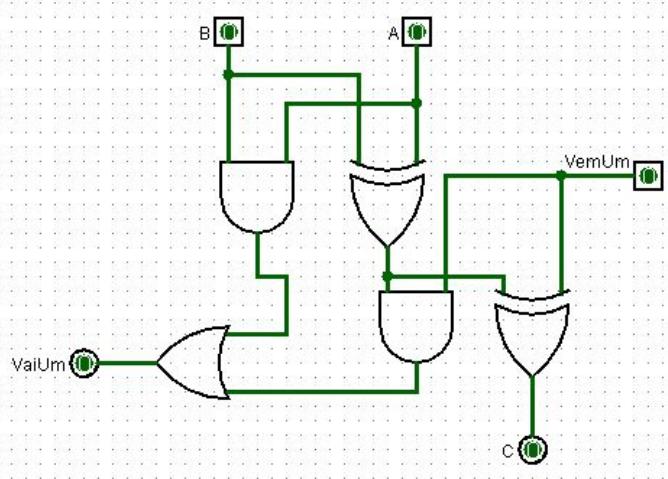


Projeto de ULA

Anderson G. S. P. Fantin	9016981
Larissa Fabião da Fonseca	11208367
Lucas Imamura	11208221
Maria Fernanda Basso	11208197
Vinícius Bispo	10875965

Questão 1:

Para implementar um somador de 8 bits é necessário vários somadores de um único bit associados, com um circuito de acordo com a imagem a seguir.



Esse circuito realiza a soma dos bits dos pinos A, B e VemUm, e seu resultado da mesma ordem posicional de A e B, no caso 2^0 , no pino C. Dessa forma caso a soma resulte em 1 bit da ordem seguinte, ele será armazenado no pino VaiUm, que será o pino VemUm do próximo somador, somente o VemUm do primeiro somador é sempre zero.

O circuito é feito de acordo com as seguintes tabelas verdadeas e expressões lógicas.

Tabela para o C:

A	B	VemUm	C
0	0	0	0
1	0	0	1
0	1	0	1
0	0	1	1
1	1	0	0
1	0	1	0
0	1	1	0
1	1	1	1

Tabela para o VaiUm:

A	B	VemUm	VaiUm
0	0	0	0
1	0	0	0
0	1	0	0
0	0	1	0
1	1	0	1
1	0	1	1
0	1	1	1
1	1	1	1

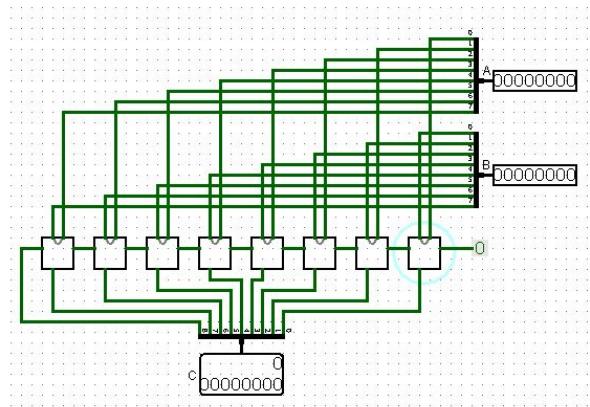
Expressão para o C:

$$C = (\overline{A}B + A\overline{B})VemUm + (\overline{A}B + A\overline{B})\overline{VemUm} = (A + \overline{B})(\overline{A} + B)VemUm + (\overline{A}B + A\overline{B})\overline{VemUm}$$

Expressão para o VaiUm:

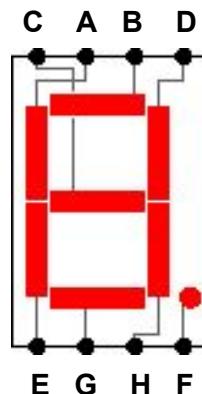
$$VaiUm = AB\overline{emUm} + A\overline{B}emUm + \overline{A}BemUm$$

Dessa maneira, para implementar o somador de 8 bits como na imagem a seguir, ligam-se os 2 bits das mesmas ordens posicionais dos números binários A e B em um somador de 1 bit (circulado de azul). Sendo assim, como A e B têm 8 bits há 8 somadores associados, e os resultados deles serão os bits do resultado C (em ordem positional), o qual tem 1 bit a mais que é o VaiUm do último somador de 1 bit.



Questão 2:

O primeiro passo para solucionar a segunda questão foi compreender as entradas do display de 7 segmentos. Desta forma, obtemos o seguinte esquema:



Após compreendermos como controlar cada um dos LEDs do display e nomear as portas correspondentes a cada um dos Led's, determinamos quais destes deveriam estar ligados de acordo com cada número a ser exibido. Para isso, criamos a tabela verdade a seguir, onde 1 representa que o led deve estar ligado para representar o número e 0 que deve estar desligado.

Hexa	a0	a1	a2	a3	A	B	C	D	E	F	G	H
0	0	0	0	0	1	1	0	1	1	0	1	1
1	0	0	0	1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	1	1	1	1	0	1	0
3	0	0	1	1	0	1	1	1	0	0	1	1
4	0	1	0	0	1	0	1	1	0	0	0	1
5	0	1	0	1	1	1	1	0	0	0	1	1
6	0	1	1	0	1	1	1	0	1	0	1	1
7	0	1	1	1	0	1	0	1	0	0	0	1
8	1	0	0	0	1	1	1	1	1	0	1	1
9	1	0	0	1	1	1	1	1	0	0	1	1
a	1	0	1	0	1	1	1	1	1	0	0	1
b	1	0	1	1	1	0	1	0	1	0	1	1
c	1	1	0	0	1	1	0	0	1	0	1	0
d	1	1	0	1	0	0	1	1	1	0	1	1
e	1	1	1	0	1	1	1	0	1	0	1	0
f	1	1	1	1	1	1	1	0	1	0	0	0

Após completar a tabela verdade, extraímos as expressões lógicas correspondentes a cada uma das colunas da direita (A, B, C, D, E, F, G, H) para assim poder criar o circuito lógico adequado para cada uma das portas. Para encontrar as expressões lógicas reduzidas, utilizamos mapas de Karnaugh:

Coluna A		AB				Coluna B		AB				Coluna C		AB			
		00	01	11	10			00	01	11	10			00	01	11	10
CD	00	1	1	1	1	CD	00	1	0	1	1	CD	00	0	1	0	1
	01	0	1	0	1		01	0	1	0	1		01	0	1	1	1
	11	0	0	1	1		11	1	1	1	0		11	1	0	1	1
	10	0	1	1	1		10	1	1	1	1		10	1	1	1	1

$$\text{Coluna A} = A \neg B + \neg C \neg D + B \neg D + AC + \neg AB \neg C$$

$$\text{Coluna B} = \neg AC + BC + A \neg D + \neg B \neg D + \neg ABD + A \neg B \neg C$$

$$\text{Coluna C} = A \neg B + C \neg D + AD + \neg BC + \neg AB \neg C$$

Coluna D		AB				Coluna E		AB				Coluna G		AB			
		00	01	11	10			00	01	11	10			00	01	11	10
CD	00	1	1	0	1	CD	00	1	0	1	1	CD	00	1	0	1	1
	01	1	0	1	1		01	0	0	1	0		01	0	1	1	1
	11	1	1	0	0		11	0	0	1	1		11	1	0	0	1
	10	1	0	0	1		10	1	1	1	1		10	1	1	1	0

$$\text{Coluna D} = \neg B \neg C + \neg B \neg D + \neg A \neg C \neg D + \neg ACD + A \neg CD$$

$$\text{Coluna E} = AB + C \neg D + AC + A \neg D + \neg B \neg D$$

$$\text{Coluna G} = A \neg C + BC \neg D + \neg BCD + \neg A \neg B \neg D + B \neg CD$$

Coluna H		AB			
		00	01	11	10
CD	00	1	1	0	1
	01	1	1	1	1
	11	1	1	0	1
	10	0	1	0	1

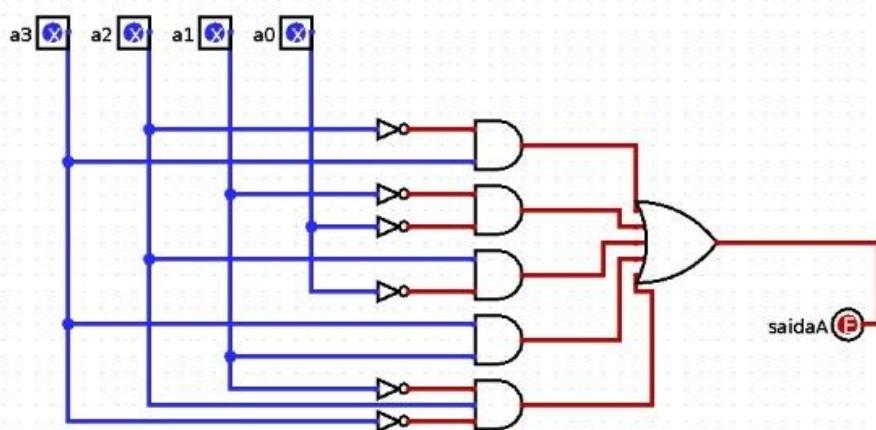
$$\text{Coluna H} = \neg AB + A \neg B + \neg CD + \neg A \neg C + \neg AD$$

Observação:

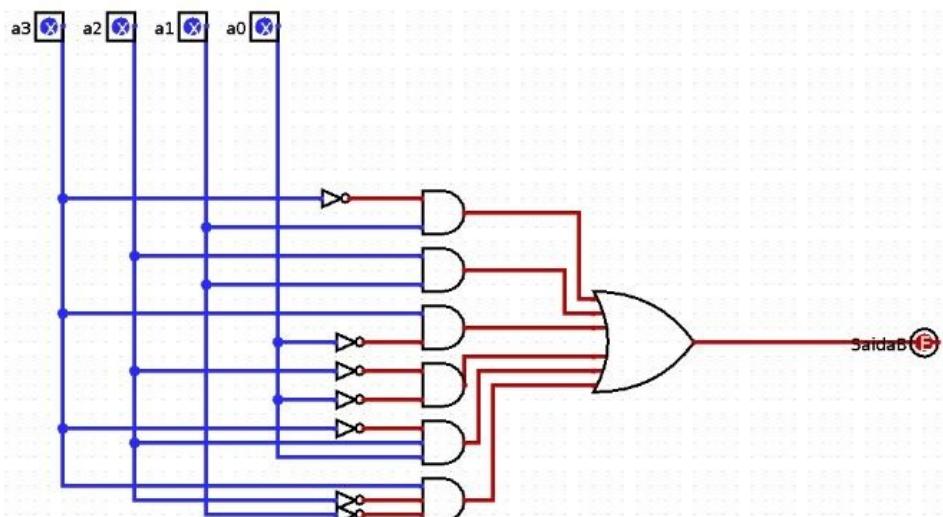
- consideramos a3 = A, a2 = B, a1 = C e a0 = D para facilitar o entendimento das $\neg\neg$ -expressões lógicas
- Além disso, não construímos o mapa para a coluna F já que a coluna é preenchida completamente por 0 uma vez que é o led dedicado para a representação do ponto flutuante, o qual nunca utilizamos.

Com as devidas expressões lógicas, construímos os circuitos utilizando o LogiSim de forma a trabalhar com cada Coluna como um microcircuito.

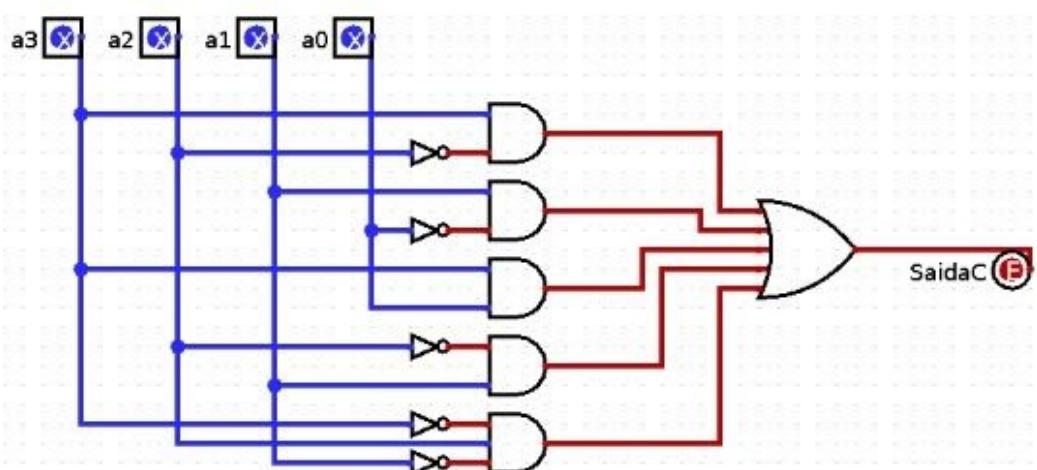
Coluna A:



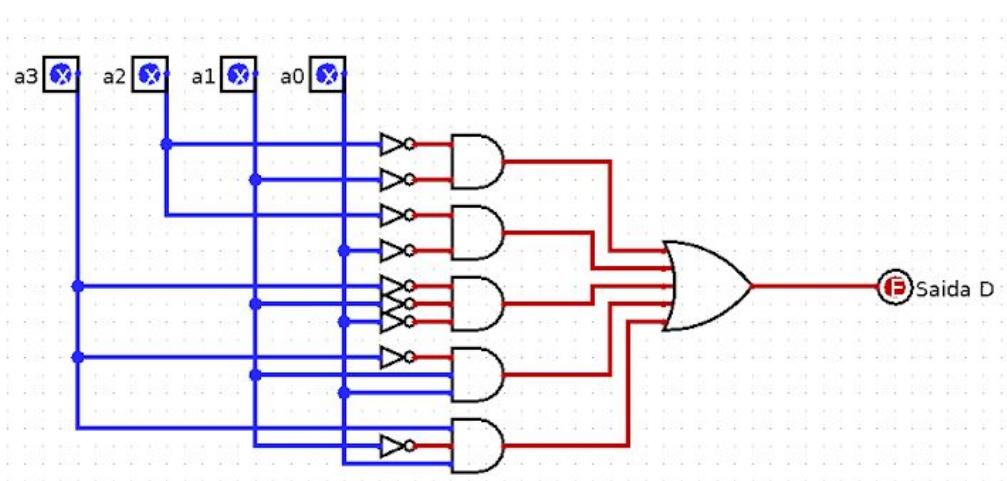
Coluna B:



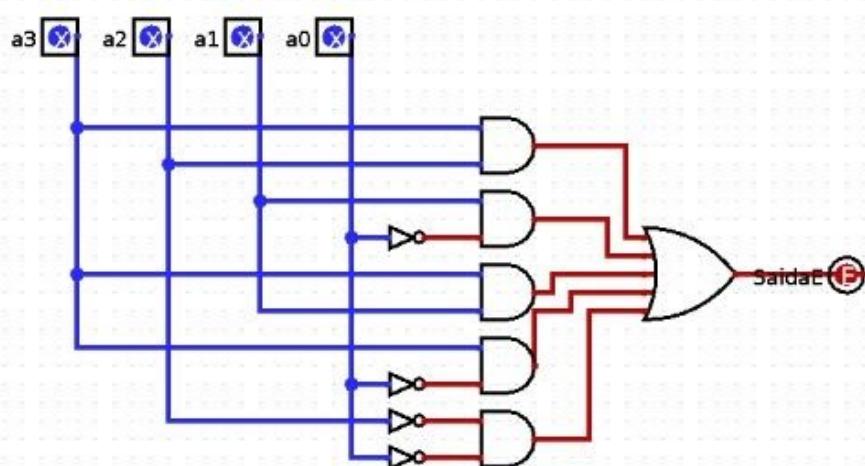
Coluna C:



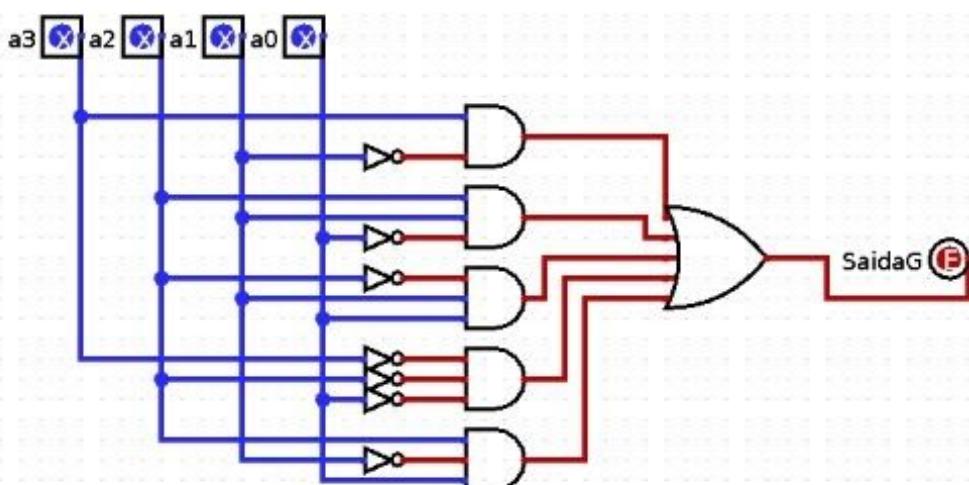
Coluna D:



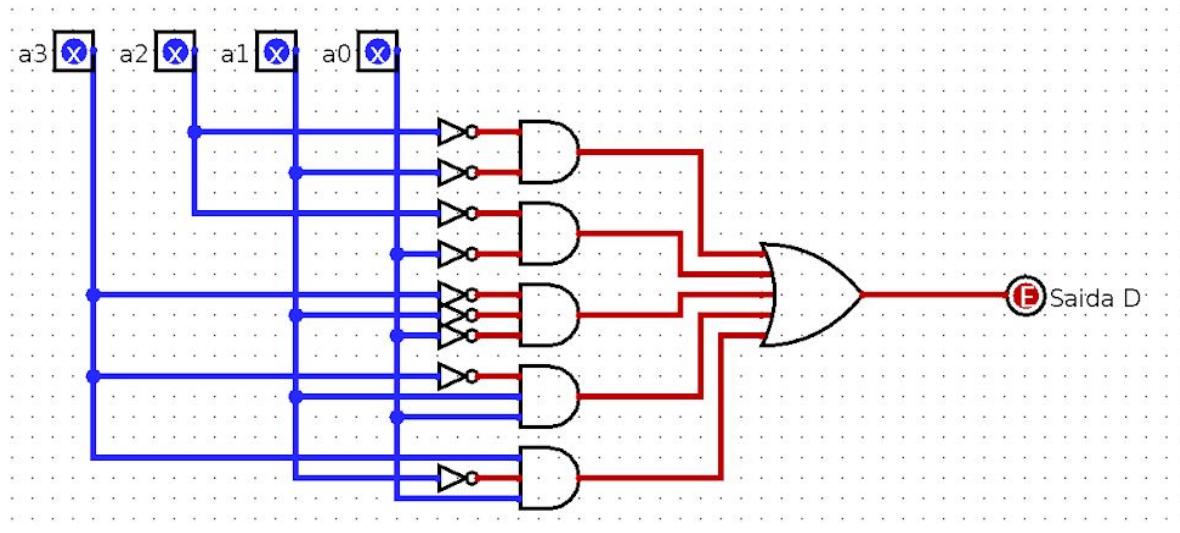
Coluna E:



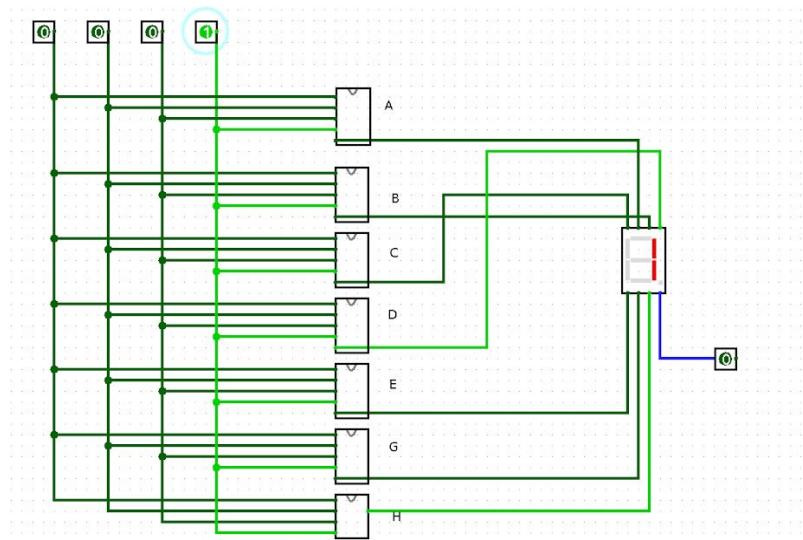
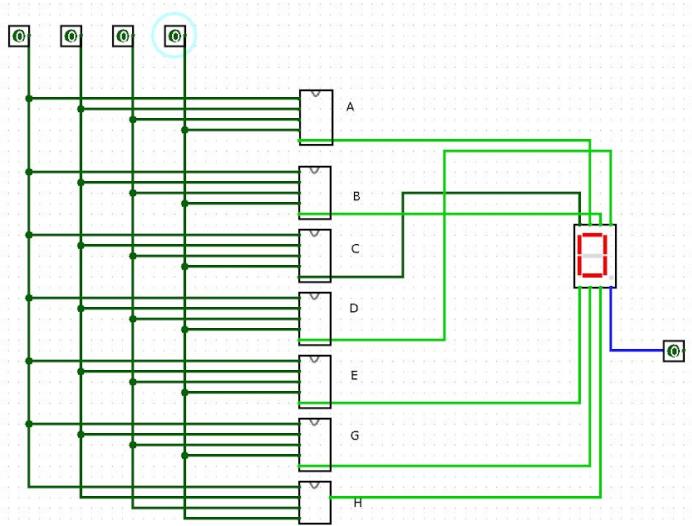
Coluna G:

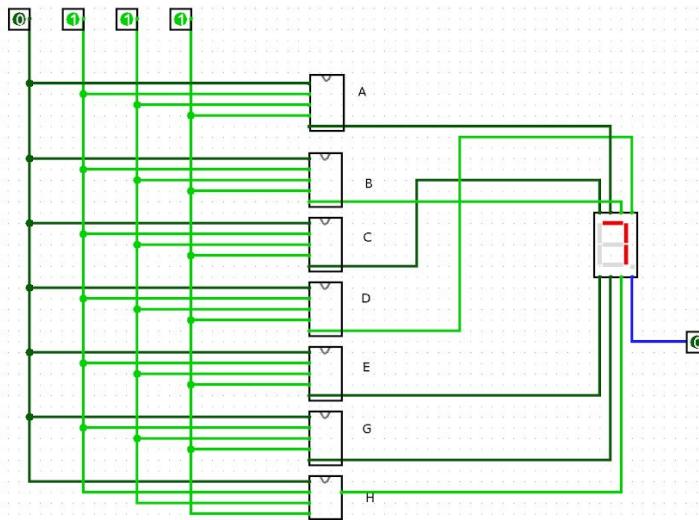
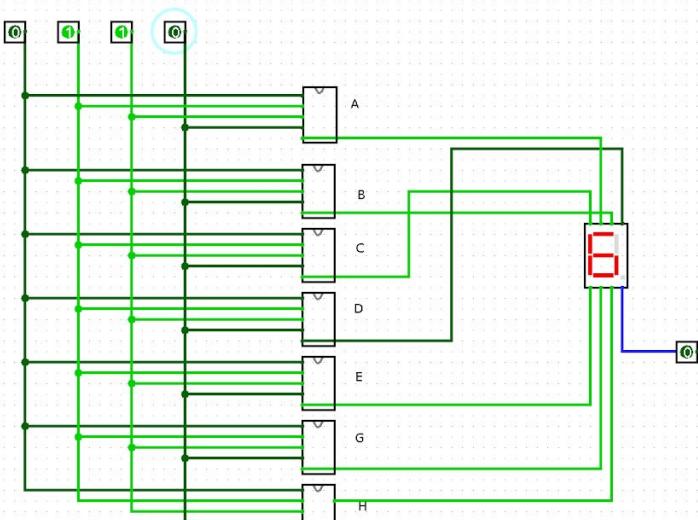
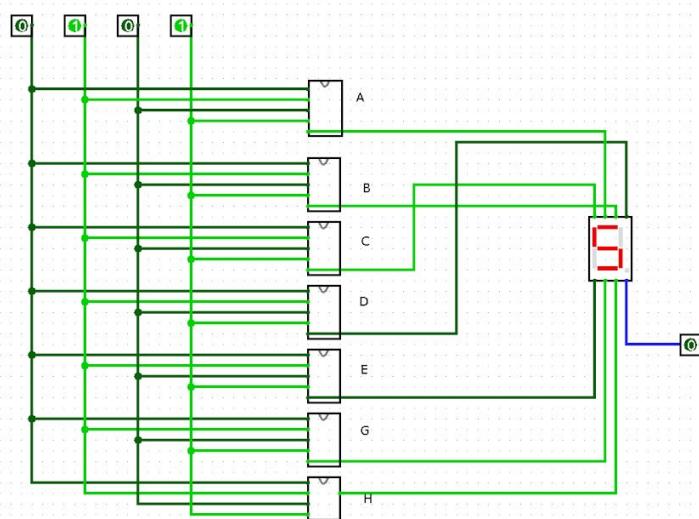
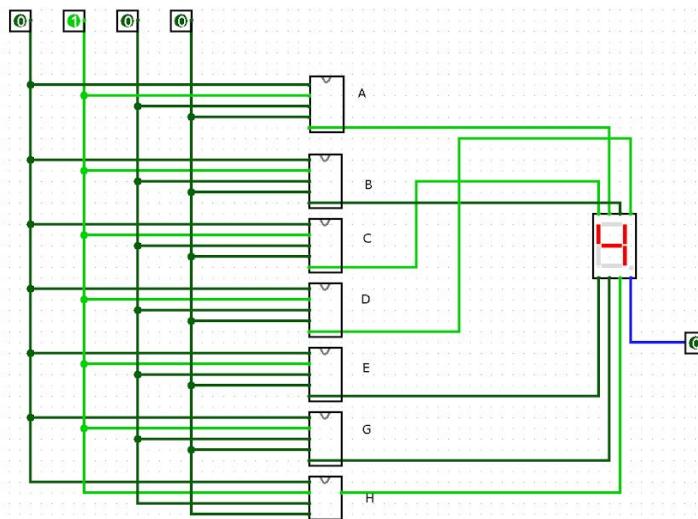
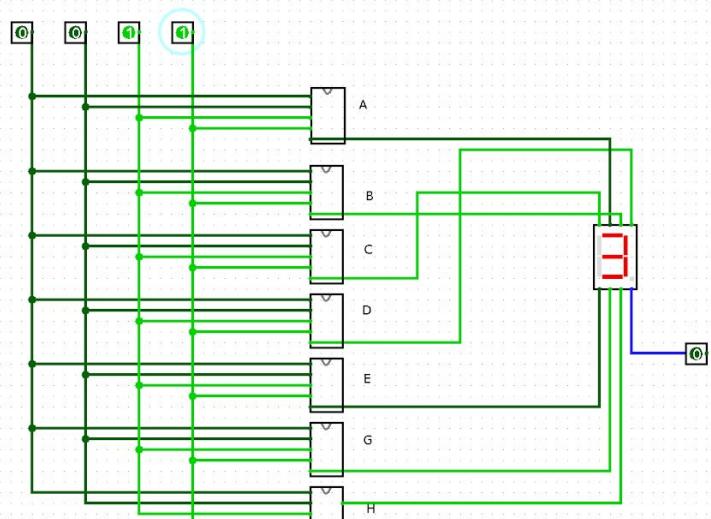
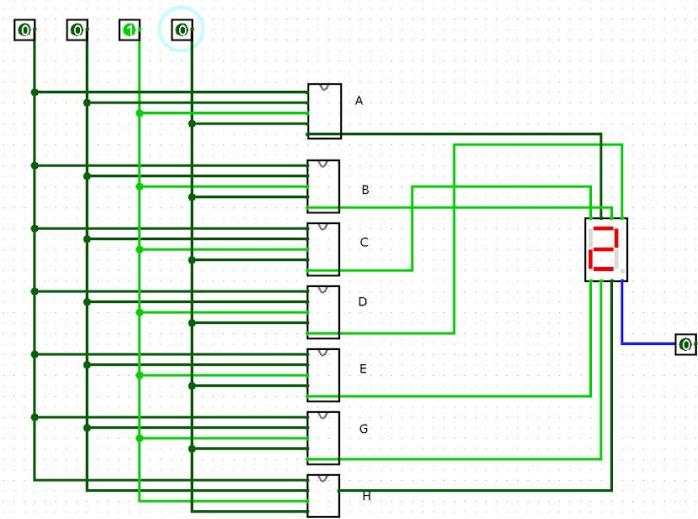


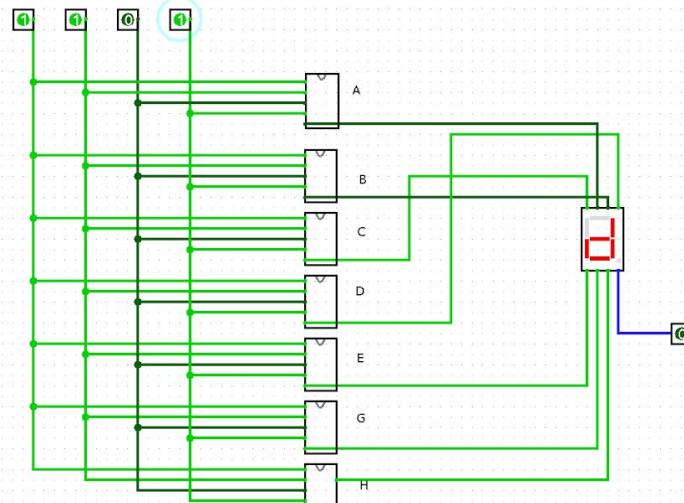
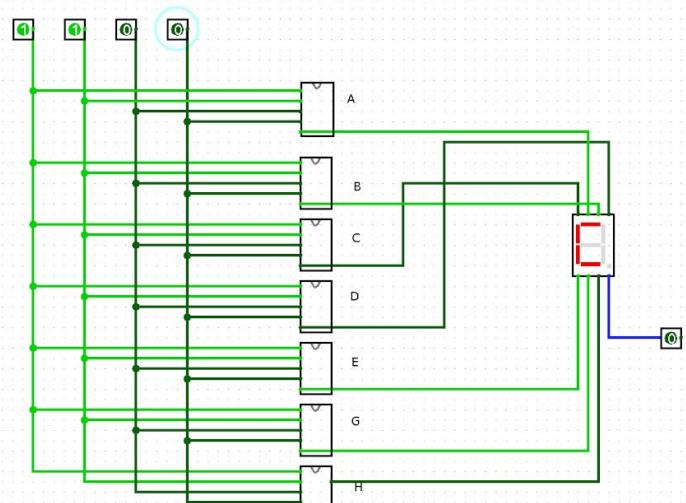
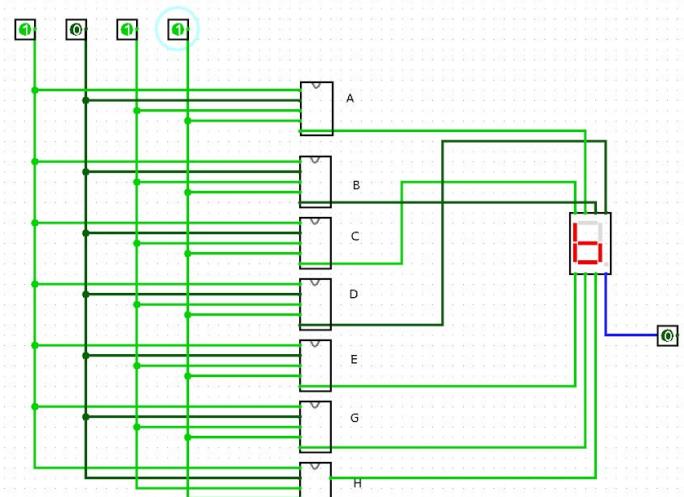
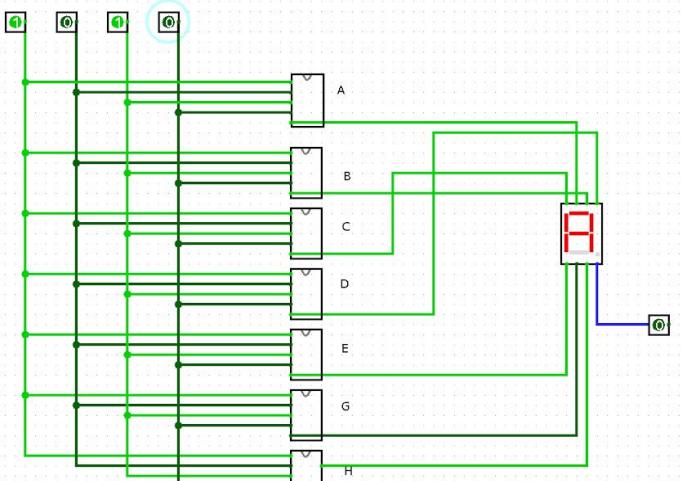
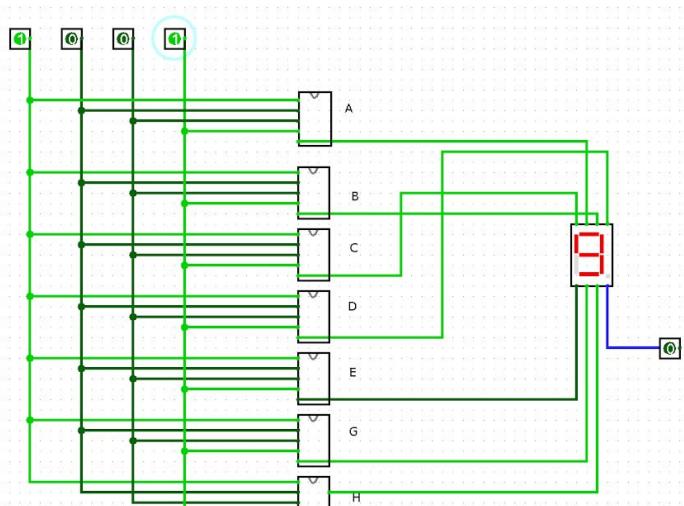
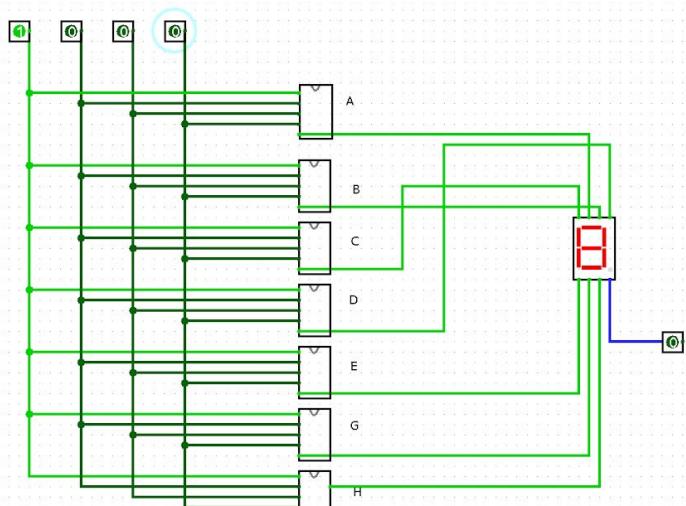
Coluna H:

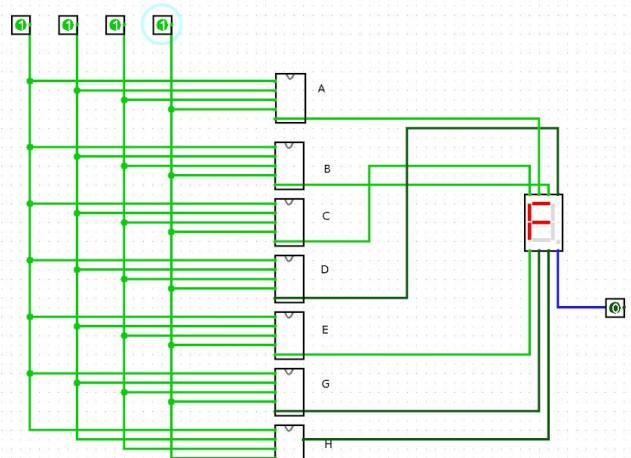
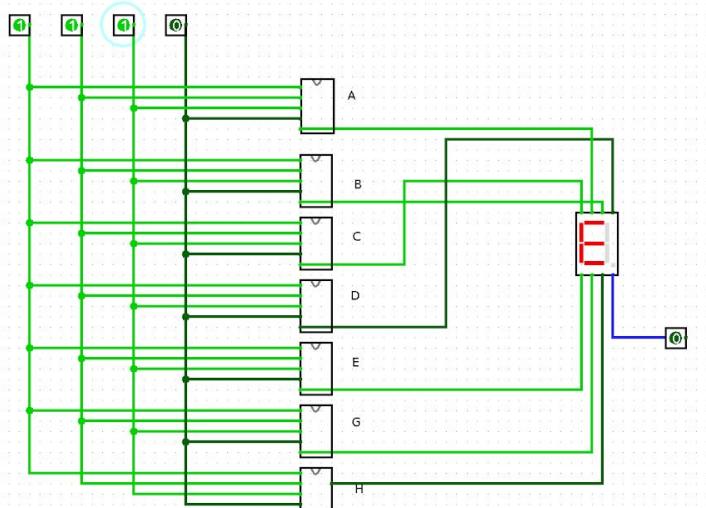


Juntando os microcircuitos, construímos nosso circuito principal onde conectamos as saídas aos pinos correspondentes no display de 7 segmentos, obtendo o seguinte resultado:

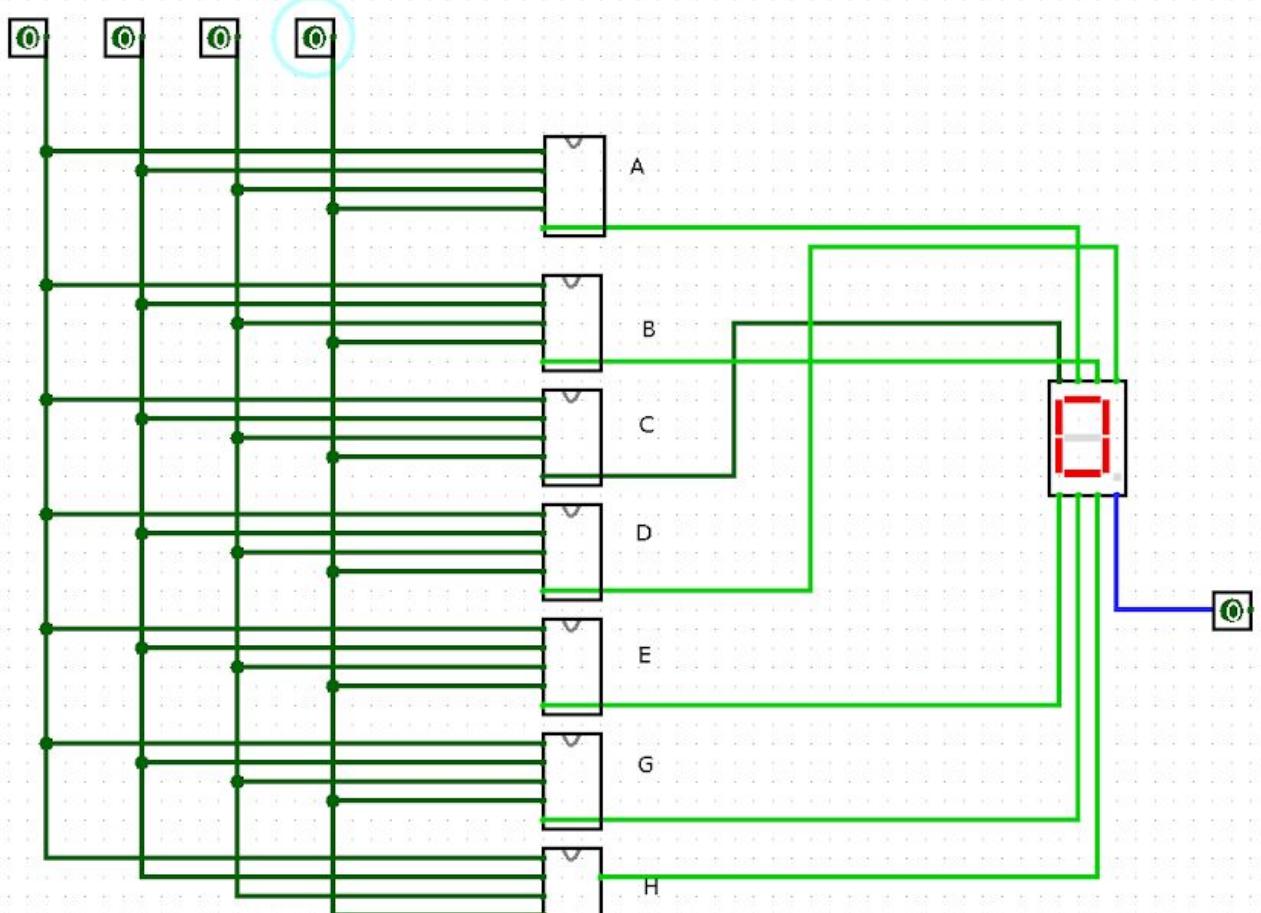






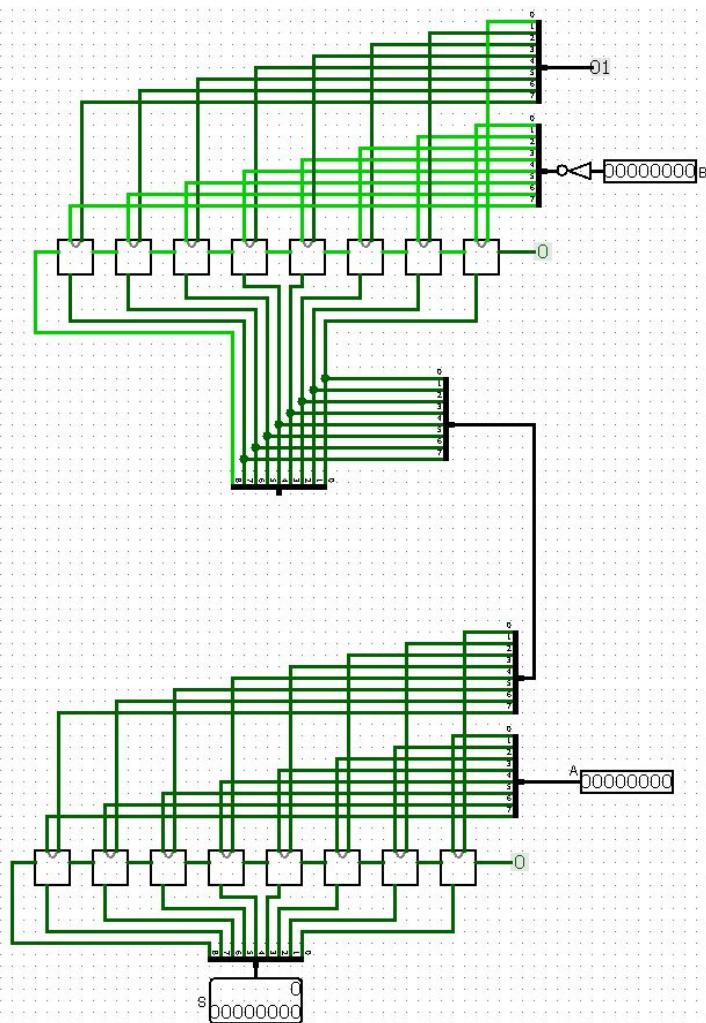
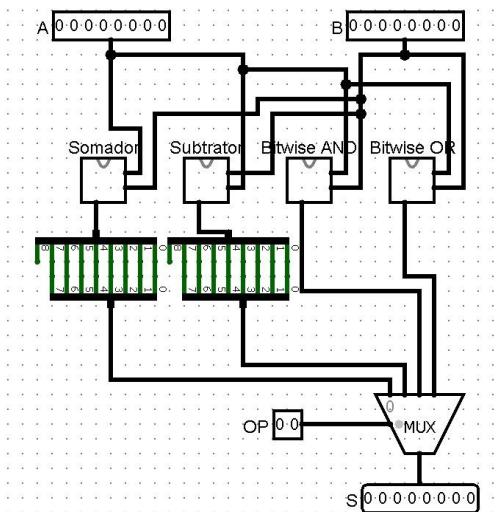


Captura ampliada do circuito final:



Questão 3:

Para elaborar uma Unidade Lógica Aritmética (ULA), como a representada na imagem a seguir, é necessário somador, subtrator, bitwise AND e bitwise OR, além de um multiplexador (MUX).



O subtrator é semelhante ao somador da ULA, que foi feito analogamente ao da questão 1, pois ele efetua duas somas em sequência. A primeira (parte superior da imagem à esquerda) é para realizar a soma do complemento de 1 do número B (\bar{B}) com 01 bit. Para isso ela utiliza uma porta lógica NOT em B que “inverte” seus bits, depois a soma é feita do mesmo modo que o do somador de 8 bits.

Após isso, a segunda adição (parte inferior da imagem à esquerda) soma o número A e o resultado da primeira operação, que é obtido conectando seus bits em ordem (sem seu nono bit) com esse segundo somador de 8 bits. Logo, a solução final de $A - B$ se encontra em S.

Para consulta, a tabela verdade de NOT B ou \bar{B} :

B	NOT B
1	0
0	1

O bitwise AND retratado na imagem abaixo é composto por uma porta lógica AND para cada dois bits de mesma ordem posicional dos números A e B, assim os resultados dessas portas lógicas compõe o número C com suas respectivas ordens posicionais.

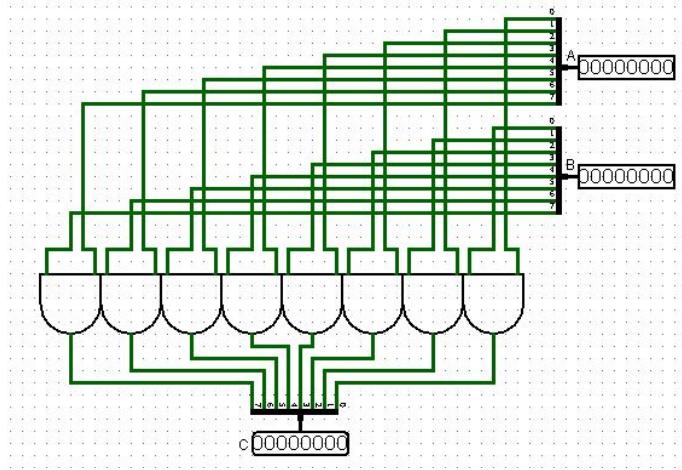
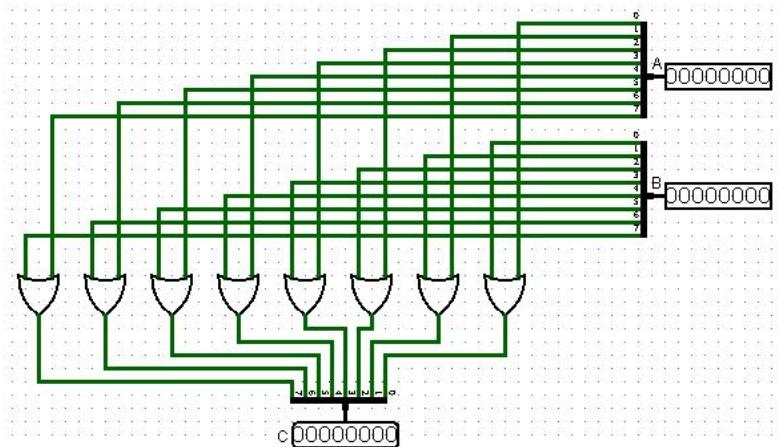


Tabela verdade de A AND B ou AB :

A	B	A AND B
0	0	0
1	0	0
0	1	0
1	1	1

O circuito bitwise OR retratado na imagem abaixo é composto por uma porta lógica OR para cada dois bits de mesma ordem posicional dos números A e B, portanto os resultados dessas portas lógicas compõe o número C com suas respectivas ordens posicionais.



Para orientação, a tabela verdade de A OR B ou $A + B$:

A	B	A OR B
0	0	0
1	0	1
0	1	1
1	1	1

Por fim, a operação com solução em S na ULA é decidida através do multiplexador (MUX) no pino “Operação escolhida” na imagem a seguir ou “OP” na imagem da ULA (primeira imagem da questão 3). Se esse pino for igual a 00, será o resultado da soma, se for igual a 01, será o resultado da subtração, se for igual a 10, será o resultado do bitwise AND, se for igual a 11, será o resultado do bitwise OR. Além disso, as conexões do MUX com os resultados das operações na ULA foi feito de acordo com a imagem abaixo.

