfHJoh4CKmqZZxmLCzGyEx9nZ4nN5clPU+DSfA4XUNH TIJo5Ja/EOUZZUenAHGM0b1E8m8RDYBFZm3QaXa6IApZGZ1aaa+b9bDxbqX6FZTyYJZQks2R8iDe slU5FpjazizFkJzWoK4WTi36q2mjKyY8XUGjbEDh4VIU4pHwlpvI8/R1Qy1RHy+LyqABkpbhiai6EiWppIU KIF0AuDJwzhWgYiCA5UbpgEkJTvc8+KEcxL3/H9Pi8QuJXcJ5fzq8l3kYLuamzaVInrQJOqN7xWJLV5s eobYWjQ1TC+wFPyiQxrGOX/BCzcGBTBcPCNTS+Bw1rDZny6KZ5KU4cxVWailPO0goVs3HtFseByO M7RA8i6VgaYDKQJAKpNOXM/Vh+0ggiLc3TmF2BDDkh7x3GrrpaEk3THFEvumDxKQ+UJRigBCO1lc9 o3iALbyz7GCjqfY2iGZ+rTMHNqRilbNVIMXZcdAqUJpmDXeAEQhFCAq4jrx5jvzVKBw+9lOqFW+alZHd Uh70m0NWchyw7kTa5HAo7xeyT5yal6tapMIRIMaFfxll5K04YUBRcJAYoUKBXk3VTccqkFpZqFAg/itMy pCeQqy2qJT5XuMVaHEuDZzEUKBuHHH4Uh5dhi5i3jFbFq63qYjFNQUopGKlaUJsSmjPgcKGk5iPcJP qy1kZsgH95L3xpTdIVDQwugdnJUB2LMd/Qqp/AgQZXg1+4UKs4vJV35ith2PFFNME10+phxPBQ9oAk nFCdbao0DfBV1AzTz3RLcX5CHKqJ65QmvIzHWDWJrvA0uwkXVq0qjudVcHxM4AjYGqIm8pnipn8yZE sSiJiCvjavm5hTZS6QutHJFBLkuBWkDiGHnzEY+ylOZYMjinn2i+BAk2P2G5MJPYS9KiYcchxe5Y6bqlO TzDFAdGCWeRkrdtbGhLEyqSMomutK5Jgi7CqnyB61kR9gFnqrZPQQyUghydfQSE9x0I0wEzLG2BLM oRjMYUkb48xxzjSjBqtXYWIGMVu0DlpkW/pZvCHgF9IYHuPrneqicUvoBVwGyYIggFybg/4Ex+RngnVitK etNH7wHg2MSRi6s4Ye9GxIAjIZJ5Qs8X3uYKlkxvSAtwLe8Bwhx3NwEbRuwZth1i5oEuVBA4hjPgSOID cSB+hQkLTkMMC0u/o0za14VNhRQEGkxYoQDf2ZK/JOwrTFukZnFRpQwmWSF9OkH7YJa8iSqVohqk p3yh+XQhgiJzOChPgWkZkrEgZVAbUbGrWhAYcJcovEZZFl8a/vq0RHuDcCtpltLbwLjarOMWOEUmbU BtHY5QijqvDLWutlcYsUiRNisWV4Qo5lfT9qE4ek2SBEGCiVq43qYk20mQsjY40oEPKNoigzqJu8iFWC XW0mjsA5XALyggRE3sAkLDnMumRBLEvxPtbM2FaMY0PYAiIJCr8cZ3leStao1Ta1jcJpEeK4GIAGLB O6ZeMFIrRHTVJEm+6LfvPcKmhZweGEpQb5EsqXBNKTaUoVQ59WrYHRY6CS+iUDo6fbShOGjKlkW AUCUL7TMIPBGIUtO4LtORk9UiCE3wUqPGvBRujzCqYdQfaFO0FpPPO2YCVYIGQEa2QPqYvfQOGf sGY0rAAxwsAR2yDDCKHmJ4SKqpC3wrG0tTSL6g+mALTIMMcNDitokyRrBPvRxavK1yc2AdJLaA8vjB GqSH6MDDhKqQw3RTRdNwYE8EX5sF6iFnoR3IXDqvZxQj+xroM910n8ujYvf74Kfqw191KdjrocGIREQ JTcpbHyCM+0GtcYinSFGmB/x34FldwODZCEB7QyC0i97DZXsdgrLE+Pf3w6/7t+faqvhrxga2htxNbil7yF SS+QfshOZEWrMF1rwgFL0A4gw36HFKhrt7CSY7e5LFMoHaLhXzHRgmN8eX4GEPxh2z//9fs/eze59Q W/P0/zt+Vq6we8vL5vMb08f3u5vw6mb5E7EsxOZfoy3nUwfZG1E0zfHncdTN9a2qmmb2+6DqZvC/wrm LBTma616UgwOwTuW5qOBLPTpr4cfh1M31bR2aa+HH4kmJ029R0aHznaO5XpOyw+EsxOZfpOY3fa1 CWanQS+tzK328FOm/qOz6+D6TPaXju4s019RvtObbrdm/bs4M7K3A5mZ5r6biPsjHaX679Xm7pc/50l3 Of6nWDubVOf6/e69p3edLtR8ttxpi+CdFbmXqXuiyDvRODb2/SGnOk7CXwnnek7CHynNvWdA74TgftOj 3u9qeuTHQlmr01dPT8SzF4q75qGl8HsKPDdo9117tY72l0Xuw6m70jynbzpdjBvuB3cDuYNl7jbwexwpu 9Q/DqYvpsfnWD6bhccCWaHM313Uo4EszNN9xL4djBvuDf1ce6dwlVfm98pXPUNQCeYPp3Y2Si7bsP0 hqsuAh4JZu8cuCsCHAlmR2f6Vo0jwezoTN8J95FgZOdGRtdhz5Fg4s4Jeddp5ZFgdCdCdB0qHwkm7b Sp62I73tR1a/WVznzY/k39t/8D

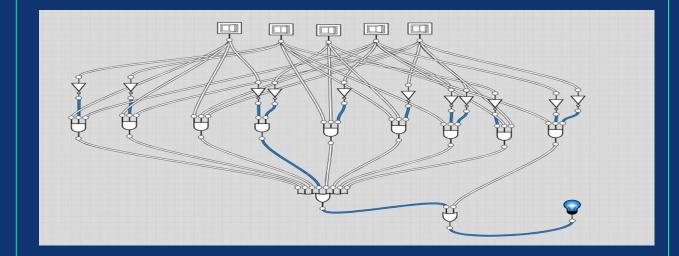


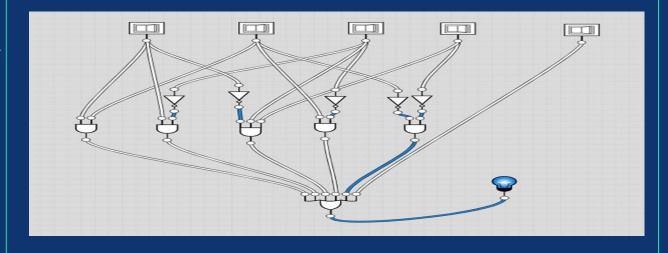
Link

- Agora, 'F' ou display Vertical esquerdo Inferior (VEI): 1011011011 1011011011 1011011011 10 (a Λ ¬b Λ ¬e) V (a Λ d Λ ¬e) V (¬a Λ ¬c Λ ¬e) V (b Λ c Λ ¬d Λ ¬e) V (¬b Λ d Λ ¬e)
- http://ns.logic.ly/1.6:base64.vZnNbuNGEIRfReB96PnpmekOVoscctkHyDmY37UCRTLWMmK/fdg iHFPLJTGCZOsoi+Ln6uqqlfRlu/++SduX1fM/293jurs/HB5+u7vbPfbHP/TblzvVu+7rl338u6TD6vDy UNbd47+bQ7r//e0j3eppk9dd1io7g0VlgipAxyqwOicKYQo5QIzJdqvndaeN7uX4ZbrVy7ozqqfxy3er H/tDOGz2u3VHslvtnw4PTwemrGH7WLq7NiqJqlxyovoiBbhsRMyShFcBlHNOa++PVNa68/sPUKb XHt7f1DeBSinKEmUQLIIUgK9SIWIBaGwIPklf9REKvO3NFMr1MIZSN4EqGkqJEARYaQQYcIKH BgJSSapmvk+NRyhD2NtPUiqWFGSVVkB0bCqyWWBySmAG1jD6mHBQikf8q/FdqVTY5Z+JMFL FXIPQ2hLbHEmg1FnkVMBaEzQldSRCOhsdnoi06XHk/QnRZjcAmRYYWXw0NiPb2WWemWOYI FE4aXVCZwFsOsI4R70cCYH/w7DF3mHgKpqMYAOVKrSRmqVRSIAGErZkKfkuIUk5JAC43o1o3 qRBttA7zcRBF8GUZJOMmmEiFZaG2DmQvZBeoQsR0WAenIPqTJo3O/sex9JcBaOqC8VSErxq PKfKHOSqEzHlmL01kGmwsQXT++mcQPf0zvlamL+GqQnMbn/4GaZ6GYoqRaRseUwyaxGiJ0H aGcCiHFEZHAznew5HGC39WXyrSVC/0agWGgaxiZ0hKhWOHe3ZNC4hz6llyR4yFdwgDdfGNHa OVhrPyV5FY1UgrFxiKjAIRN5zTJE7rUbKqKp1YagLB+yPgUDBOcHsdJoIAuYUDHuD84W7QXH2 cY1yDFcNXLGeBXrzre71QKB94xq3TcRK52UmYSiyBpiKiJa3OqNOxJVAINXJrL8qAqXM+eZcp4f 2KZqKScSgWI/ALU48HAGQqjHEe0WDP4zFXk39oTn4xjSunWb/Y1JHwCElOfKdfK0jJCmC4tNO DWAxOlOsPsGQO8/bYXcszC6Lndx+u/l+f/grPm3j5PwARVMpICSfWvhQE9kj2UghoySlk6tQToca hT2NdnXwi+XFhrkOkg8cab/bMQi/cWrIP7/90RocwwXfdrk8v37b8A8er2+rzuPnT5ereZi2q908TJvL xzALyrSZdB6mrazGMHoepi1BbgmzMKa2QJ2HaTvhNCrT1jULnmmq8VYDNz0v3XKbFpRpO2fP w7RFQ+M2ta3mLZVZqLl2TG2H4sYxtT043hJmwcDXjulymA8cU1uCN8K0PUrPw7Ql+Cdt0+V1sA Bzbc5cnsDm48Z0OcwHrnZbt32SZy5X5gM903YEaYS5dkyXn2c+MIEvh1IQpq1152Hanq3GMLB QB03dckuYpZxpStBbwiydgZtW85YwCwZuM+AtYZZypunL5mHansonMHen32C+/qc=



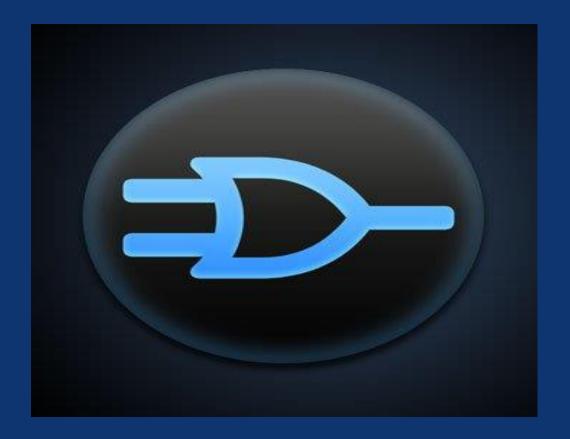






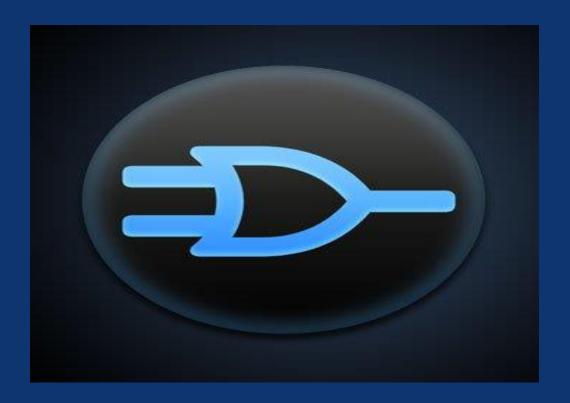


data:http://ns.logic.ly/1.6;base64,7V3LrhvHdv0V4YxPtev9CK6CDBwgnmQSJNOgntcKZMmwjhM7X59V7D4PHpJ1d6ubzB1QhmXBksjF/Vpr79rV/Mvnr3/9ID//+eGPXz5/+fbx4eenp1/ 4YcfvnybDr8xff7zBzHZh3/8S/7929PXXz58ib/Ujwl/8m8PH57+/BW/4soZkaJllbrMtFcNv2qceVmD9FHrovrfpr7L1/RfNT8tr/3tfz495Z//6fmPPHz4/VP5+JC0qUIZz5y2nGmdPlsxcNaszSaL mGyMDx/++PiquX/48OfHB+EePvz29Sk+ffr65eND4A8f6h+/fv3t6V8PH+V/Hz58/f3p19+fAKvFz9/gww8kGMHFlmzQzNUUmTbGsVhlZNa11rjnLil5qGGkmuQBiJxEePvDDIH98Z3AvNF WG1lZTElynV1h0UnBpFRF20KNjfkAzBoLm3dgTPgjaN7bEbL8nci089E5p1kNyjEtbWFBa4ROjiU61USq/lDMefWCTK5B9t/ficxEXqR3gRVZE9MiSQRw0yzlzJuWpTmXDsiCeEVm1i BLfxNZ/FJODKaM0y16hoCHK0U1QOQlay0GxHw2AH2AJWyYpDng0pxP3PCXH/I9qk9fZhSKgsBVIXPMmoUa4TLfEvPVaqZUxrurmk3Us2G8nfyCQOrJO7MTAqtCsdFnJJYOTFvnm FehMZPxn1C417UdEHjVP/iMQPMdEdQSjC1VMlNQ2HT2iAsJp0QlRMC75GxnL1j7agNtJrsfgob3sUohG3xhWinU1wgELmpvTOWu5LLUGvtqA7+jDbiVKUWUt1JrYzpojkiUSF1rkmx W1FjsUnTFSyT2wrcbgsJD4MgG1rwAgoY8DUUX1gy3yajMTZjLvnT6NRLFjl4wrriÚuWCiFYNcsl2hwqZuEh1sbMHlOReEcM9VQjk56ddkvlxAniD48vXpPYKmLT4zctCGALJrYNwgfEJ YNOVVrllouyDQM/dJ7Sb79v0vAhAUAMIK1SsSi3NBOvwqcsarRzUoOeewmCDoSb2tjrM5ZLC0oCSh4SDc4KpiXPbilHRk4N3CVDatJc4FgmQpj2Yyb5l3AeMr/RaM2YTG2OqUspZB D1SEh5MID22YsQ0JEitvC+9Kfw6NAtPtZ5qki7O8KRaNgaNKcCwoC00AxkDVCiWpsOQrn9wZMOI4dU+4bRUaz6OpNeHtue/VlwcUT4q38H5IVXInvZrrF7fnwkYpdZzHehMc8Akka7 WQbuB47bWEXYxqtiQZVlxGRTMX9HDM8Ytx5HFIVZyAOARpqqaJBF0LUQSfhRlKk1lk5Z0zXpoXdlGnYNSuQdySqsqBbRN3CnGD8oY6h1lrcu3yNvE0q1qr5Vk0dkfT5Oy9Ri4zlTzy 20SJ3sJYJopCC2CNa2UOG2/49U3jXJTgGMiAhBqjq1fMC9eLL8LT55JVylsUEmfz2x3F8ArTfP3tpPmQTgSPeJG6WNThCld1y8hqrCvVay/DEjRhEsdq8M+u/Pkk3rDCRSz+BMvnT3/ 9+ek/0++f00n9Aztp4QNLzfe+jCPdhÚNWcTQi1ddibZ1LjnRH7z5jcihERzXvnYBGywks+euXLwCDP7Dl6X//6UcqEcx/4acypf5xePXDhzz8fŹr0Pvz55a+Lv2BoHLknGH4ZDl2ULoOhqS 4iGBopXQZDE6FEN9E46TIYmiZ/C0ZeBkMruwMwpBaFClbGSHuCGbiJxtV7ghkEMI0CLoOhddNEy9DY8TIYWmNNtAytM7kMhtZbES1DmxDuWYEH2USbEw6KHonbiG6izQYvg6F xG5kOtrlpPTcN3LTVMuvBDAKYNo4cZBOJ9W/kpvWsfUU30SQl0TK0efaelrmim2h66EZ1Zr24uml2rQdzRTfRlOKN3ERTitRWZaObaEqRCGYra68XV1cseuvBDLJpa9GjydYbuWm9 B+4aatlaN3FjbJpvWWu6CZad3Ej1l7fqozAkFqNy2Bog8C3YNyg1yZp2D3B2MFMjySO9gRjLoOhSYA9wegBGNJIY08w6jIYGrfsCWY0uSIVrT3BjGQnqU7sCWZUgUkvNih6pOH+CZj PX/Nhmv+hHxD8108ODocF34qv+P4VPtf28qlUlf1IIZ5Hirx6pLDGI4l08aF+WNbh8Kt5re7det1//PPLfh3swb0tkQXkE9Ouf4KEd0DDKq0TohTL992vq9rqCBfPvAkQ9SHCR8YU5nlzyTe YQcyn6sqqSaD+cfxxK7lfTtX5ZLjzWlkZBHdBXGP5rohoc1C5xwzIQQrHUm2CpcDh0xBT8ss6Sj8DBFArrdVW9ZOuw0bg5AAYxQkx6ODga+zhlWyQ6TkyaQw6NYHk8055UCqcJlVsli wLgoJPEDjKOWG5CsboGaSdlDLGelEAFcrrk+PLPXbyOLr9YgsEVYo9ZK1h3seDBEAaquqRFDNKrybvHQwWlA74TOst+b3beQh/XVpVzBiJHEhQATEiC40oyTZeghPLRqPVk1RSw GZcSRfWQ/yuNb0WOEQbqkzIFvigDFj01qOAFJ5QgLhcjvDZ4cycS+ORW1IHx93Lzp4GeQtpldFeXFxvON1TOrfe0KAgQ7PM17rGypflEkwlaPzqYls7Lwxx+xkLMf7AY8LHDVwzmDX z6+1kspbi3DctomiLNpvDtleajeOATsnjnoG4QDhy6P3eXaeMhPnAv9llxBmdjaN9Hri2nGEnHUhrDg1P+OoiNahlj7ZQkVjmgsBOkClTtHxUqo5FNPDwpALEw9aekCRnttInwzOm/TMW8 Jlbuw2R/GKz4VCzqqDfAODQW9rdH0q2iKhKXVd1jslhxGkAw7P+wq+mMVRdhloLhzFRfmq7Mk+8SrrKGEdwtUz8DBKqW6qOBkVs6qMXKimwamLp+QkDTcO/qlrWbUYR04wCl7E SJQryS/uDJGMA3nvZbdGBlfAOC2gMEk0AU3CJSlkEM+yTuUmobSHgcCRMshn46iebAb5FuAqs21lCLyuY4xlogQJp6NsYHYnmYyuCQMmMmLePgSQCVQohFYH17hl8Q1c6RDZ zsM6aBnCCledW6hCVfEgF4YeSaMfQSB7kSCDWtc1vICA3bLepUDHSgCSVg7WWbLKugkUDZsh+xVfY5xzy6HJqVS6bELpBQZt0luUygpe2FoloCjmPR0mjEOMIFoIZCcX4XIXFfEd vNOSc8WIPWW6VZGTdl35cBuhvEMXECFq0D7i3yy9EsWUuMDRFvksesFBTTZBLJHM4UNwsUWGoxzLEwmzyjrJ6Wokiq6oNPS0Usy37JBWXqfc15bSHMjjGGbAXOjqlYBv9DKaH t9UBPoTlsijJl9BoinHgDVk5BEluDuQQfT/BgzTJVZkáelUNZq5Cxkza+17rUlv8opw6N0j8BGGH31OXPUXbNnPwBVgGuoOjAKrUGOqPŹlqLykGdSO1lrxlvKwQcaFyfmPBZlcFnk9MlJ q8cTObUNnbwLgaF7EHcasSNqgbluLfZCOsmhffltfl6Ag+TlQgcL9AjGB6WIJaoQ1K7LtoA02/yU6ktCliYcQmD6GbQyhrUn5ogfxyyzaS0LOLxCSbsmdxJkz/TJqgh6ADUFkXQ8o3lT8Xlk URMVwgJLZgB3k5QGDxphHaDoxY0iGGOcqsB5o3SctlCRgwbsDfEueW+u3ATGove0ASkkAuHoVmQLBqQVQJfpoSe2yySwikHDe0DOCEYiMBn25iJWXTDUiChrL7oKFLUGF9VV NBYGQaCvrGwSkViWITWPnHQ1ULi0J/TQT53JAjkBYxDv6SU7VoQtZtZFjUVnY5XpkFs9X5X2j40q5wJVLLGW2qlLSHseiE2nQCsROdhF6nclaL2Wl9OE27aJv1qbQlhmRDD/awK+h KVDy10hmAuLiHp3VxrQPRIKGdF18BByaX/kVOAURyoAglobKRMF9FzJR8YV33gkAVUIAo/kzyqig4fBXZemfTwCLQv2AmVRii1dGMOFKUDhDW0hwaLrwjhM1GTPVoAY0DgkG0owxChXhSlimRR3qlysc2m6VV/8haAehlWZoGjtMLG6WANoh9evFj5aMVGovRyuEf1iNE8QfkpaMCsl8pws5Cmc8cAAd4rH3pR2MleCu7hkqHcpxz8ic4f0XM5iCm7rVGhmqudmM22zx 8SJKDolfycVAZn+mRmGQp1BRug90dhSg5zB+ghCe4MmgEde9tLmJZtdsa0TW0ltG1lK63kOkRpR9lkygLVyG7ZoWDKIE7gAz9HVSgnb2F/h29zaGZgukgDQd7m32c3lCT2rlBgcpp, E6VOOCkdXZ7ghmeq1NE32AXg6RhiWBojd1lMLQ+lQiG1khdBkNrC4kTeloftSeY0QowqYvaE8zATTRhPthsIrUZ1H1kkjDfE8zATbSB9J6pPbAMbfK8J5iBZWjD29E2HKVoEgN4q2X W08HATbTh+2CZiES0VDrY6CYa0d7ITeu5aUQHGy2zHswgm2jHDIPUJrH+rdxEYv0bBTCN9YlgtrqJxvpU1t7ITeuJcrCLsTVmaBKEuoC2sQLTJMiNAni9m64YM7TR4l3qDG0yeCM3 0QaDNwpg2jiZyk2kT7YnmJGbSD7fE8xlkJOyYU8wgwq8ObVJgzhqapNebLCpTZpR3oib1oO5YnewHswVm7j1YEY3XEIT8sGelOk0hAiGdn6wJ5hBzNCOVvYEM8imrQG8HswV+yZa zN11XNHcfCNxRUsAlhhanRh0lKRzGaq4lgXgnmBGc2CSBNgTzKD00FqNPcEM6gxtwr0nmMG9B9qwZ08wgxshtGnlnmBGd2VIQ+U9wQxuEdFebMBNpLPWkzozuGpAecHRVQMK pzxSav0jhbcfKYLwkUS6lKsGP75cNXCx6eplY0ljtXQBgigbeipda5WxKphq56sG2hZfY+nXuw7m6BGMRtK5mkpUzTu7LJ9LMSGSuXHW4f9ZuTyClbuJCx+sddabvgF1jcsGfbP0kE4RgdX4acEKCwLl/vSWcx+3tUlfY0liF0UZnLG6heUJ8sa6zbQz6OsoZbSr9EkARdqyxsL3goWRChaFCe4rstzUsXkz6HUk3iLcnhx43tX+YMVDsUllRtvhEoVHWK7GCYRtrrPOFSbN02s4 pM8h1JO5i3Kk722Pa5uhAxTaidZUa7XU65ZUpAGrcGULSubzbLuAVvyU5QhTlpsyr99c+Pcc3kzrBAiTJhRWmFH0GC/FmRRA5SXKYbIYXQi2EkaZV4QLmtnSk4dnFISRdGZiyuutOf Tor7nfj3y8AgAyNTlffkkZZSnCoaU4ZeerLkliY4N+NdiyQGQUQFq+DxoVuO/3kphQpt8WDqivDuoud0JKKWakw7yrzdURmOX6A7JYvwWz7amxSmf0DQUa3oBzt0r3WCO6m6QrirYo +bAi3atHfyzoKRpjj9Fse4huCUIGFHbO0eJoaAYoXSd6N9BMQUypl7UzxLB4q0Yse+Ny8ruhiTAJV9yzYvvdNiqVsGWQzNWoXUm5xcBf0kycmkb7PR1lioleQF4KZdEFuNbqqAKCbSU jgkHoLi8LguYlzRLDyu9oGhjE8po0467BNNGhP0KAoGxGAclSel7Lgr8xZ2PYTfYtmlv7gbQYjq1qDgkic+dpXlAPi0Dx0VZJG9E25bhcEuPH1eZssbm8rUh6CnKpVlSNOEmmqxOu0BX rUFCdm+DA1hNuqX3vVMMMp+cZxVGktdKoqq45gNBs5nAUMipq6DCllfi88WgS8nPp0xNE5DvboN+dtFavhXjbk3ZFMAjabBiPMJCWujds6JEMFHDNuQJtWeAYPtk3cGY0SDT7Fs3 Gx0WbVKrTlSJvUj9gTSwWgbxyYFGp+51alZQbp8+geW+bbY+T9YpnH3lgLcnSyw0UltxzoU6x3XEN/iGUD2Dh55Knl0ih/h2abbaKwDSK/wiUaDl4Cgy4AnCkQxMLVXKHZFtslNfnTwF HgzLdwtm34K8mrcGhfdO3HrQHtSQIKGCe5DGXpC39+0L84i2bXwEFj1YxF+XO8nzA5pZmH3Ga1eBVNLg0qZ2YGcxw3F1y18RHNxW<u>YXCtRVQxOoW38KhcjlcSS+dVmosFyM8+F</u> cTol+g3E30xTf7/JDjFbZx3hllvShVbDW+as/QCfa5SoGpJY+Yxp7jGZb2HjZpy7oOmoS6i2KNqDw5ljRpYrVM17yRWsdh414LsVHMbyiFp8ujnMvsyyhY0m5l2KEsEezX5DzuRVZK7dL 0IAX3rDUbBl0j1MQBEl8ujh+5qHNCAvfBHrGKJHdGvomida/G8dVn3QCq82dbXAeTP2KZVkYN25\$r1h0oBPm5SX2CMrTWaNDNFASurjMPLROH+YdFiqs0PyZqtSk3KspXhbrhQ1a g84hdo7vN71fYv9xtMRO4u/BeQGp96HOfkkBPNjNlzUbVDAkhhqAlek0lhhaodkTzGjlgEQlgwknqdWgHpKS+GAw4SR1GkQwNOleuInUnxJjhiZq9gQzsAxNfQ6WMUjNMvFYh9YnDI4 ČSSMWoptoXcKeYEYntqQh8J50MHpaM2k+OdrNo3Abdf+BNNLd0zJXdNN6oryim2isT7QM7XxgT8ic0U20wRN1Z2ZjAK8HM/zCh21uWg/mim6ikUUiGNpxx6DokZQiFczG1F6v9EZ1 ZqNI10MZBPDWbKJp6Bu5ab0GvqKb1oO5Yp2hdRc3ctN6QX5FOqC1OjdibVqrcyM30VqdG8nOXX3TFbNpPZgrEuX6vumK2UTrSG+UTbSOlDoslg2CL4OhDW+prE3y+Z5gRpMrUkc 4kBAkyxIXBWI9z55gBiuUNHW/J5jBciltdLonmNHaLUnD7gImdPGBpNT2BDOowDRxtCeY0bieRHR7ghl+4QOIal36JspR1YodYNILDneACQ37I4UFHym8/UgRhl8kOULaAf7peQdY+2 ZzjJHJ/txyXVRiIQXevweiGR2SRGDdd4AHO8BCGHu0BPyMkk/h72cHODhzfqf43aby/+8KslfDw/s9u4PD1d/PCrDXNJTbVoB3ONUPwU/m/aG+lmFS5EP9nwibD06qqWjoKBzvlyVcf4 hEYpvIYQ31fsQli30vp9vsisIPfh288FdnCpJS0SOWuDt6Z5dl4qvJ/HaLgRmYsPXJTQy+NuxRk0QvLjNemNG2glOys5PNR6Q5AF0PSvmwBBpqSUq1kuoa+P81O87u7strnYb3M5h1 IfQn/aR84ejYmlrNYi0XRBS5Xl6ZzOnSkWQprjne1ti0Rlfu40D8zw0O9TVYcCaySTaP+0d7pfHprDWKgzK/fdUbRdYBIa5EII/aFPvNnYxVCYvxaRi1SaMDH0qzCHVSJ/XO2XJSspv9dR 53b9Cxp1SLN+L6d/kYzr32/jHcuxT6B4Vo7H5YvV5WTRtUhUVxXc87O+RZiCkiFYbx0vPFx01Oky0Rk0oTSO5iEyBUGIjAql5qqi0SAeLnixxi3PCuVycrJr8hnN8iB02R8qzfULmov7riQ0 RiVbsu3fUdWvblpVoFr7zdaqKjRmj/VktK+T9Sw3eoFkeEi/spN+iufi0UBKa7HOFCTKLpSCrW58JVjQEXraUSwTrhuXh2iAN8QaNO2ebFUtt51b9szAeL8ZcgCDQArUmmf4QnYl3TGgP Ds8a7rXGOsTHa9QsflKwl38TNePl+gvjBhlX7N4GXGkuT/owN5ok0njtVoeRJFq7kWVopftGc3kax95oLk+j2Ftt7ZBljTj72Oqm9WBG02dSal4GQyMRatEjpebg+lTEIsR52daYWW+ZUcx sXXpYbZkrHu2vBzNKbVIFHZymkzQZEQytgg6OT0hylbo3Saqge4K55gbGajcNLEOroHuKq8HxCa1O7AlmcHxCS809wYyOT0g+3xPMaGuHlA17ghH3Q4l1hwT36f19en+f3t+n9/fp/X1 6f5/e36f39+n9fXp/n97fp/f36f19en+f3t+n9/fp/X16f5/e36f39+n9fXp/q+n9y5j6/wA=

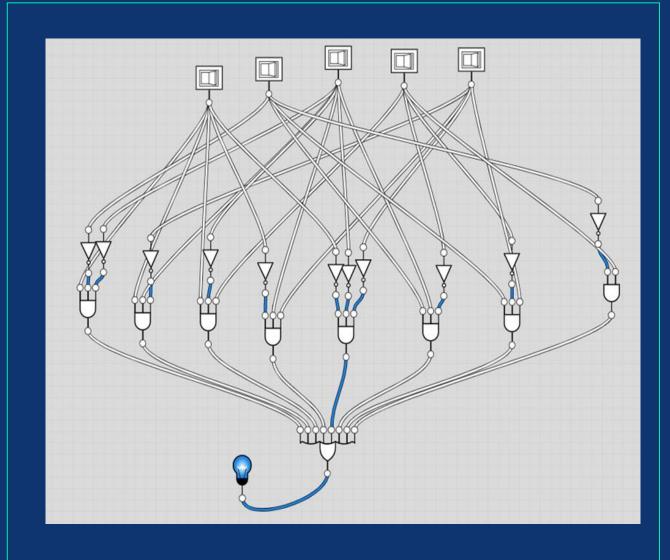


VDI-Link

data:http://ns.logic.lv/1.6:base64.vV3LihzHdvwVYtbMUi5PZhqXhhc24LvxxrC3Ri6vaFAcQRzalL/ekV01wxl2d91TrQqmlGogznRHn0dEnKxTxb98evzbx/Lpz3fffvv0+cuHh1+fnn7/h 19++fxlOv3B9OnPX9RED//4l/L1y9Pjb+8+p9/ah4d//feHd09//o6vpPFO5UQiJI+EDabjqy5F0C3qkKytZvw0910e83+38rS89pf//fhUfv2n5295ePf1Y/3wkK1ryrggvCUprM1BpBSl6ETF FZUypfTw7tuHByvDw7s/Pzwo//Duj8en9PTx8fOHhygf3rVvvz/+8fRvp4lyfw/vHr8+/f71CbB6+vSlPfzCghF96pmiFb7lJKxzXqSmkyDfe5dB+mz0CYbTZtlnlHpS8fUvtwrs208CC86Sd baJJUWtyggktdKaG2gohocpXlCRo4Q8wFMgPAGWgi0hgz8JDLrQ/LeW9Gi8cJggiJaj9lpgSZyuspNnpD5YF6Q6S3l/ucnkbkkgw4+igpbFlZliQLuVuRSZLe6du/zCVIU35G5Lciv30 WWPtezgBnnbU9BoOCRStUcEAUtek8RNV8cQJ9gKYqTdidcVspJOvnyS/+I6uPnGYXhIPBN6ZKKFbElpCz0LEIjK4wpeHfTikt2DkygKSwltJ2CdwchIBMrpVDQWDYKS96LYGIX ruA/scpgWz8hCGZ88BmBlQciaDU6qk0LV0FstgTUhUZSklEq4l1KoTkLRN9jYN1ExyHoeB8yBt0QqrDGgF8TEPhkg3NN+lrqwjX0PQbhwBhl0jkn0FttrQsbrUQlarQuuaw7qZYqLa SrXipxEN9hCKgMUalbRA8KCDr6NFZbRXeSsiNFuiiTvvb2evWgA7PgfPW5SCVUrw69QF2AYfMliY2UenR67gWl/DNLGK8n+70ZrvPQZwg+Pz79iKBbwmdGD1KMENsOxY0gZ JRFN8GUVpSlBYGdtU9bP9Hr978KQHEAKFJmMJJIMyGdvkpSyBbABrWUEpcQRDuZ1+w4h0NH4hUlC42E4EbfjJB6kFO2SUB3qzDF9Z6lVCiShR7d5F4r7wlmTPY1GLcLjaP mjSES8AMN5eE1ysM64aijQVKTfdFdHS6hMVC640KTbfUkuxHJOSSqRi+ilXgCKAZYK9Zs4tKvcvlXwKi3rXumbZvQBJlcaxlvL8NgjxJBnjBv0P2Ym5ZeBzPzl6RLZWOMedvHdhc c6AksavNYN2i8DVYiLk4JqllHk5lzvc2EHt9g/Blc/ZZZzS4wHkWam0MiwdfCFCFnscYqdFHFBO9d005FXcw5GHNoEfdsmvFQ2vv9Qd2A3sBzoFpV2rC3WebZ1ZLVF9HQgaEpJ QSLXhZKBfS3SxqzhSOhqsEIQM73OpdNcPL2ofE+aWgMbEAGx9gWjAjKD/JFeYZSi4llsULqYn/7NzW8ITSPf5wNH9qrGFAv2IYCDyckbERGN0e+tmCDjkvRxEm9dYN/DucvJ/V KFa5iCWdYPn38269P/5W/fspn/Ad1sipEkXsYc5lEuyuPrpIYRFpolaiNlKP9m3efMXkQ0RvO+8FAY+QElvL4+TPA4BsWO/0ff/1nrhDMP/DXz7V9O7366UOefp7ns0/fv/y4ug6Gp5F HapHXwfBE6ToYnutiauGJ0nUwPBPKTBNPk66D4Xnv12D0dTA82l0BwxpRmGB4inQkmJU08bT6SDArBcvTaOtaeNM0Mzl8dbwOhidYMvPDm0vua+HNVszl8E4li2TalW7inRO qhqS3HcxKN+0IPZ5tvVOathvyITTtjQxvurhTN22PzA3TxJsu7qTa20eVNTCsUeM6GN5B4GswfmXWZnnYI8HQypkeyxwdCcZdB8OzAEeCsStgWEcaR4lx18HwtOVIMGsnVyzS OhLMmu1k8cSRYNYYmPVik6THOtw/A/PpsZxO89+NCwR/iEsHp4sFX5qv+OMrfGr95QU4Rvo9R3iec+zVe45qvGeJLi7UL8s6HL6a1+p+WK/7z3952a9DPGSqmkREPwnrxyfle AcMrJq8UrWSPHa/Dm5DolyCCC7C1MeEHDIXRZDd59ARBjVfVTdkJgX+k/h20jlsV9Xl5KQP1pCOSvqobrF8V1WiEk0ZNQNx0MqL3LoSOUrkNKacw7KOMq4BAihplktmXOk6b QROHoBBTqhBjwTfYg8PZYNOL0lo5zCpKTRf8CZAUpE0bVJXcVkQVHKCwTHeK5lmOmdnkDQZ45wLCgWqTLBnly+P2MmTmPYrVRiqnEbJkhMhpJMFQBuaFtAUM8pgphA8 AhaNifhM2vP5s9t5KH9bezPCOY0evHABKaELnagZugzRg2WikevkiValmTTax+0Qf2pNr0cJ0waWiYWAD85ApEABBFJIBgrJvVzCF6dr5iK7gN7SNnrpX3b2LMRbaTLOBnV1ve F8T+nSekOHg4ydRKiD6zuYL+mshCnJh9xVcTRvzAmaHEm8H/D4KMGBcwf7cf3aGm0CEcpx3yaKIYzfEra9thEcB3XOEnwG4wDjK1MIZU6ecZOUCv9op1BmNldGBztJ6yVKjn yMG66aX0hUwujQ6jjZAqMJK5WCHlCpc/Ky1uZOZHpaGPJxktHqACg6SFr2yZC8yc66pZyWjvYlSjZ8LhC5aB72DQoGv20x9ZlEVcNT2rasd2qJlGgPHEGOl/jolkTRpEAuEuRiQj R0tk+8KTpGkUe5BgEdBhXYDonTyQgCM0hluoWmLpnSk3bSeeQLqkRmCY6eEBS8iNOglty2v7gyxggN7H/SIRoKGIDg9gpg0hoCukRIVM4RnWafykzl2IEDQSB31c3DMaDaHf otlldu3MgRdtyklNFGGhbNJdyi710ln35WDEjk1bx8CyAQpVMqaU2r8svgGrfSobB8QHYwMcUOqLi1UgVUCxEVgRrKYR1DIQWXYoD58jawQYL+sdxnlsVGAZl1HdJaulj9BohEz dL+RW4JzaTk0e5PrsE2gXmCwDrNlbaLihYlggJKa93SEch41gmrVsJ1SxeddVdR3DN5qKY3UdK50myon25bK6W6Eqk9TQIKBwfilf4sORIVX0wLHEvpZDcIBJ7uolkqWyCG0mN DhoGN9ZmE2RSd725wG6UBK42grl0lvHm0VbC5ibSnPhev8g3phqiMKsbHPYEZ5k43llUwWgZKyobGc4CAbukkYktl9xCGFcQUP1gQ041gUDTGYtQodM9kQBteBi8LinlY2aPw GY4c/M9czxds288gFVAa+Q4IATe4C/KOFkapJSCdae9lrxtsqhQQ6P05M5OyQkLPJWxRTMB4h82afOgSfokH3oG4t6sY0BzoeYzbKumsVAnptvj1Bxok0CicozAhOxqWINXhIWz 9MG2CGXXmqrSeFCAupERDbHUZZB/5pGfbHo9tczssinpwQwtHJQzTls2xCGqKNQE0gQZI76c+kJNFEwjYYCau6g9/OcBgyW5R2R6IWNKhhCbqFAQvOWL1slaOGHdQb5px kGCnchYYwG7alFvLxdGaWtYaOQpWhlzli5naLpfDGw0OHCE2IDibwOTZu8oRpWCs0FNmriWJViQvNJAOPVRAa+BtCVBoGw6asDVlCrhYRh/+cTvZ5IEEhL2A85iViaHhBzF1u X9U0TDrBuA6zNeZdTePQrEmhwGRd9txrX0ryDyJ2QwBIY/KgxSoPiOAeCpg0kaZ91q+1nlGWGTU8rlXBX4L5MEIXGObqM5rez1wDoUdDeVLDA0ejl/lHTxFB8ZAKGKC4UzJ9 wsyVQxTSjAOHouACQfxCy2QaJnwQ7LwyGZAReF+oE5hGGbNMYx4SZSOMNbyHhYpvKOELVVMCRgDnIOCwbaBhmNCgKkxFJtBbMi71OTSD9adAADRo2LgFjiGUjbeRHK ofWbzKfDvv0aBeifSYUTFWZig/Aw9YbAINd4I1nScGGPDBfJhFEQs6Ee7pJqOkz3ikE5M/qud6EXN2W5MBmvubBRUa5w8ZFICNI5vSTYFmhuxmGwp3hRhq9sd8BZZczh/qhCe kMloU9ZhtrmLZtNuaMDX0XjC1qOG30OkJ1A/acYaQKnTX7HBQJUgHkGG+gwukOVuY3zHbnlYphA7WcGXPdZz9XD+hZ412KxdSWJMq84CTN9kdCWb1ujrH9K3sYrA8LBMMACWb1VgHACWB1VghACWB1VgHACWB1VgHACWB1VgHACWB1VgHACWb7C7DoY3pzLB8Aap62B4YyHzhJ43Rx0JZm0FmDVFHQlmJU08Y76y2cQaM7j7yCxjfiSYITTxDqSPbO2VyPBOno8EsxlZ3uHt2jYchzSZBbw3MtvIYCVNvMP3lWUiltBy5WBnmn hCe6c0bdemNTnYGZntYFa6iXeZYaW1Wap/rzSxVP9OBcxTfSaYvWniaT5XtXda03ahXNnF2FszPAvCXUDbvcA8C3KnAt6ephvWDO9o8E48wzsZvFOaeAeDdvpa3nEvV5tYn+ xIMGtpYuX8SDBrhpzVDUeCWWHg3a3NOojjtjbrxVY2tVlnIHfSpu1gbjgdbAdzwyFuO5i1O1xZp+Qre4KsqyFMMLzrB0eCWakZ3qWVl8GsdNPeAt4O5oZzE6/m7mSueGm+k7niN QATDI8nViZK1nUZrrliFeCRYNbOgVkW4EgwKzzDGzWOBLPCM7wT7iPBrNz3wDvsORLMyh0hvNPKl8Gs3SvDOIQ+EszKXUS8F1vRJta11jOeWbnVgPOCa7cacDTlPYfr33N0 +z3HEL5nie7lWw049wBYqqGlOu67OuEcpYUJz/uWazl9eFq2wrWaUGLSefL4f6SXZ6NKP0kVlpGn4MZq0o9X4X9qEXzseJ4KOvEJUA38lvHeoihfxvpXKmHemohioSSqc1hu8o 7sC6yztYmfgtViq3XcwZKVRupJdhEDKRFVrFZVr6RtyyNK1RQuwbKTeg3rbOHwp2BFUh6ND5bEK4MVkkcdVSc0SsSO8wTT560OMnLSl2Dpyb2GdbZD9nOwCoJlvRbV+EFW 0opsoLu9l1i9GCpu2aVAtOQ5rBgnsydYl55yW/A5Y0KQCogKkYKojJtsCB1lgs4pLo92U5Em7Yx7gbQscRk9DTTGaFCMd1cXRnlPewVblnGz4emGepi+JEJtBU2ekLJavaH5xhGl 0WbRfo/Qso7iDECSPAaNiTKMu3yEMm7cexyBVh5eqQx6zzkVVPi8+SzNGzDLzQToRPsazL5nsBpb4MlrHLEDq1sz7qpNmBWybaBAVctp4XowwHil5ikaR2/R7HskbY1KRxC1I BgYLBQYvtGr4a27q6gp87LEhaJVr9CoZQtbT+EwNAkhkUYGUWncKQbdh/ZELXxL1tdceorypa/UeWhsODJRrroUFMyaMgRP7XtHoirkqteCCoY8+rKs27k3aJYaNuHA0CAgJFu 2QvqO0CSPaQMFAmJMCgagypKWdXnnLtawn+g1mKvbdrwaTr1ZCUHXZYirrCDAqkA+loymhCGkpOWWK/mWbS6SzfXdP9YzhWsj1SzqJLuugcZgxrSxgo67ksA2Gm7hvh+kfo Yz+ovTKNaSJaZ+20aEZFGRSBQ6Klm4GmNhn4lLsNzlmfrsBEv2Q2wwPU7Wmu9EvO+5iSo6FG1xQiYEvGo7xh9MHA5+spXSaLYucJvc6BWcGQ0aiV6i2fnwZZdr83acb7s8Lld mkapCX3nlprbjDlWz0l23F9D8GJt9D2cNRpaQZBQ96zroBn5VZ7h+Cl5aOAy17NM6r99karkFU/6AZl9skqlOy9yQGgsFB8HAU0MzFYpY+VYabNgSG2WmcF44Bpr5Gs6+fXmjZ VMew4Bi4+JlhNnPQIHgZF9gFkOVz4/NVxfRHF04GF06l9Cfl+N6jTdWBFhm0WowyZXa4XJmZXBv6+ZKqnY+8LhS8bHCXXWMVLaPZzqogh5H45MvysTlNrMQL/WUGvcDHha aGsad8XCfTY9DMTQRprgmRB/6NR5Hk2i5sQFWv14IDb1Fs69sqh5nGBqkWlaYbvrm+Bv7F9XWpvroeC0Xr/W2bNQzFb+p4Q1cfL6GLYMuusaBJZdBxShhiO6ioudLr7o1SUvR QBdeqdQcGYx8U1QMS3y+hn3hcccoi9AV5r6k0d0W/iarPv6mGd9CthkKNo+j0Qco9Xcsy/q185P5jsVGvmBeXwlPkDxbLIY+Bydhqy8iwOuMo7HTeglpK5+lykzGfw/Fy5q6omgt5B xm5+3dQmtHyiy5XjlsZ4063lNTVr2uLLaxZgsuGJYgrYBh2TlmGB6vHAlm7Xo9i/9XjgdZkwX3CiOL/leOB1mDBRMMT6dX0sQaR5k1w/MwR4JZiQzPbK5sMrBmY+Y1Ed5YsHldjX WiwkwTbvq4Esza5U7WQe2RcrD2qGPWqePaYhtH27iLA6xD2SMic8M0bRfKG6aJp/rMvPCO9I+MzA3TxDtn4i6c7Czq7WBW/7aEfWnaDuaGaeI5RSYY3uWMFdJiOUUumJ2tvd 3prfHMzshsB7NSwHu7ieeh75Sm7R74hmnaDuaGPMObLu6Upu2G/IZywBt17qTavFHnTmnijTp3sp3b56YbdtN2MDcUyu1z0w27iTeR3qmbeBMp97Clde57HQzvrJar2qycHwlm7 eSKNRGuWAhWZJlbdry550gwK/uHPHd/JJiVzUze0emRYNZ2Vlke9kgwa3cNsJzakWBWGJhnjo4Es3ZczxK6l8Gs/m0JHNJam5s4V6bOwLxsiv4

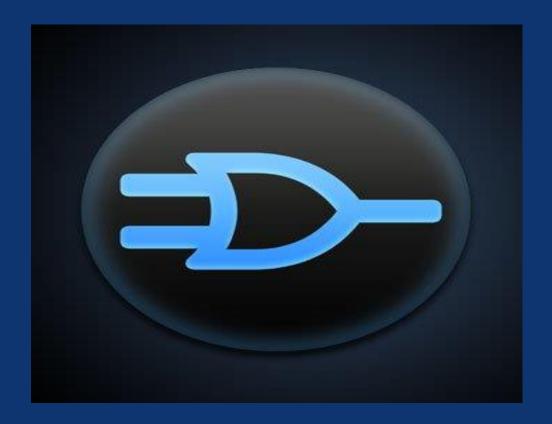


VDS-Link





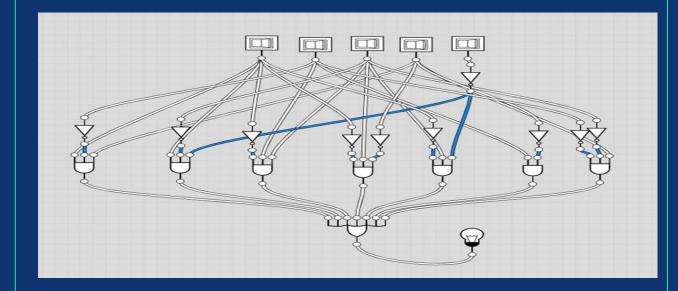
data:http://ns.logic.ly/1.6;base64,vZrbbhpJEIZfBXHf4z5WdUdxtBd7kwfY61UfY69Yi GKsjd9+a5iJDcEzKsRgbmxhoD/q8P9VLX/e7L495s3L6ue/m+3T/fphv//+6e5u+9Qd/ tBtXu5UB+svn3fpn5r3q/3L93q/fvrvcZ8f/vj1kvXq+bHcr5N1VRnnBVqQwtrkRYxBiga QXVYxQYzr1c/7tZV+vXq5Xytcr37s9nH/uNver4Ncr3bP++/Pe+JocfNU13escwPGliB YgTVFYZ1DEauOArC1Jr3EZPThXKdNpw8n606F44dbhsQ7C9bpKmJSWtiMRUT USmhtioLiHcR8IAEHFNWeRCh/wul9LlJi0UdEtKIGg8JqKCJYCyLkWCKaplKVBxT 05hVF3wTFRVm0xyCKrklYlbTwulmRcpbN6tlQ0wElqDcUdyVK3JazkBiHtkUvqEop O6o6QvBatBYDFWp2RHngUBA67Q4qVspOOvn60L9jPG4HCsMhwKp0jtmKUCMl xbckfAUrjMl0uqnZRTtEwkPnRwJtO49uIQIwoUD0mZrDBmEBUXgTmnCZfoQiva3t QOBN/8UHAisXJKglOChVC1caqUT2VAiakhKNUoFOyRmGLAC8xcC6DpYjaHQO GEP17ouwxkjhlxFgtN65KrHkMuoFvMXALxgDCTqlSBJVam3CBiupEjU1J7ikG6ga C4xKqV4rsRevxQiKDEFSN4imFRE0asxQbBHNSUiOZOnCoNUa7VslqqWz4LBqvl IJ1YqjXoAmSDRTHxlbILbg9NALSuEvWTCoO/vWjNME+oxqu9v/TtAs0HemHoQQ MsUAswjKJyqLZrzJNSsLI4EdDEtb7OD4/EkAxQFQoEyvSClOgnT4LUohqyc1KDn nMIYg2M4cy+EQDh2AV5QsGkmmGbAaIXUvTsIGQd5ZhMmutSSloiIZ5dF17tg9Rx jf2WMYdxWNg4rGAAjy9ErlgZrKwzrhoFGDxCrbaKXav0djyMuWC02yBUE2I6JzlKg SUAQDZPPkGKRaoSQTxn6VHb4Do05b98zMLqLxMrpaEx0vfa8eOZB40sRFzh5S 1RK1N4N+SXivblwxp31sr8lhP1GUI6Dxi0zdeqspLk4JKEkHE6Mz0Q2CHk5NfQyO PIVWcxUMUpGm6qiRaBilsYdyFkooQmeVjUd0XrtXdzHnMGbRIm7JVIPktkmioboh eSOdl6IVufYjapJpmEzB6ndpYMHQ5Oy9pV4WSnnqbxe1CNGBUMXQ3A4OWxnK xit5+9AqRk0eQ2NAlo2x1RvhFfbiS+Xpc8km5HEUUu/2N57U8AWh2f04WyA0quCp XrQtQDocKWF9ZHR1gKV663UYiyZ06nQafOmHedmpl1eYZPFnLJvHbw/7v9PzJp 3pH7mTVT6l1Hy/TElqd4XUVZJ2i+prAaiD5Gq8OX1qQhKiE82T/dl5t93S4fTEOD7/ 9fVPrvAPb/i6LfVn/2nDlzq8nzdXH14/vl1Nw/A8cUmYmcjwTGgahjdlMWF4JjQNwxs 6mWniedA0DG8GP4bR0zA8mZ2BYa0kTBieAy0JM5MmnjcvCTNTwDzJn4bhbc/M yPDccBqGt0gzI8PbRKZheLsUMzK8a7wlFXimm3h3ezOix/I2Zpp413vTMDxvY9vBd Wm63Jtm0nRtZC6HmSlg3n3jTDexXP+D0nS5a98wTbwRhBkZ3g31kpG5YZp489A H6czlw9UNu+lymBumiTcpflCaeJMid1W5Mk28SZEJc61rXz5c3VD0LoeZ6aZrRY83 tn5Qmi4fyGfSdG1keNvFB3XT5ZG5YZp428UHufblq8ocDGvVmlbhXfwdw+DMrs2a YZeEgZk7PdZwtCSMm4bhjQBLwtgZGNaVxplwZhgG5y1LwszdXLFEa0mYubGTp RNLwswpMOvDZkSPdZl/BnM3/nPWI/8B

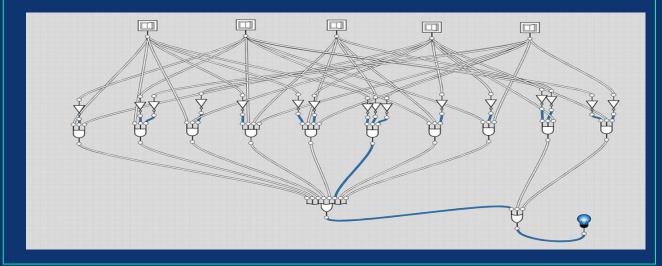


Link

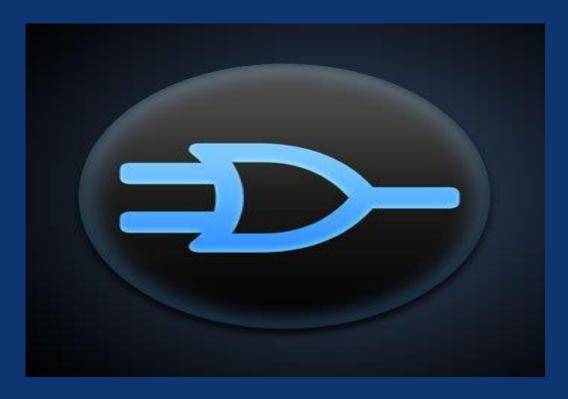
- Horizontal central, 'G' ou HC:

 0011111011 0011111011 00
 (a Λ ¬b Λ d) V (a Λ ¬c Λ ¬e) V (¬a Λ b Λ c) V (¬a Λ c Λ ¬d) V (¬b Λ ¬c Λ d)
 Λ d)
- Horizontal inferior, 'E' ou HI:
 1011011011 1011011011 101
 (a \(\tau \) \(\tau



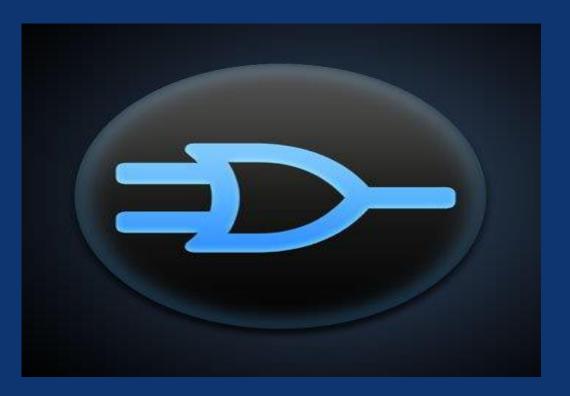


data:http://ns.logic.ly/1.6;base64,vZpLbxvJFYX/isC1q13vRzAaDJAEiDfZJdugnmMFMmlYVCLn1+c0u2U3S bHnUt2SFoJgS+TH+zj3VN3+5X73+12+/37z9OV++3C7+bzff/3Tx4/bh+7wH93994+is5tff8mPD/vdl5tt/FJvN// 89NfNzf77V/zlg1ZB8cRc04bpFjyLSlimchZGJ9WEEvhz6tvs0r9r3o+v/fDfu33+/Nvzr2xuHu/K7aZIUazyleGtG 9MvNeabtawGn2OJOqVsNidPtxupZMenX2pz8/12o0QXpl9uc/Ntt4/7u932dhP45qY+fd192//98En/t7nZPe6/P u4B3eL9Q918pEFy70Ufh+YqZ9oWxVLhgTkRtbDWSuncAdIYe4wzMKpOOv3zH+UcY3oIY86J18Qjsykkpn0 fSB8kC16ZGF3mrskDo3amU+eMttNTRjHH+J9XMlapa006Mm24Ylppy5BhzXSuWbSCt23pwKiC78yyOOZ XMqaal28c1Z8sCjKYwny2gvmiEeHkUvZDHFEPL+X6mjg+/SFj3JZTQJ9C86VFJqUJ6BgfmOeysJKrNkZFG bl4APpwlGc/AkrV+UkbnQHebQcgRYHh1SVlikcr2IKMWsBE7pnlRmZvjdYmH2CsDR2fxMX/gEE9/oTRi2iK1 yaG2phUXCI0QrBQdGCmFs7xLjFzPoiJtp2d0DyHxqPAftKc1ddVMDWbzJMETAoVoQkoJF0c4054G5P3yp ehkLw4Cs1zsbvOT0OzCEY0G6sJmaH9kKcGjmCbZSmXVJxRuoShqo1WnTvPk5Zd+MnSa+/LMPoMZrvb n8l0x2MVtbJcDNLEi2QxucCCtEr7KmwldahgfawC+gAjuTuaBOJM859pBIUGlCajMlgLFalkHYrGZo88Vc5R Q6ppO4QGE+hclA6lNM2TWURjRAy+YR6KCBCd0Oc+J4zHlkLxohkbh1FjNepjlBD6mOBidkqE0ZccFWoD+ oJBliCFmMgQ6SY1prVDgJ7rVnZyJJCO2Ma0jBhuHS+BqZAQA58rSwZdXbzMAQMjaC7GYn1pTAihjjtnWT ykyzA+PrMUBeIR4QACksO0zk2pgL4KQ30o4ztxXh8SwjelsXSa3bez6aQhUhySb3k/nXzgLAoYpxa18cmga uQIE+yx3g69Y/TFZjFnb39/9/vn/b/S4306Mxu6ylCrZhyOB4YooUaK4ownHoTMtuk6GiLhuzDp1aFeDBpbX5 pBJzOyd6cAy7vtFmT4jXFk/uPTX6hKMvzBp22pT4eXP3ziw9/TZunh98c/F5dhaC7xMgyt7Kcw/DIMrWovw9 Cm1xRGXoahScqaMDNpoinsZRia5SFGhjZ8ZmqGNNepBUw6i63ZTTORofnwyzA0aSB2E60114zMDMzS NNFcMiFNtGPnmjAzBbw0TdfDvGGaaApOhKGdvC/D0BT8nbrp+nEwA7NUZ65XYPV2aboe5q1bmzbb3qlm ro/MG9YMzYIQYZam6Xo/84YKfD3MTGRoU/cyDO2wNYXRM+OANFvWhJnTGZKCrgkz54FJrbkmzEwB0 wpwTZg5nSG92GUY2jH9DOZ+lw9n8Zv+qP+tvwQ4HPsfiC94+gr3tf14AYop+kA5SH+gyN4Hyjz9QFJzfKiP4 9oLPw37M9I+wReRRE6BZaHR95E7FlufSFeU5twaZYfLTiVN5+XkymZccIXOTy9kz+5vXrflyDY4DVXMaHd 8cFVZLL0UBG2FM9XrNI4zGn5ENVw5atGJ6Y312U35q6AkWizk6liTDolA3JAtw1mp1kARsuRFD7dKNnRy GqoBSndqCnV2Y/46KB9Ss1BrgeJGHUXOvABj0apklyMXLg57tSA7PY2UfsNQQb/huVH4tVRUdxNoRAMqt ERtRWQTzXBJC6v+Uv4WhuryzWCwlbvsI+MS33QSDm1ZYfp4c4WXWNW4hrRCdk4e7Tf6m0EEUV7aep zcDP7tz6TNhw9Ba4xcKwqKCocmFjBqmSrRWTBa64bLZC2Pb2/HNYwzR+syuWxDFarzIYbIam4QHFQ1S 9JG5qKp0AfjqxvUAPOm8+dLIX+8L1u2oUI2MGCKYK1K3ht0A501huUgVMy6NZ3jKE2ys2YkEEeX68sIdJI ZI04xVVtviirKWFeBMrYtleoiN0PHe6NeCofzR5uPZTCuGV2LdUyWjGM2Jq2LGDVMNpqC3WJ1bthliqCOF3 bmR6mslxyYMutq30NceNRt01Do3kk3xYN3laschro9XR8+J2rNZWZUvf+RlhkMBqbxxXxFqFSQOUmXnRm 7yPWbwvMu8sdL54swkrKRQZ/oUJNksByIT/RoaetgElCzTjkHmx3HHZk53TMc9kO2s1OYZRu7GjLOc8kyV 2AWdSyGRZzhUM049koueGt83Gaq542dOyFYBJAV5pGEncoeaguTI1hymAUS9QKfpkQs41MVXHTjCtUb WheTABJsm+mNjYehA4CMaJzD0Qa+J7je7gyiZvhJcYw04aRxliVE4QghDBxX1TbDGeKbr/ASPBpvtMc3M 9aqeXGh6zWtcUgwmisIVbZ49wKFy7CpseInEaJ0OPTDgA6iotzxUznjZtUSu5i2XC7JVgFJwxhDZBLGYEjw D4pbJeBIYkuDpnjx4irTS5rckmC4KQ2OQDCjHE5aKXmWYs4Mp6+ABgFqaIPAne5V1ctVsyw00fmoK0olqt4 Gp8RCwxmHO1VD4c2750e8vH0pNOjo9fJUZPDCysJ4gTfQClMgSZxiKlfBGDRY7XfaTwdzc1w1z/qijnbOV9 Ccr5yzyLJyyEtRQSAy8C1JwZ8njKdYssXZYXx0pJcXfZan/pm4qQhfhHEzB2HSceXyQZg2MYinclpLz1xFkh wpeWdHKZXLMDTTQ4ShTaKZ/QvJkFlXiKSpNFMzJANGrBma2K0JMxMZmtZdhqEdYoqwtGk9s5kiOVNim mgDciZNpPMMMTI0Y7kmzExr066x1hS9uc0U6fZjJjlkBac+aUO6tVozMnMrj4Vpun4czKSJdnU2Mw5Is406t UkWYM3lvGGaalOWukBc2E3XT+0ZnVlcM1fDvGGaaFOfqjMLl3O9hZhL00LRo5mzd0oTzQ+9U5quN1cza VoqetfDvKHOXO/03rCAabb1nSJDs63vVMA02/pO5up6DzxnIUgvNjO1STcpRBhaN6wJM9NNNJ1YE2bGdtI G3ZowM0/a0PzImjAzzyDRPOyaMGbmsR/SuWdNGDujwKQXmxmUpLX3WWv/eKTk/w==



HC-Link

Y7lqglGFuqZKYgsOnsYRcSSolKilBvydnOWrD2TQbaTHY/BA3IY5WCNJClFKlrxEIXNTemMpdyWWJNIZNBn5HGXArU4olbbXWxnTQHJY04brWJNmsqLHYJeiKV0vsgW83BlWHw0ENH kBB41-GoouBluk1GZmzCHlenomyWKHbVgXHEpsBEKwa-YB0hE1dLDY2lKRgy84V8ihHJyonOgB29EPn+uzzakNW/M8EEbba,Dl@YYwkYYRWkSZGFigsCPs-org1k7/KgBB
ASCsLUDisTgHpMO7Im88ekSbkmM0VmRxdzFiYGGSUDUICDa4gmuDFYkHrxbmMqmtcSsgJEs4dfM5jjzLnD8plBmE1Gy1GVW3Z+ECF6TgJ88CGGdvjgLHyurdisShUbcihd-kmdelsb4pFYGCEhwLyoTIFGMgaoWSYFjBUUAuhhxigrpuW0VGs4jqTXhfr7noSMMEE-dNh-T8kKrif* not xy9uL5mNUurlyUmOMgnotX2ygonbzudeRiBLMyg8bNCgaCaCHbyyGEseR1 a1CYyDkaZq4EjgiSBF0FixoTCZRVbxOxOleciox8h,M2;WW 1JVOWTbuJ2C3SC8lc4h1pxO71NPM2s1mp5EY3dUTO5e6/hy0wIDi82Uak2MJaJoJACWONamc3GG3570TgXJXIMaEBCJ NUZUZARZ4EJMZ4GAJ7TTmx4MV1HJWEgngas9S70s4hCEvryxZDXWieq1tEzmjCJUZARZ4EJMZ4EjFn+G86DP1/n4-b55SSWMbddLG58ma37UZ/P1sEBMESqyfX3OxC0 8UB4fdxEMF01Y47AG2SET juji8 HAGZYYHTJ/H93BPMEN+F72+X+Wn0w8GOHVG3jc4sb/11d8PLJXXC4dBBJLSdTADHLGOMK18HGSCHRTbscd8bMj2JMg5HXwbDC7gAM4DUngqF1 pDBBNRE;9V7ghYNlCcFX4dbgaJkcFix+igaIU UTK0yuQ8GFpiR2QMrU042wQaseBOTTgleqTcROTTd4HQwtf5HTwTY1c9M2VlccFXAdbgaJkcFix+igaIU UTK0yuQ8GFpiR2QMrU042wQaseBOTTgleqTcROTTd4HQwtf5HTwTY1c9M2VlccFXAdbgaJkCdtGMAMbDptJg85Eyy0Q1B96H1DNdECCFEyH72pbfSc2pf0fbcWV19cbqMSdbAFCdcMixJSPNtcOTTf7layAMkhQhaBoDvTQGGAGANHAF7 72ppfSc2pf0fbcWV19cbM3cHRC4CMsavNYRtMLTcssVYebFAT2DD0maG4sSeVUQCmdfggdGaS42pf3HJZe1ASA6EVG)gcFvxkMD3cmfptcPdpgfTE8FfCr2SSs2HULFDX NMg6HX81 JdarG672X1.nieyAP7m2JLMCmHb3U0J4BhSsdjohSrF83AbsA0OOHMinwBSHyJU0ZExhipa/HINV4HyrqryalB0lfx+3kvWJ1PhjuvZVBd9lELYbvliog2BSW7zSASSOFYqlZvFDh
Oostyl Commission of the Co P1DHkYoUA3pDgZFbOIDFyoppFTF03JSRpuHPSF7GTVlhw5QSj4ECMRriS/OjNEEg7ovZddGhE5BMJpAYFJoghoEioRISHxLONÚbhJKewglOVIG+Slc1Z3NwN8CVGW2jQwhr+sYl5wogd LpKBsyu5NMŘteĚQSYyYp4+BJAJqVAlrQ6qccvgG3Klg2U7D+mgZAgrVHVpoÁpRxŠO5MNRIGvUIDNmLBBrUŎq/hBQnYLeNdCulYCUDSyǩE6i1dZNyFFQ2bwfsXXCOIScGhyKpVOmxŠ6 gUEb1CKisolPthYEKip5TocJ424jsFYJ2sIFoJJVhX0H77TiXHFpzzPdKsJusZ82EY08IAFRBAYII/4b5ZeiWJKXOBoC38WPeAgJpsgFkvm0CFysYWHlxzLMwqzSjrJ6Wokgg5SaehupZhv2c GnvE65jy2I2ZCNM8hegOCUgGz0C5hu3IYH6BAkyylkX0OjKcKBNmTIICS5OSSH6PsJHqhJrsrUwCikMOcqeMykve+xDrHIL8yp5waJf4HY4ffUdU3Rps0cdlEsA97BE0BVagzxRzLFReVInX DZ34ZXysEFGh-75jwnSFBZ3PTMCavHETm1Lb34F0MCi4Du9WwG iUNwiEvs2HWTOn4WvzsgjPk5UwHC9OugsFDWENSu07aANNv0lOpLOpImHeIgehmUMosu3+aOH8c:MZkA 2800k777xy/SZ3tDUBHCN1ygbZv0H8cRgsHrYnCSGgROPALSepSc000S to SiteNe4Ribe EDBVbLFDb3cDFgbs57 sfkN6c9k3 H78c96dBBmCGalk7powY 2c80wyR9F4-uCEA/alt H78c9C9BBmCFBWSTUUJWGMyogxdMATHWEIGASCmillONLVsTBF6cDl6fYMgLGld6SShbu5Dly-H8aoqHaM8c9q8a6bWW ORD/IZ23TepThyTiCs6ArASTYddeHKHNihjPSp nlTaDQ5SSJUqscFJq+z2BDM8V6eQvsEsBonDEsHQCrvrYGh1KhEMrZC6DoZWFhl79LQ6ak8woxFgUhW1J5iBmmjEfDDZRCozqPPJJGK+J5iBmmgN6T1deyAZWud5TzADydCat6NpOE rOJBrwVsmsTwcDNdGa74NHlKpaaDJWqJdd7qWl9bhqlg42SWQ9m4E20Y4aBa5Oy/r3URMr6dzJgWYngtmqJlrwpzbijblpfalczGJstRkaBaEOoG2NwDOKocDXq+mG9oMrTv4pzhD6wz eSU2bwCdDJWTqbmJUPtleYkZplOt8TzijQk7xhTzCDCLzZtUmNOKprkz5sMKN6HekTetB3PD6mA9mBsWcevBiDZcSV3ywZwg6TSEClZ2frAnmlHN0t5W9qQz8KatBrwezA3rJpnN3Ylct dRBJSJFcwAiGFqcGFSUpHMZKrkiGeCeYEZ9YBIF2BPMIM7QS009wQzIDK3DvSeYwd4DrdmzJ5jBRgjiW7kmmNGuDKmpyCeYwRYR7cMGuYi01noWZwarBpOPHK0aUHLKRQqsi0jJ2xx8phPAjKalSVg2+e101cLHp6mVjQs01dAGCKBtqk11rlbEqiGmVQNti6+x9PWugz6BaOQdK6mElXzz7D51JMsGRunHX4(6xcrmDlbuLCB2ud9aZPQN112aBPlh7cKUL1yFX4VwIUloXL1egsZjj Pao0+xhLEOUozOWP1K8qzYY11E+iXUdZQS+irNEIAhdryxoK3ggURihbFCa7rck+qmPwllHoSxyiHixtfO8ofrHAIRojc+CJEquhg28UwCbPVvceh2jxpYhWf5CWUcjLHKM/m2vZY3QgZoIRO sqlejdesBRADVgDKFWNjdt8Oy50co05gJWZR7Bj0r28GVIIESLMCK2018JXwwyAHKyxTDchmidTh8SRj0XhMYmZTB8eURFB05wjKv14Wt3T3bl0j1rAgkZG5kNOEsRG3FK TugdesJkd5TWDLAJFRAGn5PmiU417JTGhTN8WDoxDww94SUUswh3tWmisTMMx6A7xYH4PZdmus0h17dwHNag5MV2Hj0Hj0kmKdC1KPpyj8+jRrwU9R2PsKZpH+W1GRAYOC cJY45ZWDTGJXA80U2JR8HTJDDYsjNGKZ65673w1NhEi44pW23tbwFSOLYNkixtisSsd8by7gZu4Nhvq5hTTPOC9FloyyMcZFFSTYYjl56And5WYAUJygWWY+R1FA4FYxplm3D WUjrURZAOMMOOFKC3EZBDImogZ7yRBDuTolSLPh2KimoChy9z2NC-UjrEQq-2ippiBqmH.uliMX4abS4Gm+-/TiqRbiEuTom/VSTKdhXCFqjiHjqujeBad27iBL7H+HGmY43cBoiKNUZVd Cdbr (DibXBoiRG) UlimMBdbyHM42-PYEPpiMR0H 2adWUlfdchoggfrm36 jEJTUW BDD/YAgGO/PUJrAdyR7bm4HbowxmoxXRZuliqd9i282PSBLR19H34 Llanyl HiBunLiS BBLSH18ixADPLCW2OnHBg3sUqp I juw7TEMgFstDxF1Ll0yH+SabKGWDys9QJUVGR4BBFYCKWD wWw.dkWZQJQ-1XPDldz2x3CZTgryatwKF307catAeVJAgoJY7MZuk1/mox SABAĞHUSSQUESUČXig0GHk1RpEMHQEvdATaT6iGgzNFKzJ5BZGjsczCMQSQWicc6iDphcBRIarEQ1US/E-V7EMzgxJTWB9XMH09uaSt3J0Wwa,bdRsxselLd09JXNDNa1PiDdUEy3rEy VDOxYUzJ3VB0I8USdmdlowOvBDB982Kam9WBuqCYaUySCoR13DIIeiSISwWxX0TVMbxRnNkpmPZBAW/1JhqHvpOa1nPgG6ppPZgbxhladXEnNa0n5DdMB7RS505Zm1bq3EINiFLnT Rzfd10Q29aD+aGiXJ93XRDb6JVpHfyJlpFSm0WkRrB18HQmrfUrE3S+Z5qRp0rUkU4oBAkyRlHBWl1z55qBiOUNHa/J5jBcCmtdbonmNHYLYnD7qlmtPhAYmp7qhlEYBo52hPMqF1PSnR 7ghk++EAJWQ06iXJUWIGmPSBwxigOsH+KZIFP1Ly9kcKifxloiOKGeDvX2aAW82xxiZ7PeW66iSCynwig5EMzxkCcN6zAAPZoCFMPZkCPgFJZ/Cn2cGODhzeiOb43aTyP3YEZEPh4t2c3 UHh6stzAwr/DeW2EeAdTy/DBJN5f6ivZzdJ+VDjeBLkxxMoKhopCsf7soTri0UghVSXhTyQ9hmUPv8vins4Loo8eTD+b8xaWWNEikrAS5muXoHFY4P4RUCM8Fz9wJUJv192KC2jE5Kdi 0hsn0Ep2VnJoqPWClAug6c9NIEGmpJSrWS6mr0/9U7zN7uw2udi3uZxDqA+h3/aRs0dhliKrtUgUXeBSZbmd07kLwUJlczqzvW2QCM7PneaBGR76PlV1CLBGMonyT3un+/LQbMZCXRi57 4qizQKT0MAXQuiXPvFmYydDYX4WkYtUmjAx9FWYwyiRP432y5CVIF+rqEuz/gWFOqhZ38vpD8m4/r6NdyzH3oHiWTkel4fV5WRRtUhEVxXcy13flkxByRCstw4eHq4q6nyY6AKaUBpH8RiZ AIGER.wVYc0Wi0Ug8XPDijVvuCuVycrJz8hnNchG67Bdic/2K5uq8KwmNUcmWbPsbVX1106oC1to3W6uq4Jg1ZrWMkPdHDY7QLJIECzvpY2RXbwslock+V4ggs1gKvLr1nmBFQeBIS?IEZN 2wXK6NpCG00LhLslkx9HVp1D8L4/FhzAUQAi0Qa5LpH-gUIGFCeXC4a7jHGutgH29Ws+hJQV7+yGrGw/VX2g2kXLB7GXCjvjzph7tTJ5GW1+51GEIKa3eSDC1036kvT8uxd+rL01LsvaZS EmN2Py/qqb1YEbd25J/XgdDSyLUoEdyzcHxC5mLEPtiW21mvWRGNiN1fGG1ZG54tl.8ezMi1SRF0c.jpO4mREMLQIO)g-ldFV6twkkYLuCeaWEwi1TSQDC2C7kmuBscntDixJ5jB9QnN NIcEMzo+leiBTzCjqR2SN+wJRjwOCb76kOCvr2cE3YRUI7cdGe7eY7+GkvvCZOE6Od8PUN0f6ixAKTnZozOCuWiG+q0Dk90RqCMOelhvaDkpzdwnD14c9czAsGDvNDbAkzxJzokQCi/dA OHFOK0/T48b7n9IYHyl24zkTzQRXmXQwKhjT4/JTDm8GIW6ZTgr5RDgphakFzBQVKvkVx/G9uC2kVYmYpBgHQuhwRaTfqopWleDwnkcXPqahORsqvp+1oG0jTzqQ9CVQUcXCJht9oqrx 7sJSx+K8wFNEb6SR+hudpDpN21g0iaYkmwKdVvkW+dKoD/gos3Li0izPLSZN+Ol0dgXjrhcliTdu8mNN5nrmS/jV6Jzl56tHDgdYp7b1FSWrecnyBamEm49w1WEfjkjxus2y6iQGgXBTVr5y+ P9WdWUI9zbguQWrDa1kOlwJ04o70Ll94hf606hGaFe+3XmgxthjhKoUVrvFliFA7SyrhgYbnrdJzWU9McU1RGd2eo+zuCR3C2GU4qqNpQ22cZXB+nCixlmDGPHmSqJATNJbADAXz8l Wde+Hed+U1omuzjhRl2U1rpY28SoRx2lKQyWZfK5lLOpEC1ectwvN1TVaDiCvyMGdy735kw4J1laGWbQbkwhk+qwhlATNl2zcqaY2ouHZymo56kgHpmAllo4vHqoSWgoujFyeBil9uqRjNF qyV227+Aae3FwFnaq9QonKsGgN4p8P3HBfm1wOuyyS6nn4E6AJ7i3gnD/4/gLm/Cm6C2B0080YIVhp/TJY45GzgrFMRBDAqK0xUr0e60hzRQHM0YHOGjCX7ujQvWEXPCoMC97ZHFhI H+W0PiQZbNbKL7vzn3Vx+QbGvd4Qa4T6aaZtz166wlvvMJikWp/r6aO7CrwbEdBJk2VOdnGp/aK8Ozte8nzS3L2lgatWQ0JirG9e1dBztu2TQv2BEZQCUoC4NhQCzi33I+BPTursIND5vR+hu a4p2m0qPFnrPZIY6X1OGw2DpDKLTIRdPxP73NKcNUFfw5is+r1W-6FpyloVjsxy6y0T3nsQnZZZE87jd/rLGksOV85ekl3xE29Gs01RoTYlgmJG9WCDCMO8donV2q/tc2A5v9xQ1J+\$PVeU mex-ZgP+m42ElymBlKCLB70RUbCs0byE4K4s23rj1+HkjHR-Sta6SR6LZuNlKtUhP6nKvNGIr+tqBlPgDHk9eq+0DHqmlqhHLlhNEHuKBtxBxAyqp3q2RAKHDYIS62pgSixK5creNTpWfblov
7Evbdh00mxNsTERDu8lwJe0RtWrKXKkRYcaGhcx/EurJZo9BWNNEaC9sJ0MrcaRATBNCZ1vRlbXEoBt27SLMRu7oTSooo3SooMv9iCQ5sDlLMoio5wBo5Jv6EZIJ5rsNl9NS636F129QU r8zPYJ3sqBIQxY33l2kxryLMhfzwUYdgVIJUyMr/apaXZE0k0Aozl7kzAsKvqVQQLÉ7CQTI0O80N2Z55by/m4q0jR+n32frtiXKqHJVvLeLauppW8N4rQAg2+/uFQiHSi3tonDiSHaOeF7tab0h9L eEsdlus/TadGa_trewarRob5Trzw8droppkcGVEn1Eujn74KFU_yen1 + 7:A7DYXCIDMYSSR-Ve8RDFBrdHK/XkxoXKhiSE+A1ZnT_IDSn5De6SSLOBRCF[g]GGVg.Keochlyr15940zaBba a7g/cibidei.12Gp3aBex04deNPceMsp8kcTBB-m41BokggGVJ9180BpNRDXRCrdBOID 14|gGTKxp9wCz84y/3898Mg.JBBQUBSOXVC9gQz2glUVJABryGRszAlwg51aW bdE8xgp58W4M 013GAdhEY89gQzGAqhZdd9W1w6BCm37AlmV82QcsueYEbpgJRb8gQz4sCkcLsnETIIM6QlueYW15N7qlu+HVdLSK9E44+sc0Mtra8bd6gmWq1OndLeehcRqVa1zeRaw/7DbmuLvxv ec3YaA3-t8hpP2hb2gyppUK1mY1qWiBSueUVnKT-p3iarHiJ3NBmaM2m094MrdIDJ>Rnik13ibImi0U3)MDrwdxwsYfWhiuTmia30W4Y9GgwTi503iJ3FJNpAbihYierUF5r1Jlddd2hmqiN SykJaf1s2bik53H5NEEE2a0hHJIKCG5b3x29YgbhB14C555-2DW0qemx6Vshley5VWsAgKTDT W\$Mhn1YgVMK41mMF7GC99JbqqyP2w4rHifyNy8xmWYWxxSt1WMF6rGA9VAK1KF2a2HCYMMAK1mMF7GC9JbqeqgP2w4rHifyNW8xmWYWxxSt1WH6rGD9SYewXheW



HI-Link

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:

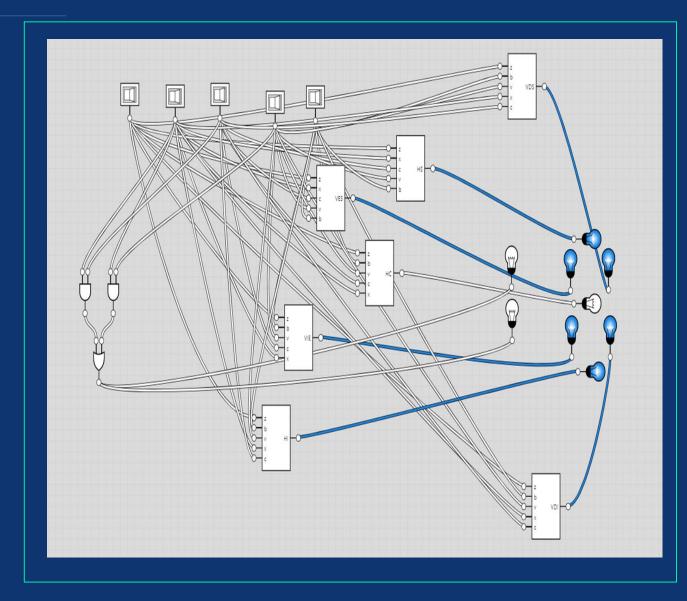
BINARIOS	DECIMAL	PALAVRA
00000	0	
00001	1	А
00010	2	В
00011	3	С
00100	4	D
00101	5	E
00110	6	F
00111	7	G
01000	8	Н
01001	9	_
01010	10	J
01011	11	K
01100	12	L
01101	13	М
01110	14	N
01111	15	0

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

 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.

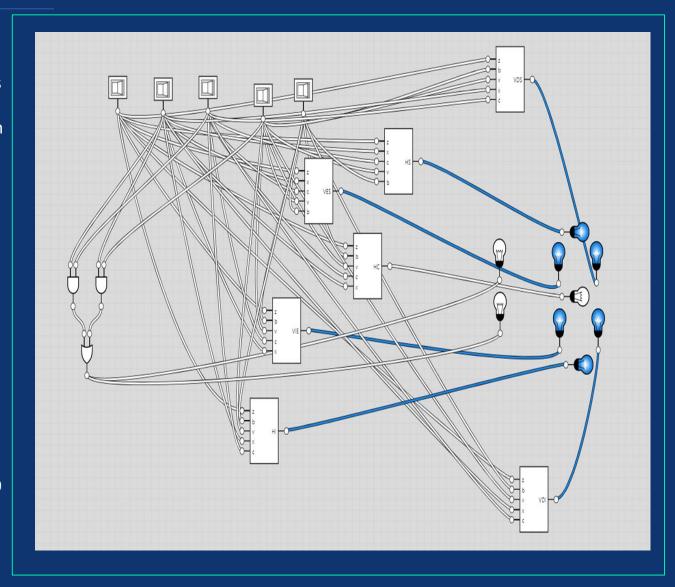
Ø 1 1	7	G
01000	8	Н
01001	9	I
01010	10	J
01011	11	K
01100	12	L
01101	13	М
01110	14	И
01111	15	0

 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)



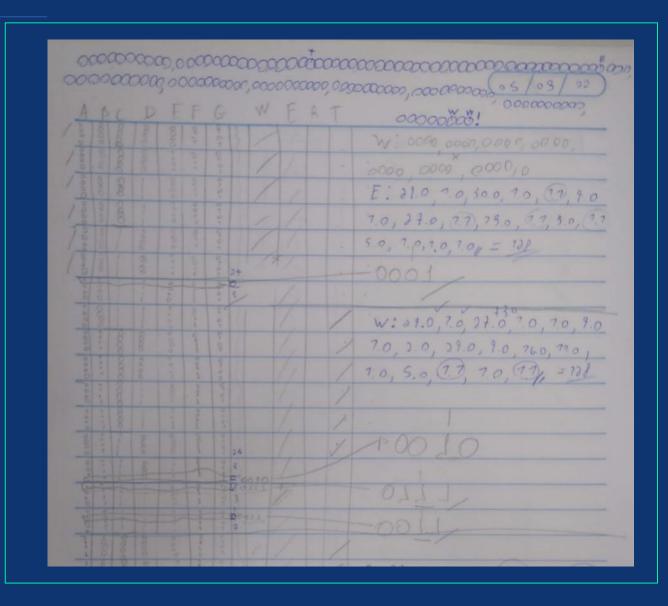


- Para o decoder, é simples a lógica que devemos usar. Mas antes vamos definir algumas coisas, primeiro: O usuário deve desenhar o dígito nos displays e isso deve ser traduzido para um número binário de 4 dígitos, entretanto, faremos com que desenhos que não existem darão o valor 0000. Segundo: apesar 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 VVVVVFV. 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).



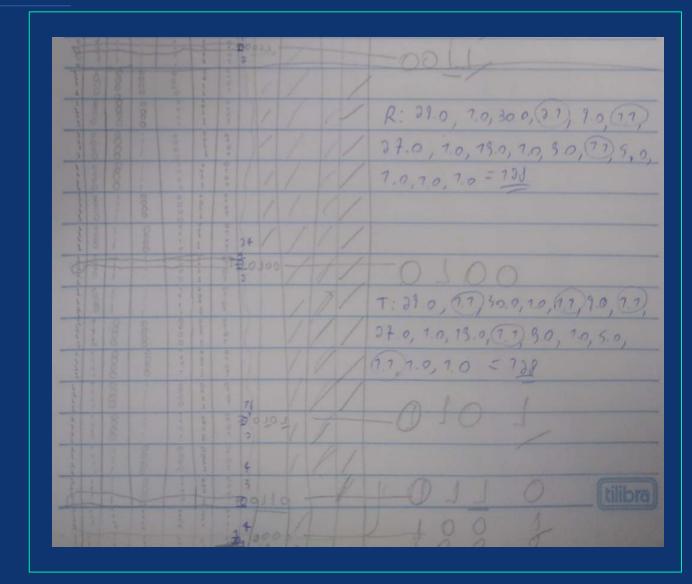


 Lembra que precisávamos da tabela verdade por escrito para colar no site? Bom, aqui estamos montando ela, já que cada uma tem no mínimo 119 zeros, temos que descobrir em que posição os outros valores ficam:





• Continuando:

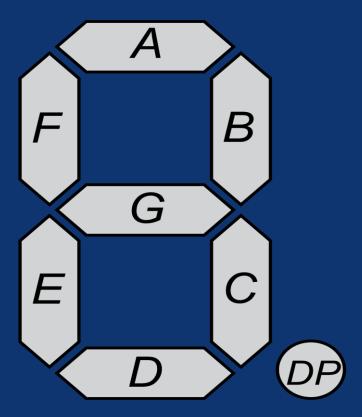




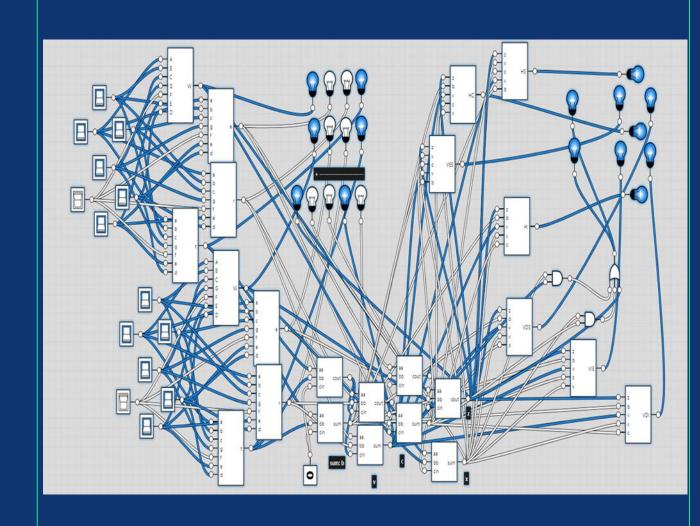
É bem simples de visualizar o que está acontecendo aqui, a primeira linha cheia de zeros representa apenas onde fica a posição de cada número, sendo que o zero é representado por um 'O'. Aqui por exemplo, podemos ver que já que cada saída (W,E,R e T) representa um algarismo do valor representado pelo 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.

```
ABCDEFG --- WER
                 126 ou N 127 na tabela e 128 na linha
                 24 OU N 25 e 26 na linha
                55 ou N 56 e 57 na linha
                61 ou N 62 e 63 na linha
                 89 ou N 90 e 91 na linha
                109 ou N 110 e 111 na linha
                111 ou N 112 e 113 na linha
                56 ou N 57 e 58 na linha
                127 ou N 128 e 129 na linha
     1111111
                 125 ou N 126 e 127 na linha
              as colunas em pe representam os numeros formados por cada configuração dos displays
```

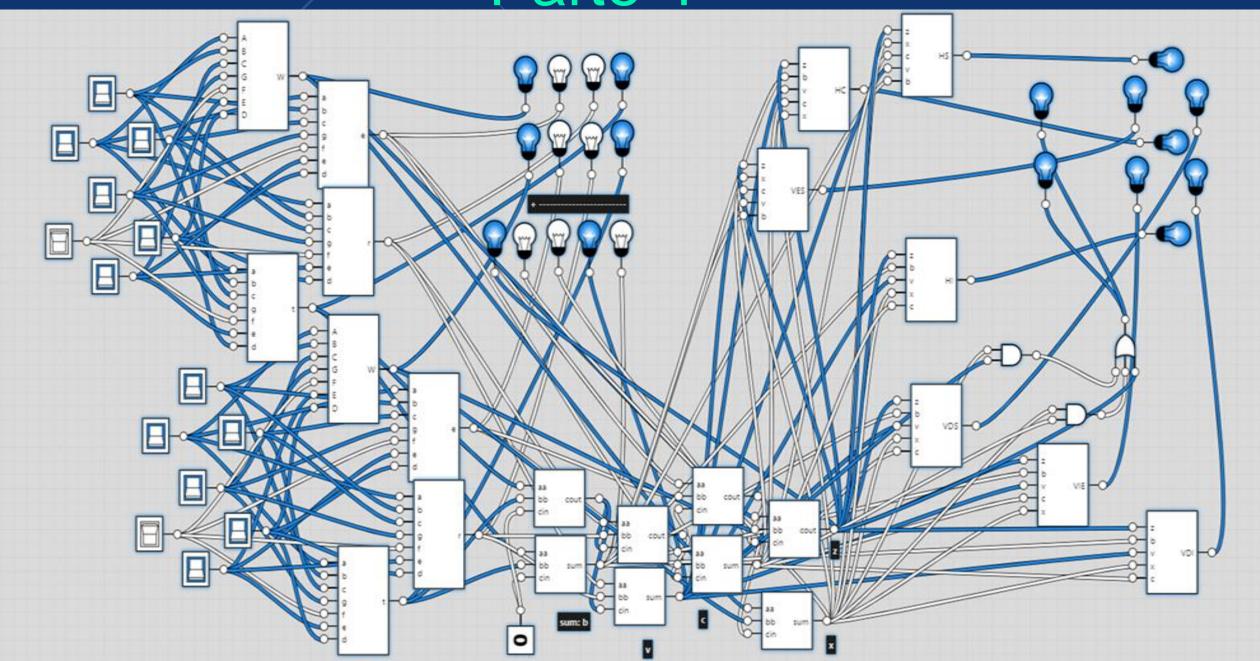
- Para o dígito W (milhar), a expressão lógica será a Λ b Λ
 c Λ d Λ e Λ g
- Para o E: (a ∧ b ∧ ¬c ∧ d ∧ e ∧ g) ∨ (a ∧ ¬b ∧ c ∧ d ∧ ¬e ∧ ¬f ∧ g) ∨ (¬a ∧ b ∧ c ∧ d ∧ ¬e ∧ ¬f ∧ ¬g)
- Para o R: (a ∧ b ∧ ¬c ∧ d ∧ e ∧ f ∧ g) ∨ (¬a ∧ b ∧ c ∧ d ∧ e ∧ f ∧ g) ∨ (¬a ∧ b ∧ c ∧ d ∧ ¬e ∧ ¬f ∧ ¬g) ∨ (¬a ∧ b ∧ c ∧ ¬d ∧ e ∧ f ∧ g)
- Para o T: (a ∧ b ∧ d ∧ e ∧ ¬f ∧ g) ∨ (¬a ∧ c ∧ d ∧ ¬e ∧ ¬f ∧ g)
 ∧ ¬g) ∨ (b ∧ c ∧ d ∧ e ∧ ¬f ∧ 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/compartil har.txt/file



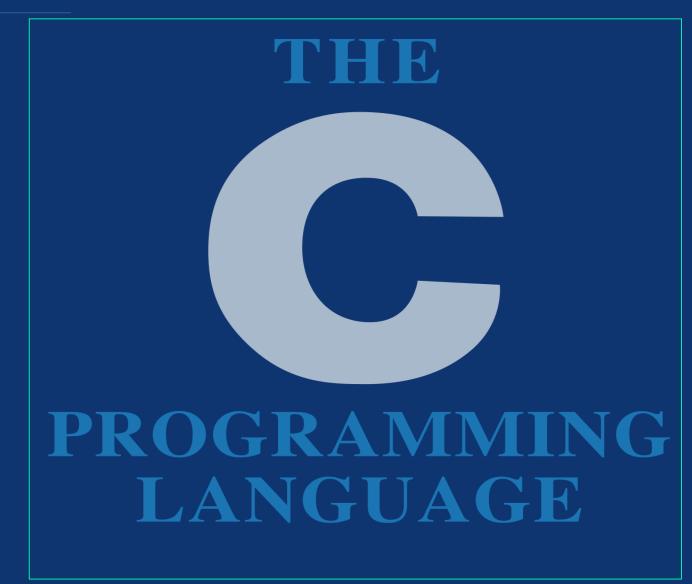
 Esse projeto pode ser dividido em 4 partes, na esquerda está o decoder, na direita o encoder junto ao overflow. Na parte inferior central vemos a soma, e na superior central vemos a demonstração, onde vemos 9 + 9 = 18 corretamente em binário. Resultado:







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





 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 carrys 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();
}</pre>
```

 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();
}</pre>
```

```
//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));
```

 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

- 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/h scas9r6wc258bs/Projeto1.exe/file

