

1-) Qual o valor da sequência de 8 bits: $(11001101)_2$

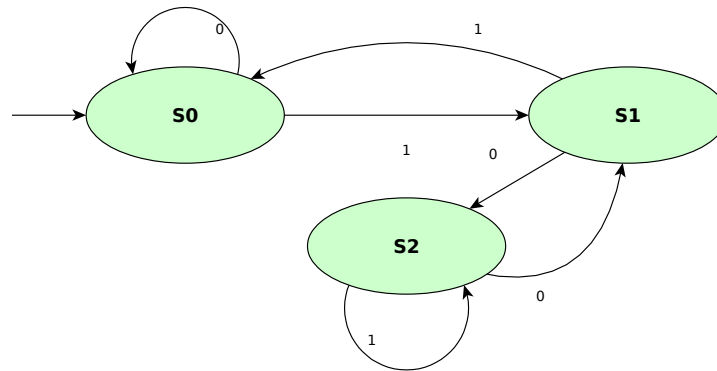
a-) caso represente um inteiro sem sinal **(0,2pt)**;

205

b-) caso represente um inteiro com sinal em complemento de 2 **(0,3pt)**.

-51

2-) Implemente o autômato abaixo:



a-) identifique (liste) os estados, alfabeto de entrada e estado inicial **(0,5pt)**;

$Q = \{S0, S1, S2\}$

Alfabeto = $\{0, 1\}$

$q_0 = S0$

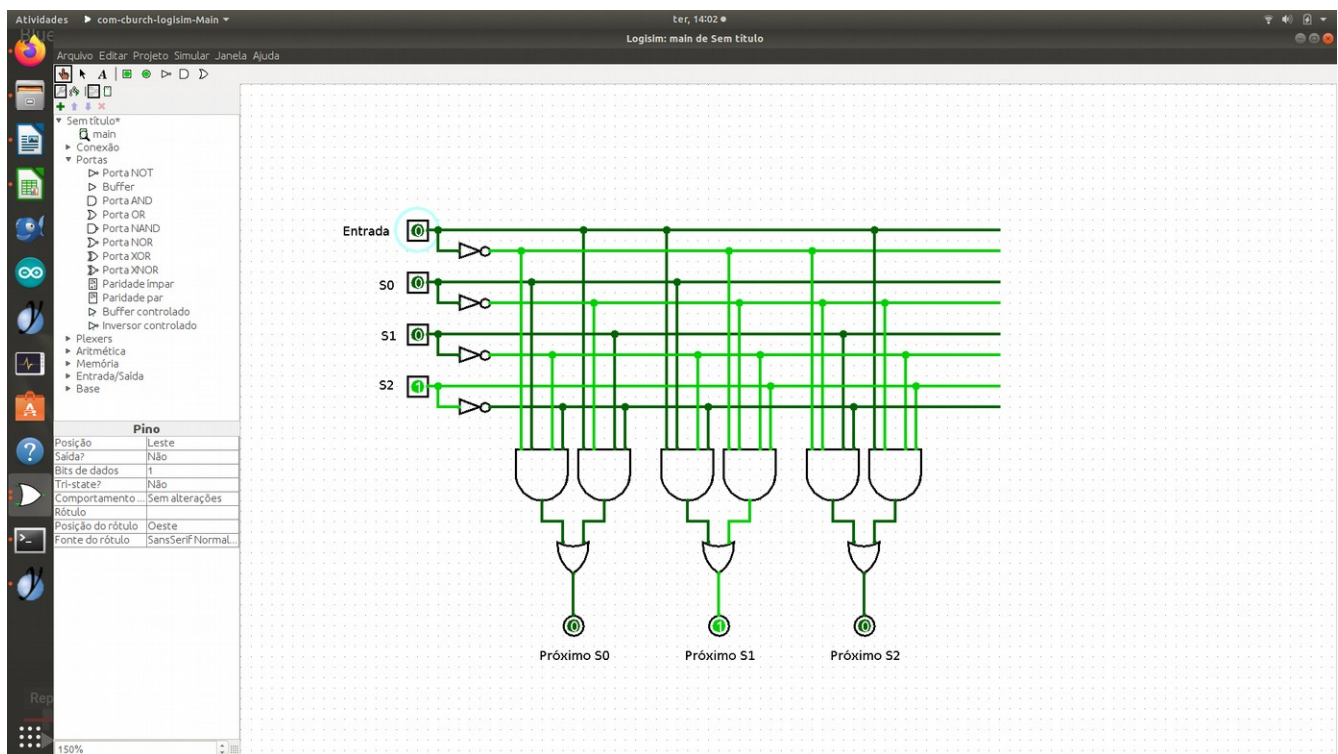
b-) faça e apresente a tabela de transição de estados **(1,5pt)**;

E	Estado Atual	Proximo Estado
0	S0	S0
1	S0	S1
0	S1	S2
1	S1	S0
0	S2	S1
1	S2	S2

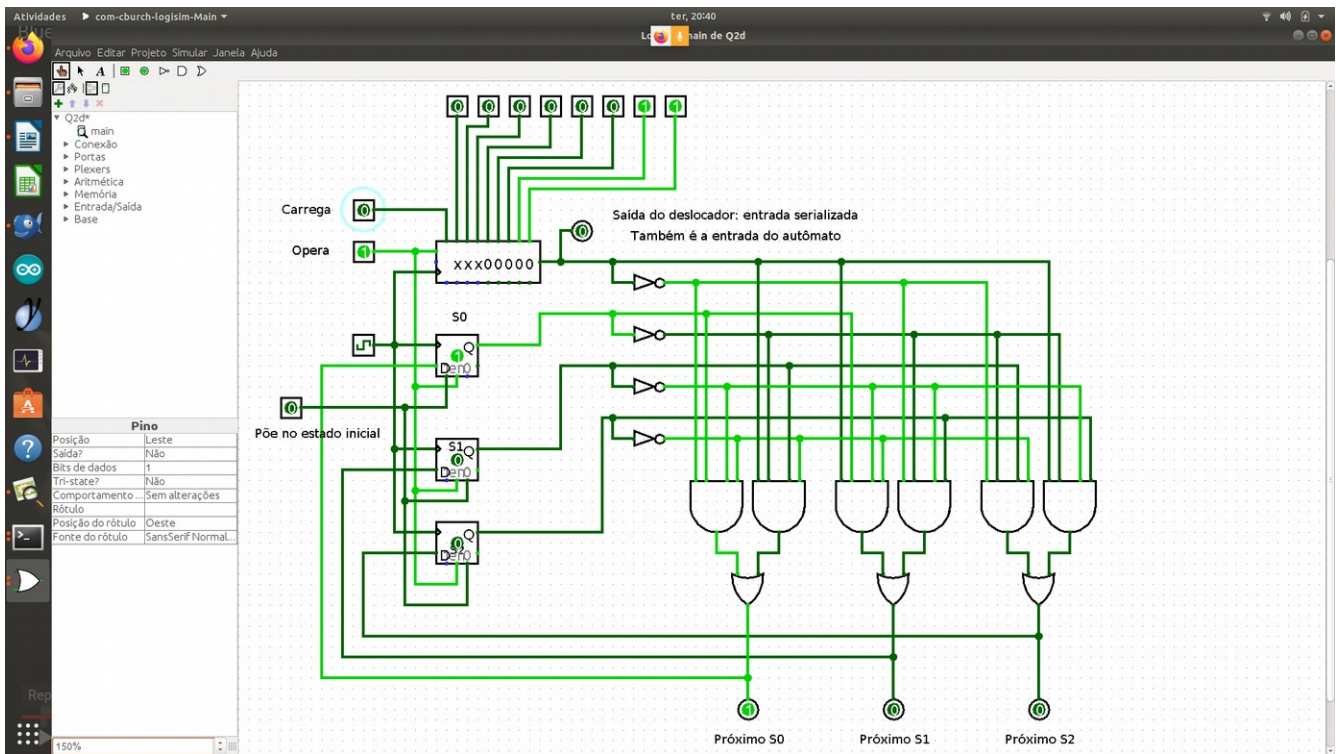
c-) considerando cada estado representado por um flip-flop, como nos autômatos dos vídeos explicativos, apresente o circuito lógico que implementa a tabela de transição de estados **(1,5pt)**;

Usando um flip-flop para representar cada estado, a tabela de transição fica:

E	S0	S1	S2	Próximo S0	Próximo S1	Próximo S2
0	1	0	0	1	0	0
1	1	0	0	0	1	0
0	0	1	0	0	0	1
1	0	1	0	1	0	0
0	0	0	1	0	1	0
1	0	0	1	0	0	1



d*-) apresente o circuito completo, com os flip-flops, entrada de 8 bits e deslocador para serialização da entrada **(PONTO EXTRA 1pt)**;



3-) O circuito abaixo calcula o produto entre o multiplicando (m) e o Multiplicador (M). O circuito para simulação no logisim pode ser baixado em (<https://drive.google.com/open?id=143cLxPqs4cs4SrLuitW9N4wfH9rKO52X>)

a-) apresente o algoritmo (código ou pseudo-código, como preferir) que simula a operação do circuito. O simulador deve conter um loop e operar números inteiros. O sinal de controle “pronto” deve ser simulado. As inicializações não precisam ser simuladas, os sinais de controle: load, clock, enable também não precisam ser simulados. **(1,0pt)**

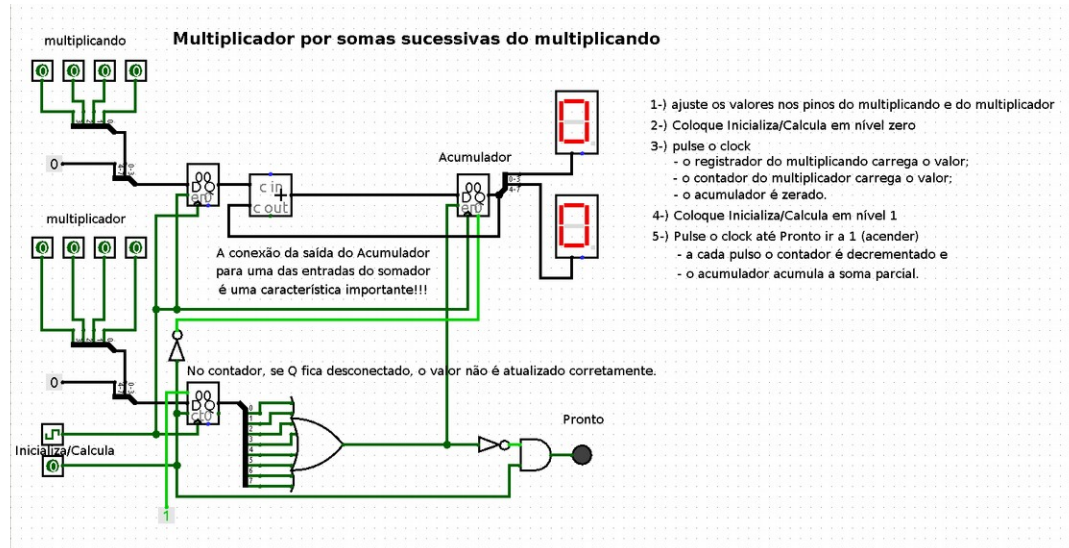
```
class Soma {
    public static void main (String[] args) {
        int multiplicando = 5;
        int Multiplicador = 4;
        int acumulador = 0;
        boolean feito = false;
        while (!feito) {
            acumulador += multiplicando;
            Multiplicador--;
            feito = Multiplicador==0;
            System.out.print ("Resultado=" + acumulador);
            System.out.println ("feito=" + feito);
        }
    }
}
```

b-) em quantos ciclos de clock a multiplicação é feita? Ou: apresente uma função de complexidade de tempo do algoritmo de multiplicação baseada na contagem de ciclos de clock (contagem de ciclos substituindo a contagem de comparações usada em Análise de Algoritmos). **(1,0pt)**

A contagem de ciclos de clock é $C = M$, onde M é o valor do multiplicador.

Caso prefira notação assintótica, $C = O(M)$, ou $C = \Theta(M)$ estão corretos.

Caso prefira notação assintótica, $C = \overline{O(M)}$, ou $C = \overline{\Theta(M)}$ estão corretos.



4-) Considere os multiplicadores i-) combinacional ii-) por deslocadores; Circuitos e explicações disponibilizados no material da disciplina. Responda e justifique em quantos ciclos de clock a multiplicação é feita (apresente uma função de complexidade de tempo do algoritmo de multiplicação baseada na contagem de ciclos de clock (contagem de ciclos substituindo a contagem de comparações usada em Análise de Algoritmos)).

ítems a avaliar na questão 4:

	função	justificativa
i-)	(0,7pt)	(1,3pt)
ii-)	(0,7pt)	(1,3pt)

i-) Não há sinal de clock, mas sinais elétricos levam algum tempo para propagar, logo o número de ciclos de clock é zero, mas o tempo para executar o cálculo é uma constante muito pequena. $C=0$, $T=O(1)$.

ii-) A cada pulso de clock o multiplicador é deslocado em um bit. Considerando M o valor do multiplicador, ele é representado por $D=1+\log_2(M)$ dígitos. $C=D$, ou se preferir a notação assintótica, $C=\Theta(\log(M))$