

Ex. 1 Desenhe os circuitos CMOS correspondentes às expressões abaixo.

- (a) $X = A \wedge B \wedge C \vee \overline{A} \wedge \overline{B} \wedge \overline{C}$
 (b) $Y = A \wedge B \vee B \wedge C \vee A \wedge C$

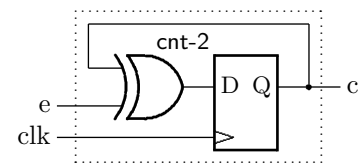
Ex. 2 Projete um circuito combinacional que computa a paridade par de uma palavra de quatro bits. O bit de paridade deve ser gerado de tal forma que o número de bits em 1 no quinteto seja par. Use um decodificador ou um seletor, além de outros componentes que julgar necessário.

Ex. 3 Você deveria estar se preparando para o exame dos Níveis Ordinários de Computação mas foi jogar xadrez ao invés de estudar. Hermione leu o Livro Verde nesse meio tempo. Após longa pausa, você retorna preguiçosamente ao trabalho e ela joga o Livro Verde na sua mesa com um marcador na página 307. Lá está o “feitiço da máquina multiplicadora de oreos”: você coloca 2 biscoitos na máquina, e enuncia o X -ésimo número na sequência de Fibonacci, e a máquina lhe devolve $2X$ biscoitos. Tudo o que falta é projetar o circuito que computa o X -ésimo número da sequência.

```

1  for ( c = 0; c < n; c++ ) {
2      if ( c <= 1 ) {
3          next = c;
4      } else {
5          next = first + second;
6          first = second;
7          second = next;
8      }
9  }
    
```

Ex. 4 Hagrid lhe pede para projetar um contador de 6 bits que deve operar na máxima velocidade possível para circuitos de tecnologia $c\delta r$ de abóbora. Ele lhe propõe um contador módulo-2, e avisa que você deve usar seis cópias deste contador e somente as portas lógicas especificadas abaixo.



- (a) Hagrid lhe mostra um circuito com os seis $cnt-1$, interconectados através de quatro portas $and-2$ e se declara contente com o projeto. Desenhe o circuito completo (mostre somente uma ‘caixa’ para cada $cnt-1$), calcule o período mínimo do relógio e verifique T_{hold} .
- (b) Você propõe outra implementação, garantidamente mais rápida, e Hagrid faz cara de incrédulo. Desenhe o seu circuito, calcule o período mínimo do relógio e verifique T_{hold} .
- (c) Hagrid tenta se safar da vergonha de perder para um aluno de primeiro ano, e lhe diz que o $skew$ de pior caso é de $-3ns$. Seu circuito ainda é mais rápido que o dele?
- (d) Repita o item (b) considerando os números nos parênteses.

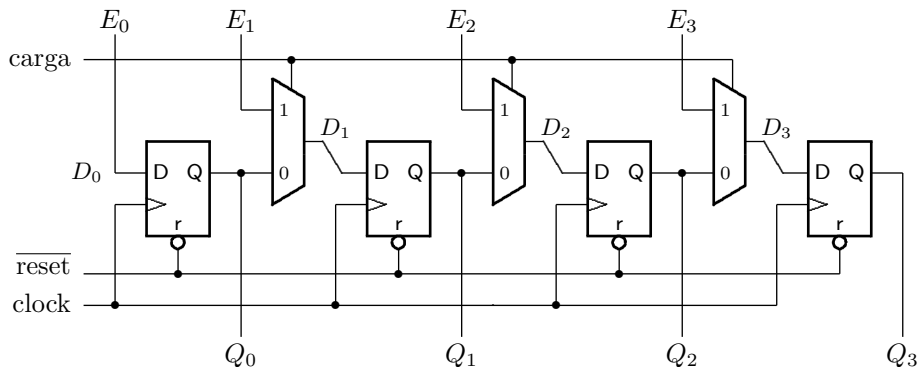
	T_{prop}	T_{cont}	T_{su}	T_h
FF	6	2	4	2
xor-2	3 (3)	1 (3)		
and-2	3 (3)	1 (3)		
and-3	3 (4)	1 (2)		
and-4	4 (5)	2 (3)		
and-5	5 (6)	2 (4)		

Tempos em nanosegundos.

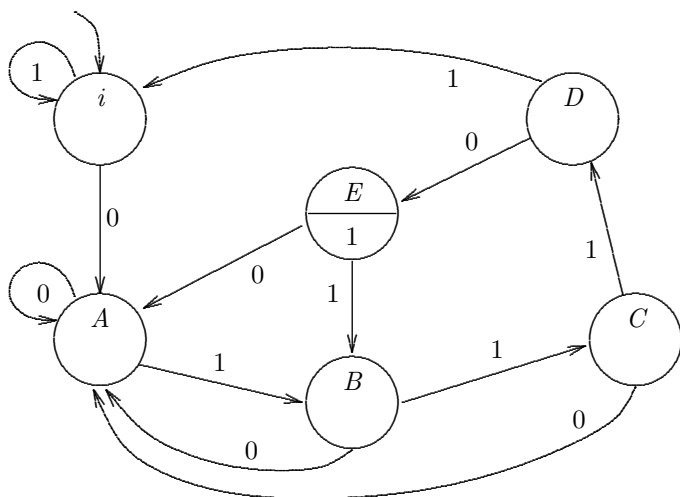
Ex. 5 Projete um circuito sequencial síncrono que produz em sua saída a sequência indicada abaixo. Sua resposta deve usar quatro $flip-flops$ e mostrar claramente todos os passos da solução.
 0000 \rightarrow 1001 \rightarrow 1111 \rightarrow 0011 \rightarrow 1010 \rightarrow 1100 \rightarrow 0000 \rightarrow ...

Ex. 6 Projete um microcontrolador que pode seguir uma de várias sequências distintas. Uma entrada de 4 bits de largura determina qual das 16 possíveis sequências será seguida. Suponha que cada sequência possui no máximo 32 microinstruções. Ao final de cada sequência, o contador de endereços é reinicializado e a primeira microinstrução é executada.

Ex. 7 Analise o circuito abaixo e desenhe um diagrama de tempo com os sinais *clock*, *carga*, $D_{0..3}$ e $Q_{0..3}$, considerando que $E_0 = \overline{Q_1}$, $E_{1..3} = 010$, e que ocorre um pulso em *carga* com largura de 1,5 períodos do relógio.



Ex. 8 Implemente a ME abaixo usando um *flip-flop* por estado.



Ex. 9 Um *árbitro* é um circuito seqüencial com N entradas e N saídas, cuja função é determinar qual dentre N requisições será atendida.

Considere um árbitro com 3 entradas. Inicialmente todas as requisições estão inativas ($E = e_2e_1e_0 = 000$) e todas as respostas estão inativas ($S = s_2s_1s_0 = 000$). Uma requisição é sinalizada por $e_i = 1$, e é atendida quando o árbitro faz $s_i = 1$. Quando o requisitante completou sua tarefa, ele retira a requisição fazendo $e_i = 0$, ao que o árbitro responde com $s_i = 0$, tão logo quanto possível. Novas requisições que ocorram enquanto alguma está sendo atendida são ignoradas. Após o encerramento de uma tarefa, uma nova rodada de arbitragem é iniciada.

Projete um árbitro para $N = 3$. Seu projeto deve conter:

- (i) um circuito combinacional que decide qual das três entradas será atendida, segundo a prioridade dada por $e_3 > e_2 > e_1$;
- (ii) uma máquina de estados que atende e/ou nega os pedidos, conforme descrito acima, implementada com flip-flops tipo D.
- (iii) Com base nos dados de temporização indicados, compute o período mínimo do relógio do árbitro, e verifique se *setup* e *hold* do seu projeto são atendidos.