

Exercícios Resolvidos: Prova P4

1Q): Usando componentes MSI, FF's e portas (*use soma de produtos*), sintetize um detector de sequencia (*M.S.S.*) *minimizado*, onde a saída Z será um (1) quando a sequencia de 2^8-1 de 1's consecutivos ocorrer, caso contrário a saída permanecerá zero (0). Obs: o número de FF's usados é no máximo 10.

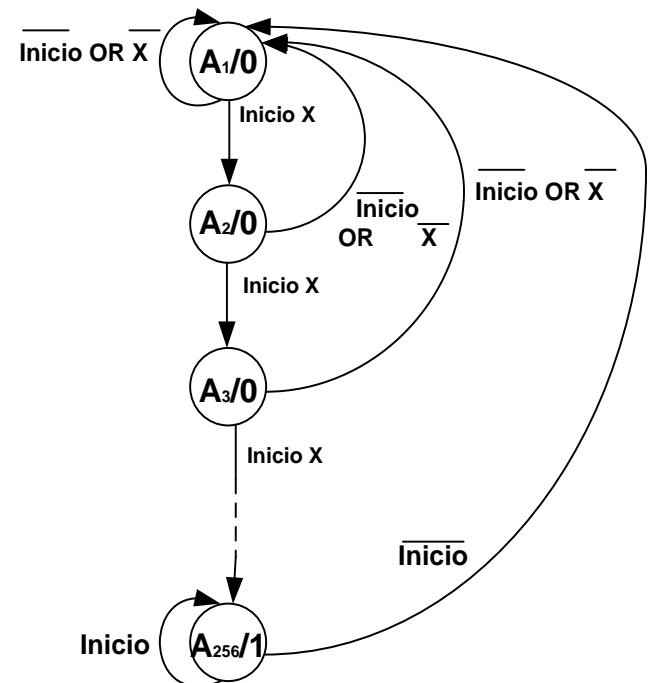
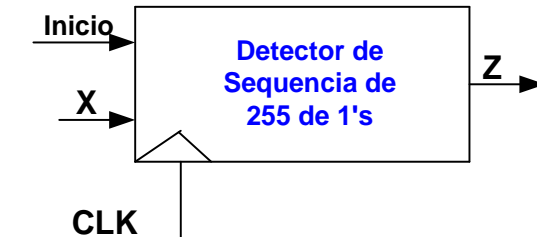
Obs: Ao detectar a sequencia, a *M.E.F.S.* permanecerá com a saída $Z=1$ e ignora as próximas sequencias.

Inicio (síncrono)=1 começa a leitura das sequencias.

Inicio (síncrono)=0 a máquina vai para o estado inicial

Solução: Nro. 1

**Tratar o problema
como se fosse
somente uma MEF**



Exercícios Resolvidos: Prova P4

1Q): Usando componentes MSI, FF's e portas (*use soma de produtos*), sintetize um detector de sequencia (*M.S.S.*) *minimizado*, onde a saída Z será um (1) quando a sequencia de 2^8-1 de 1's consecutivos ocorrer, caso contrário a saída permanecerá zero (0). Obs: o número de FF's usados é no máximo 10.

Obs: Ao detectar a sequencia, a *M.E.F.S.* permanecerá com a saída $Z=1$ e ignora as próximas sequencias.

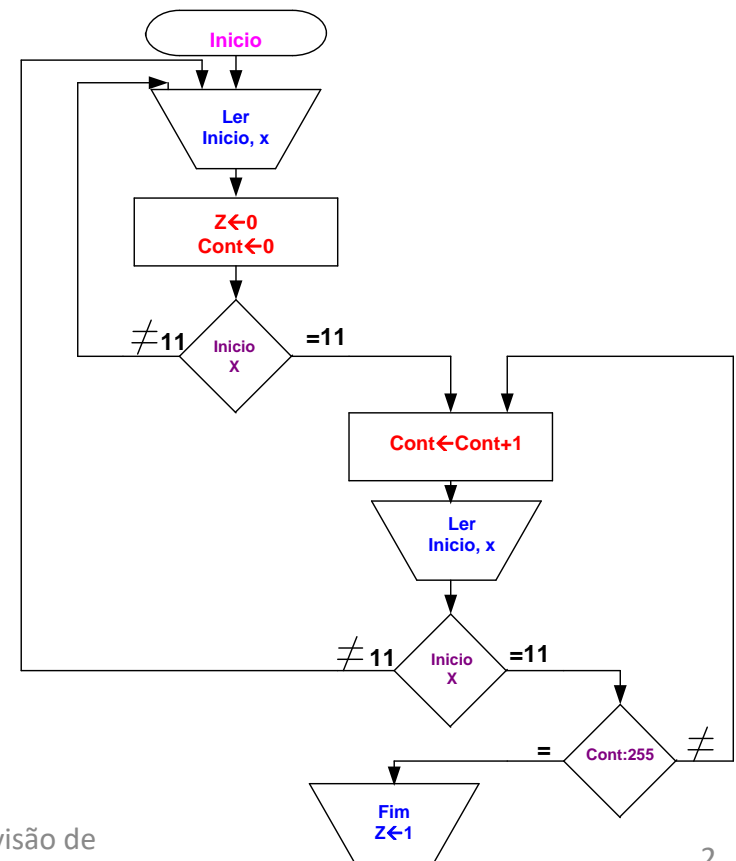
Inicio (síncrono)=1 começa a leitura das sequencias.

Inicio (síncrono)=0 a máquina vai para o estado inicial

Solução: Nro. 2

**Tratar o problema
como uma
Decomposição:
MEF + Data-path
simplificado**

**Fluxograma do
algoritmo → 1 parte**



Exercícios Resolvidos: Prova P4

1Q): Usando componentes MSI, FF's e portas (*use soma de produtos*), sintetize um detector de sequencia (*M.S.S.*) *minimizado*, onde a saída Z será hum (1) quando a sequencia de 2^8-1 de 1's consecutivos ocorrer, caso contrário a saída permanecerá zero (0). Obs: o número de FF's usados é no máximo 10.

Obs: Ao detectar a sequencia, a *M.E.F.S.* permanecerá com a saída $Z=1$ e ignora as próximas sequencias.

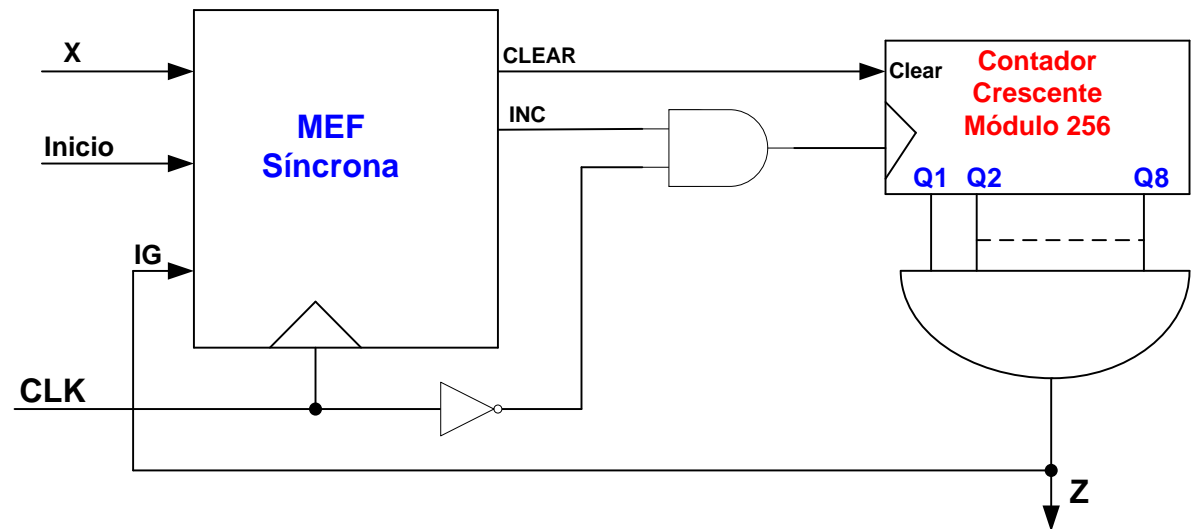
Inicio (síncrono)=1 começa a leitura das sequencias.

Inicio (síncrono)=0 a máquina vai para o estado inicial

Solução: Nro. 2

**Tratar o problema
como uma
Decomposição:
MEF + Data-path
simplificado**

**Estrutura geral da
decomposição: 2 parte**



Exercícios Resolvidos: Prova P4

1Q): Usando componentes MSI, FF's e portas (*use soma de produtos*), sintetize um detector de sequencia (*M.S.S.*) *minimizado*, onde a saída Z será hum (1) quando a sequencia de 2^8-1 de 1's consecutivos ocorrer, caso contrário à saída permanecerá zero (0). Obs: o número de FF's usados é no máximo 10.

Obs: Ao detectar a sequencia, a *M.E.F.S.* permanecerá com a saída $Z=1$ e ignora as próximas sequencias.

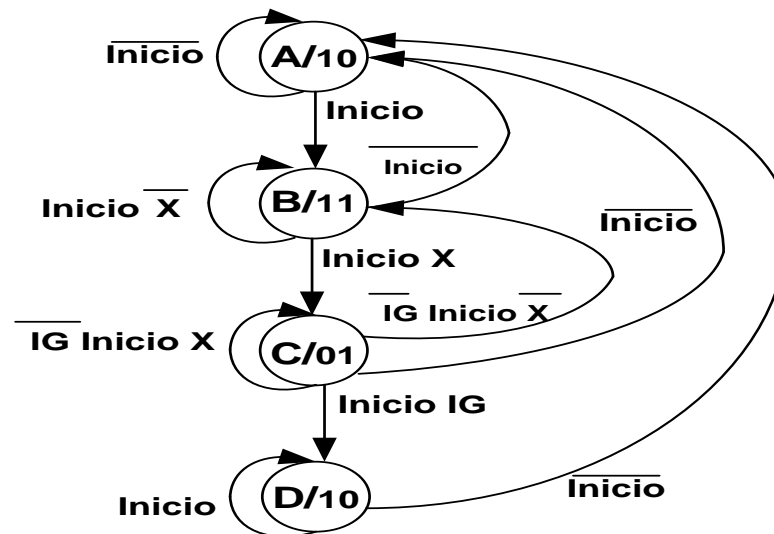
Inicio (síncrono)=1 começa a leitura das sequencias.

Inicio (síncrono)=0 a máquina vai para o estado inicial

Solução: Nro. 2

**Tratar o problema
como uma
Decomposição:
MEF + Data-path
simplificado**

**Diagrama de estados da
MEF: 2 parte**



Exercícios Resolvidos: Prova P4

1Q): Usando componentes MSI, FF's e portas (*use soma de produtos*), sintetize um detector de sequencia (*M.S.S.*) *minimizado*, onde a saída Z será hum (1) quando a sequencia de 2^8-1 de 1's consecutivos ocorrer, caso contrário a saída permanecerá zero (0). Obs: o número de FF's usados é no máximo 10.

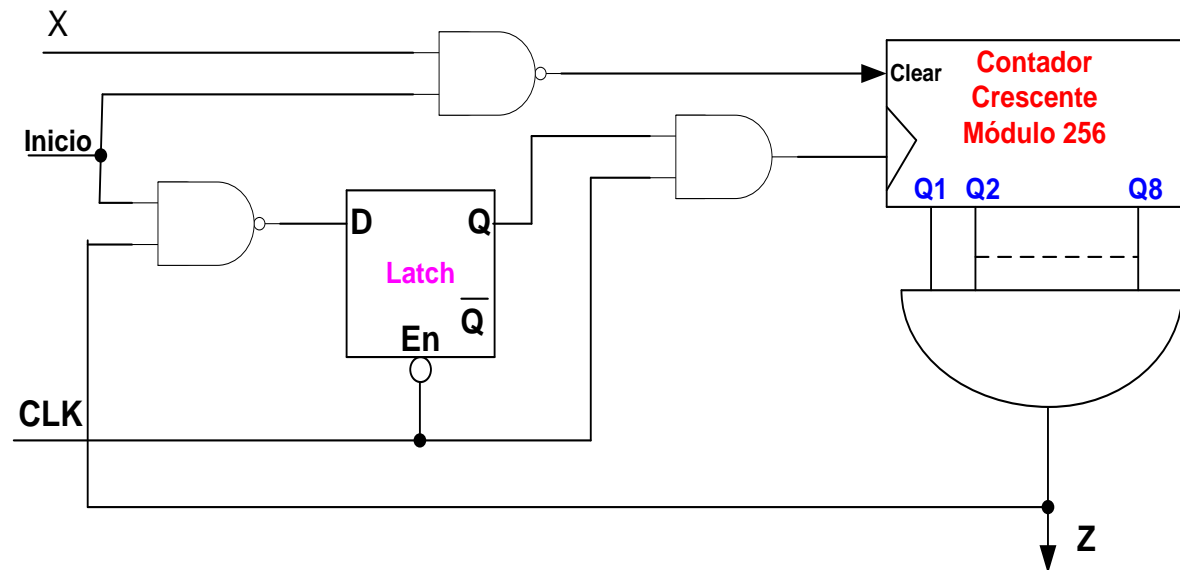
Obs: Ao detectar a sequencia, a *M.E.F.S.* permanecerá com a saída $Z=1$ e ignora as próximas sequencias.

Inicio (síncrono)=1 começa a leitura das sequencias.

Inicio (síncrono)=0 a máquina vai para o estado inicial

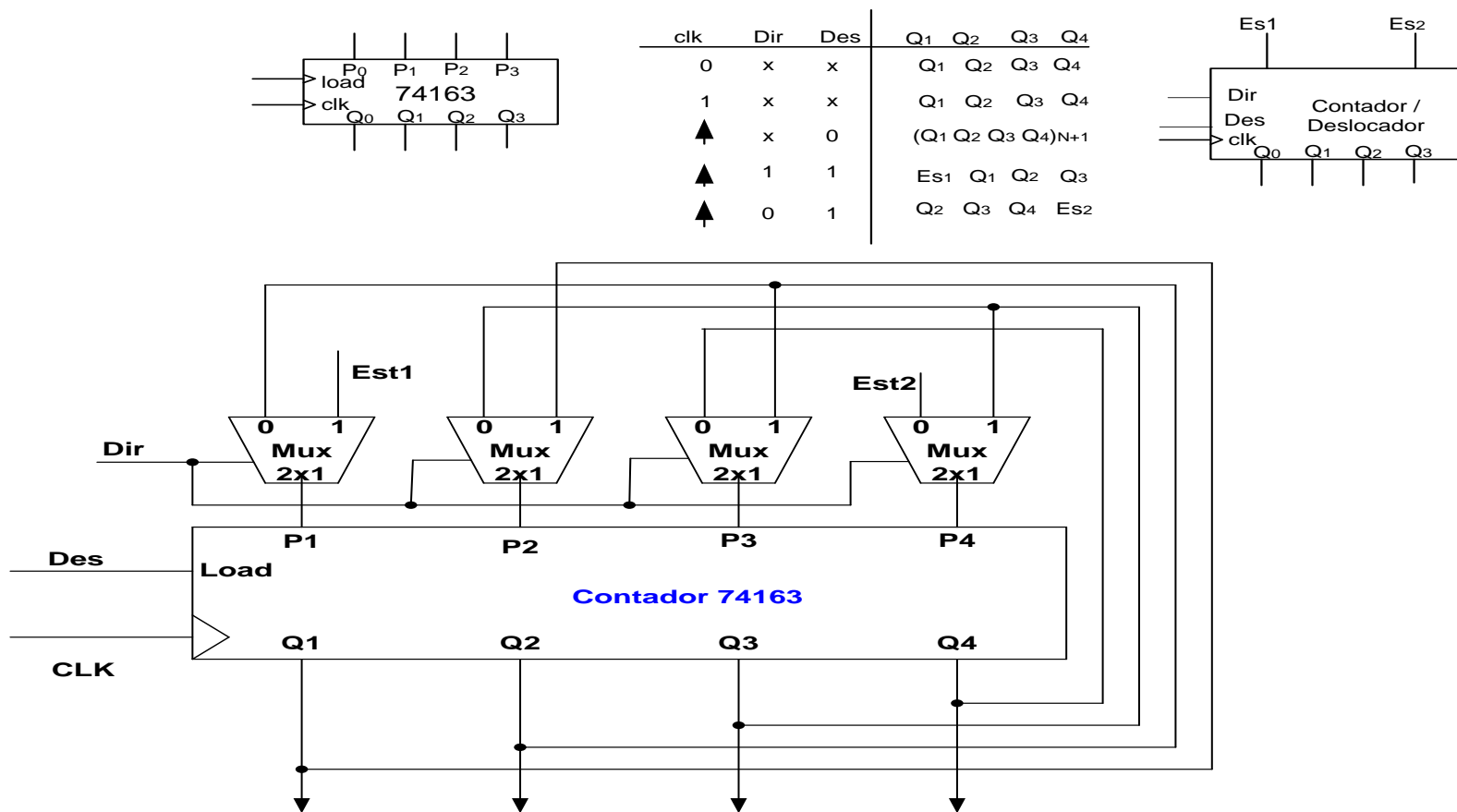
Solução: Nro. 3

**Tratar o problema
como uma
Composição de
componentes**



Exercícios Resolvidos: Prova P4

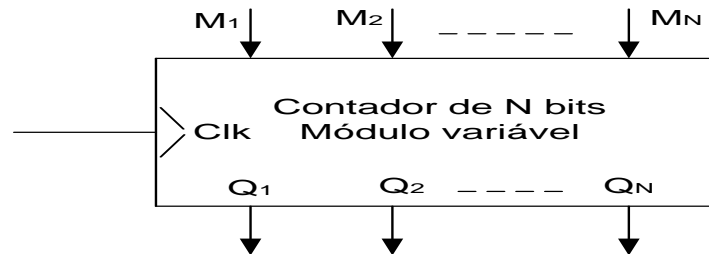
2Q: Usando somente um contador 74163, Mux's e portas, **projete** um **contador** binário de quatro bits com deslocamento bidirecional, onde a sua **tabela de operações** está descrita abaixo.



Exercícios Resolvidos: Prova P4

3Q): Usando a técnica de redes iterativa sintetize uma célula básica de um contador síncrono binário crescente de N bits com módulo variável.

IF $Q_N = M_N$ THEN $Q_{N+1} \leftarrow \text{Zero}$
ELSE $Q_{N+1} \leftarrow Q_N + 1$

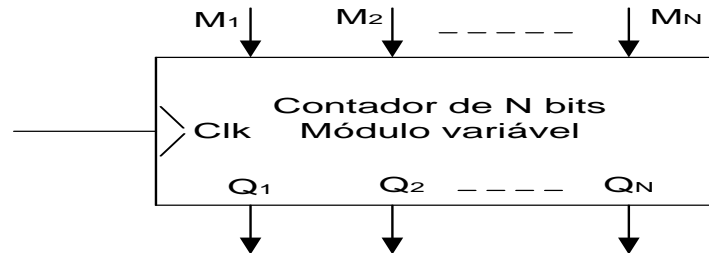


Exercícios Resolvidos: Prova P4

3Q): Usando a técnica de redes iterativa sintetize uma célula básica de um contador síncrono binário crescente de N bits com módulo variável.

IF $Q_N = M_N$ THEN $Q_{N+1} \leftarrow \text{Zero}$
ELSE $Q_{N+1} \leftarrow Q_N + 1$

Solução: especificar uma célula básica de 1 bit.

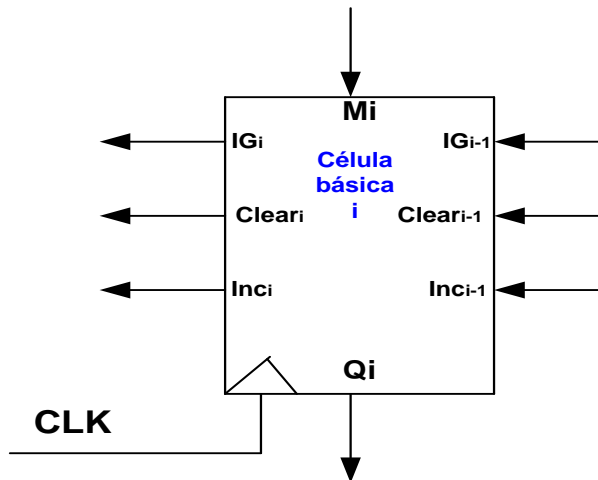


1) $I_{Gi}=1$ Se $M_i=Q_i$ e $I_{Gi-1}=1$
Caso contrário $I_{Gi}=0$

2) $Clear_i=1$ Se $Clear_{i-1}=1$
Caso contrário $Clear_i=0$

3) $T_i=1$ Se $Inc_{i-1}=1$ OR $Q_i=1$ and $Clear_{i-1}=1$
Caso contrário $T_i=0$

4) $Inc_i=1$ Se $Q_i=1$ e $Inc_{i-1}=1$
Caso contrário $Inc_i=0$

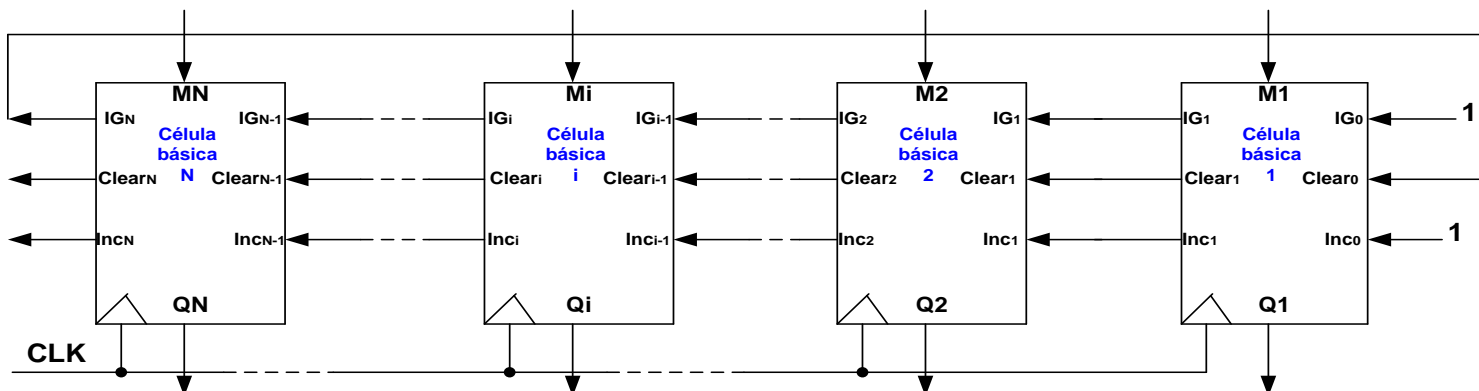
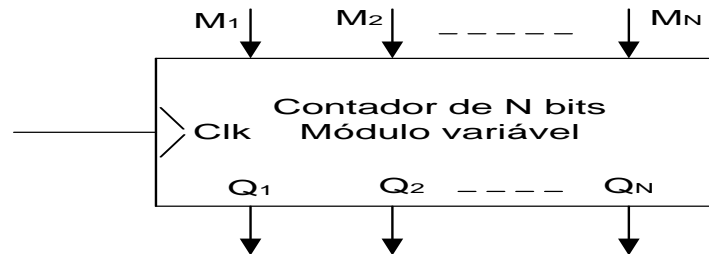


Exercícios Resolvidos: Prova P4

3Q): Usando a técnica de redes iterativa sintetize uma célula básica de um contador síncrono binário crescente de N bits com módulo variável.

IF $QN=MN$ THEN $QN+1 \leftarrow \text{Zero}$
ELSE $QN+1 \leftarrow QN + 1$

Solução: especificar uma célula básica de 1 bit.

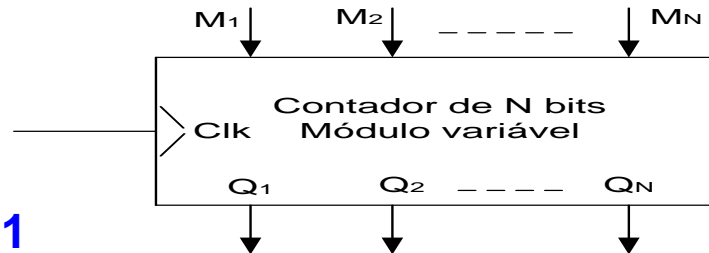


Exercícios Resolvidos: Prova P4

3Q): Usando a técnica de redes iterativa sintetize uma célula básica de um contador síncrono binário crescente de N bits com módulo variável.

IF $Q_N = M_N$ THEN $Q_{N+1} \leftarrow \text{Zero}$
ELSE $Q_{N+1} \leftarrow Q_N + 1$

**Solução: especificar
uma célula básica de
1 bit.**

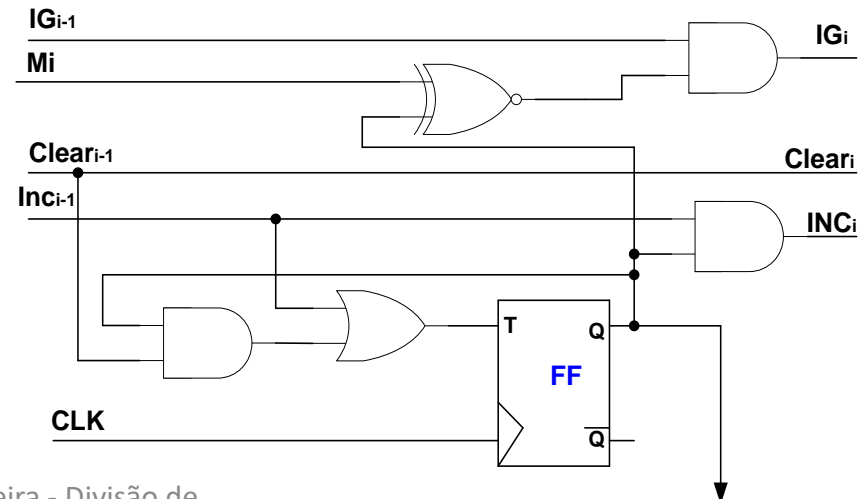


1) $IG_i = 1$ Se $M_i = Q_i$ e $IG_{i-1} = 1$
Caso contrário $IG_i = 0$

2) $\text{Clear}_i = 1$ Se $\text{Clear}_{i-1} = 1$
Caso contrário $\text{Clear}_i = 0$

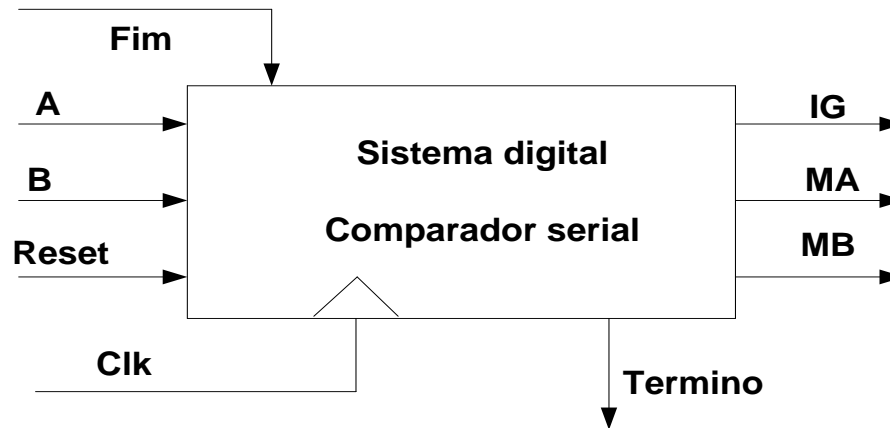
3) $T_i = 1$ Se $\text{Inc}_{i-1} = 1$ OR $Q_i = 1$ and $\text{Clear}_{i-1} = 1$
Caso contrário $T_i = 0$

4) $\text{Inc}_i = 1$ Se $Q_i = 1$ e $\text{Inc}_{i-1} = 1$
Caso contrário $\text{Inc}_i = 0$



Exercícios Resolvidos: Prova P4

4Q): Usando funções MSI, Flip-Flops e portas, sintetize um comparador serial na forma minimizada que possui as seguintes características: Para a variável $Reset=1$ o comparador inicia a comparação das variáveis A e B . A variável Fim indica o tamanho de A e B ($Fim=1$). A variável $Término$ indica o fim da comparação ($Término=1$) e as outras variáveis de saída assumem os seguintes valores: $IG=1$ ($A=B$), $MA=1$ ($A>B$) e $MB=1$ ($B>A$). Importante As variáveis A e B são inseridas pelo menos significativo. **Obs:** Para uma nova comparação ou interrupção a variável $Start$ deve ir para **zero** e no próximo clock para **um**. A variável Fim deve ir para **zero** para uma nova comparação.



Exercícios Resolvidos: Prova P4

4Q): Usando funções MSI, Flip-Flops e portas, sintetize um comparador serial na forma minimizada que possui as seguintes características: Para a variável *Reset=1* o comparador inicia a comparação das variáveis *A* e *B*. A variável *Fim* indica o tamanho de *A* e *B* (*Fim=1*). A variável *Término* indica o fim da comparação (*Término=1*) e as outras variáveis de saída assumem os seguintes valores: *IG=1* (*A=B*), *MA=1* (*A>B*) e *MB=1* (*B>A*). Importante As variáveis *A* e *B* são inseridas pelo menos significativo. **Obs:** Para uma nova comparação ou interrupção a variável *Start* deve ir para **zero** e no próximo clock para **hum**. A variável *Fim* deve ir para **zero** para uma nova comparação.

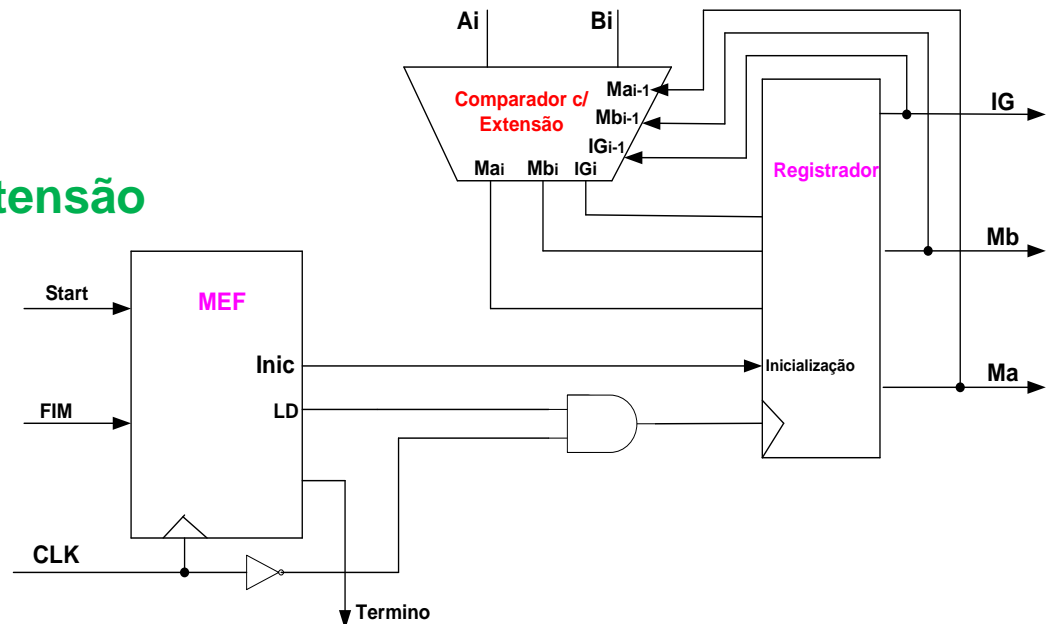
Solução de Nro. 1
Decompor em
MEF + data-path

Equações: comparador com extensão

$$Ma_i = A_i \overline{B_i} + \overline{A_i} B_i Ma_{i-1}$$

$$Mb_i = \overline{A_i} B_i + A_i \overline{B_i} Mb_{i-1}$$

$$IG_i = A_i \oplus B_i IG_{i-1}$$



Exercícios Resolvidos: Prova P4

4Q): Usando funções MSI, Flip-Flops e portas, sintetize um comparador serial na forma minimizada que possui as seguintes características: Para a variável $Start=1$ o comparador inicia a comparação das variáveis A e B . A variável Fim indica o tamanho de A e B ($Fim=1$). A variável $Término$ indica o fim da comparação ($Término=1$) e as outras variáveis de saída assumem os seguintes valores: $IG=1$ ($A=B$), $MA=1$ ($A>B$) e $MB=1$ ($B>A$). Importante As variáveis A e B são inseridas pelo menos significativo. **Obs:** Para uma nova comparação ou interrupção a variável $Start$ deve ir para **zero** e no próximo clock para **hum**. A variável Fim deve ir para **zero** para uma nova comparação.

Solução de Nro. 1 Decompor em MEF + data-path

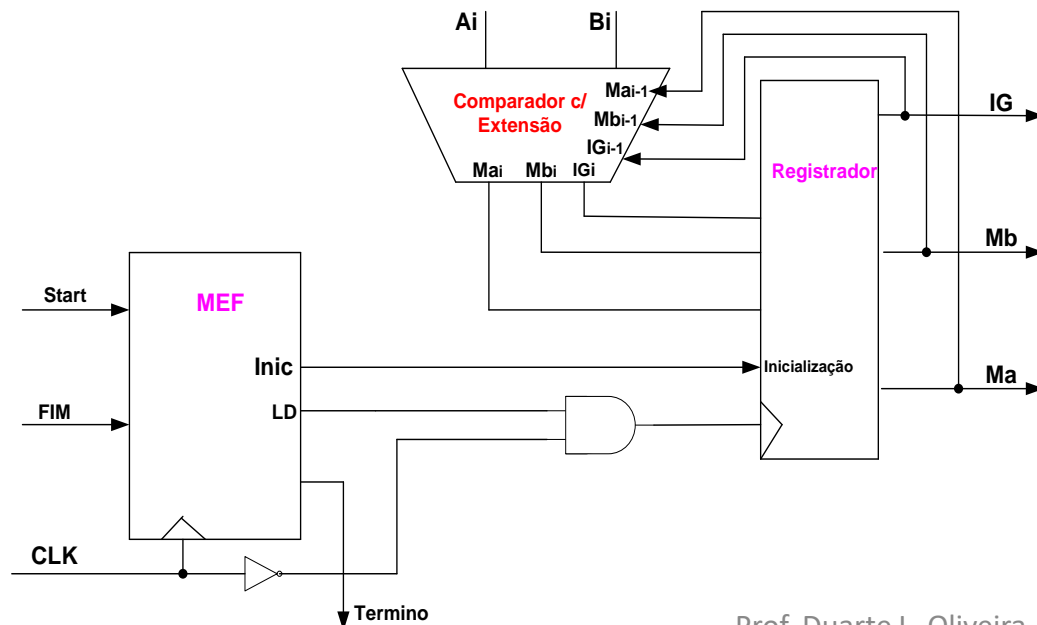
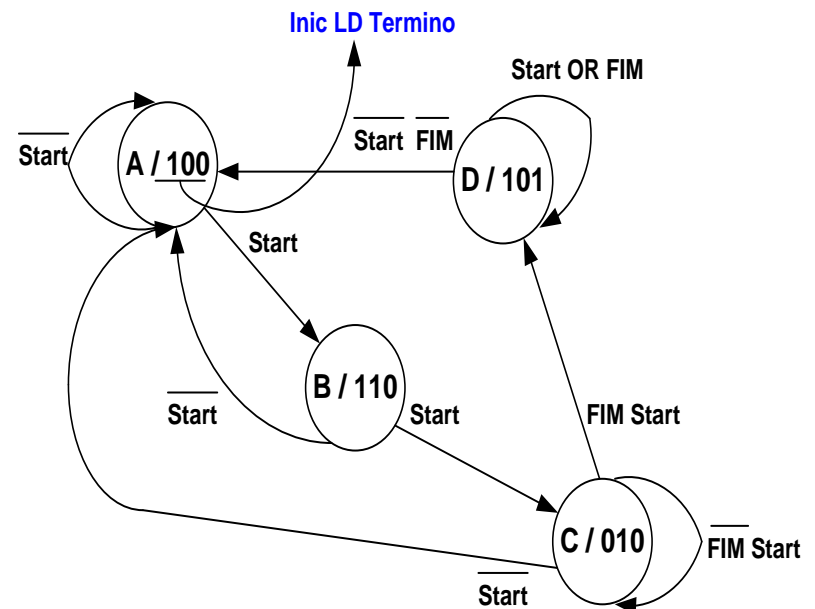


Diagrama de estados



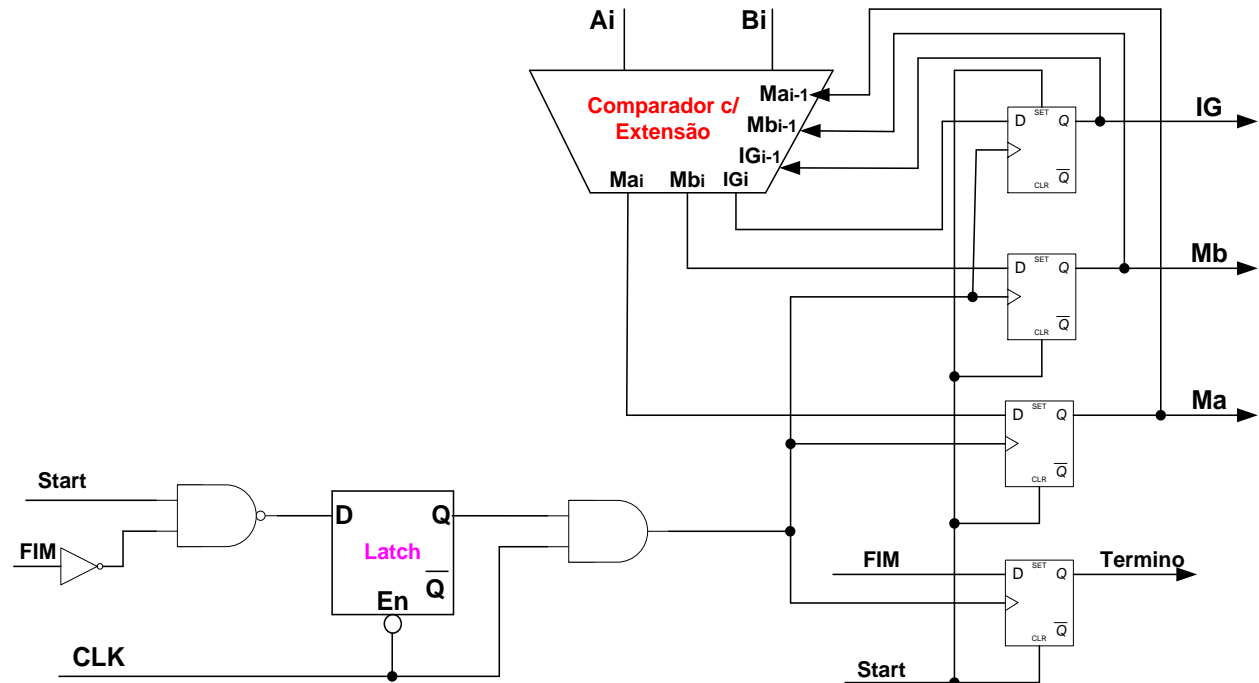
Exercícios Resolvidos: Prova P4

4Q): Usando funções MSI, Flip-Flops e portas, sintetize um comparador serial na forma minimizada que possui as seguintes características: Para a variável $Reset=1$ o comparador inicia a comparação das variáveis A e B . A variável Fim indica o tamanho de A e B ($Fim=1$). A variável $Término$ indica o fim da comparação ($Término=1$) e as outras variáveis de saída assumem os seguintes valores: $IG=1$ ($A=B$), $MA=1$ ($A>B$) e $MB=1$ ($B>A$). Importante As variáveis A e B são inseridas pelo menos significativo. **Obs:** Para uma nova comparação ou interrupção a variável $Start$ deve ir para **zero** e no próximo clock para **hum**. A variável Fim deve ir para **zero** para uma nova comparação.

Solução de Nro. 2

Tratar o problema
como uma
Composição de
componentes

Não recomendado:
Não tem
procedimento de
engenharia → não
tem diagrama de
estados → difícil de
testar



Exercícios Resolvidos: Prova P4

5Q): Projetar um circuito digital síncrono que tem a função de incrementador programável. Este circuito processa segundo a tabela de operações descrita na figura. No término do número de clocks segundo S_1 e S_2 , temos o novo valor dos Q's. Use funções MSI, FFs e portas. Por exemplo, quando $S_1=S_2=1$ e depois de quatro clocks temos um incremento $Q \leftarrow Q+1$. Quando $Q=1111 \rightarrow 0000$ é o próximo incremento.

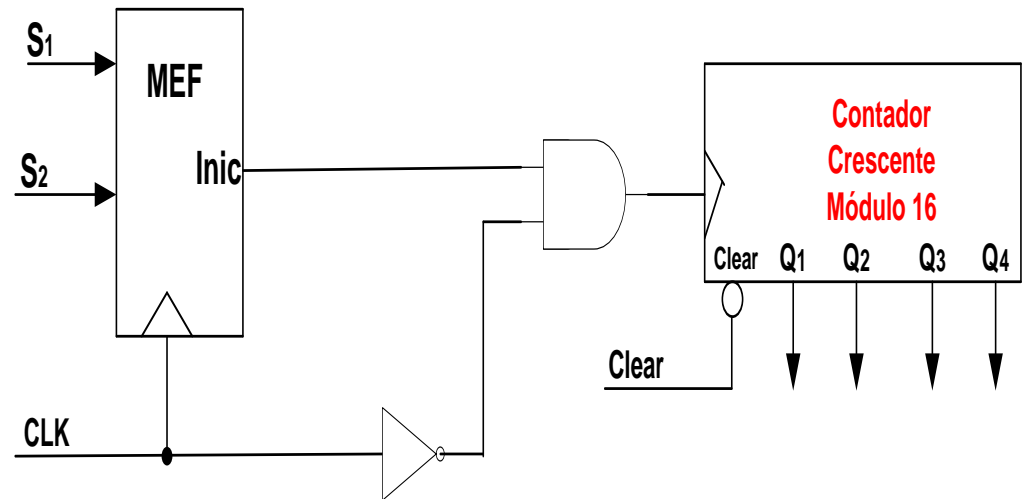
CLK	S_1	S_2	Q_1	Q_2	Q_3	Q_4
\uparrow	0	0	inibe			
(2) \uparrow	0	1	$Q_{N+1} \leftarrow Q_N + 1$			
(3) \uparrow	1	0	$Q_{N+1} \leftarrow Q_N + 1$			
(4) \uparrow	1	1	$Q_{N+1} \leftarrow Q_N + 1$			

Exercícios Resolvidos: Prova P4

5Q): Projetar um circuito digital síncrono que tem a função de incrementador programável. Este circuito processa segundo a tabela de operações descrita na figura. No término do número de clocks segundo S_1 e S_2 , temos o novo valor dos Q's. Use funções MSI, FFs e portas. Por exemplo, quando $S_1=S_2=1$ e depois de quatro clocks temos um incremento $Q \leftarrow Q+1$. Quando $Q=1111 \rightarrow 0000$ é o próximo incremento.

Solução: Decomposição \rightarrow MEF + data-path simplificado

CLK	S_1	S_2	Q_1	Q_2	Q_3	Q_4
\uparrow	0	0	inibe			
(2) \uparrow	0	1	$Q_{N+1} \leftarrow Q_N + 1$			
(3) \uparrow	1	0	$Q_{N+1} \leftarrow Q_N + 1$			
(4) \uparrow	1	1	$Q_{N+1} \leftarrow Q_N + 1$			



Exercícios Resolvidos: Prova P4

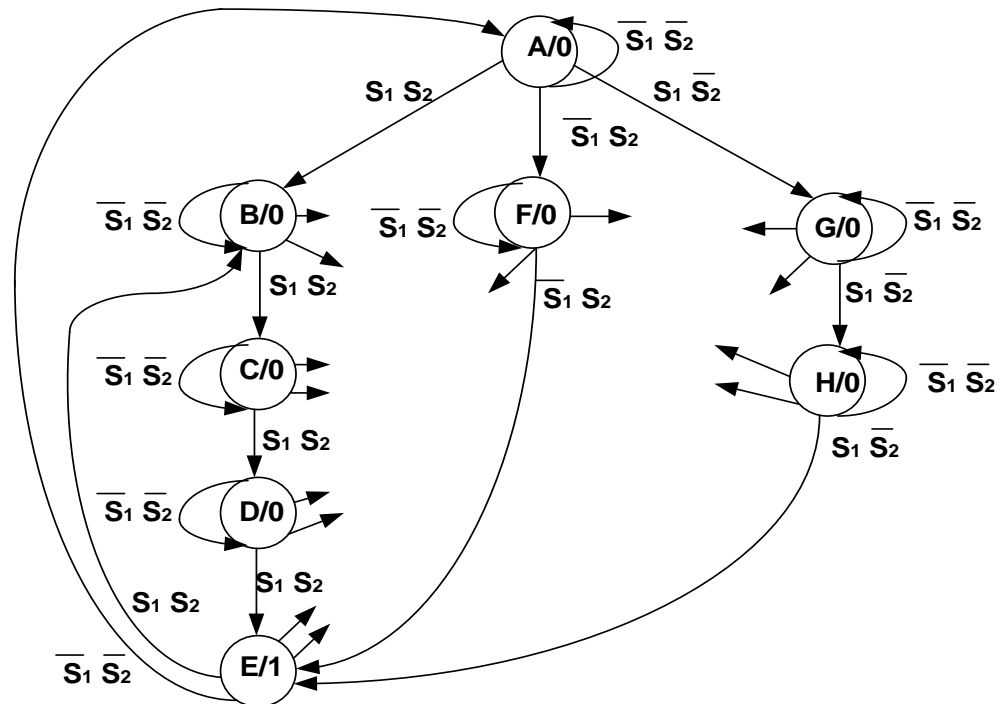
5Q): Projetar um circuito digital síncrono que tem a função de incrementador programável. Este circuito processa segundo a tabela de operações descrita na figura. No término do número de clocks segundo S_1 e S_2 , temos o novo valor dos Q's. Use funções MSI, FFs e portas. Por exemplo, quando $S_1=S_2=1$ e depois de quatro clocks temos um incremento $Q \leftarrow Q+1$. Quando $Q=1111 \rightarrow 0000$ é o próximo incremento.

Solução: Decomposição \rightarrow MEF + data-path simplificado

CLK	S_1	S_2	Q_1	Q_2	Q_3	Q_4
\uparrow	0	0	inibe			
(2) \uparrow	0	1	$Q_{N+1} \leftarrow Q_N + 1$			
(3) \uparrow	1	0	$Q_{N+1} \leftarrow Q_N + 1$			
(4) \uparrow	1	1	$Q_{N+1} \leftarrow Q_N + 1$			

**Diagrama de estados parcial:
Descrição total é confusa**

**Melhor forma de
descrição é a tabela
de estados**



Exercícios Resolvidos: Prova P4

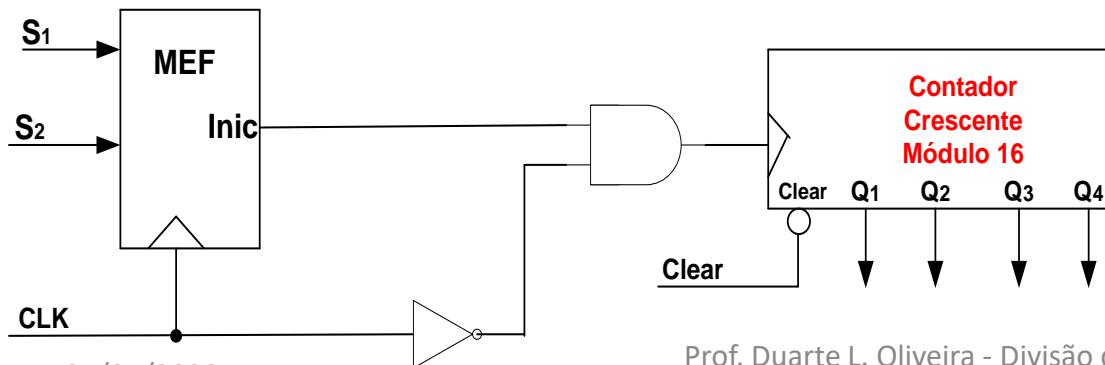
5Q): Projetar um circuito digital síncrono que tem a função de incrementador programável. Este circuito processa segundo a tabela de operações descrita na figura. No término do número de clocks segundo S_1 e S_2 , temos o novo valor dos Q's. Use funções MSI, FFs e portas. Por exemplo, quando $S_1=S_2=1$ e depois de quatro clocks temos um incremento $Q \leftarrow Q+1$. Quando $Q=1111 \rightarrow 0000$ é o próximo incremento.

Solução: Decomposição \rightarrow MEF + data-path simplificado

CLK	S_1	S_2	Q_1	Q_2	Q_3	Q_4
\uparrow	0	0	inibe			
(2) \uparrow	0	1	$Q_{N+1} \leftarrow Q_N + 1$			
(3) \uparrow	1	0	$Q_{N+1} \leftarrow Q_N + 1$			
(4) \uparrow	1	1	$Q_{N+1} \leftarrow Q_N + 1$			

Tabela de estados

$S_1 S_2$ Estados	00	01	11	10	Inic
A	A	F	B	G	0
B	B	F	C	G	0
C	C	F	D	G	0
D	D	F	E	G	0
E	A	F	B	G	1
F	F	E	B	G	0
G	G	F	B	H	0
H	H	F	B	E	0



Exercícios Resolvidos: Prova P4

6Q: *Obter* o diagrama de estados modelo Moore minimizado de um controlador de semáforo de um cruzamento de duas avenidas de sentido único. Os **sensores** Sv1 e Sv2 respectivamente detectam a presença de automóveis nas avenidas 1 e 2. A prioridade das duas avenidas é a mesma. A sequência em cada semáforo segue o tradicional (Verde → Amarelo → vermelho → Verde). Os dois semáforos nas duas avenidas são ativados por (AS10, AS11) e (AS20, AS21). O semáforo tem o seguinte comportamento: O semáforo no verde permanece 15 segundos, no término deste tempo a máquina consulta o sensor contrário (semáforo em vermelho) se ele estiver em alta o semáforo irá para o amarelo e a assim por diante. Caso contrário o semáforo permanecerá mais 15 segundos. Figura 6 mostra a estrutura geral do sistema digital.

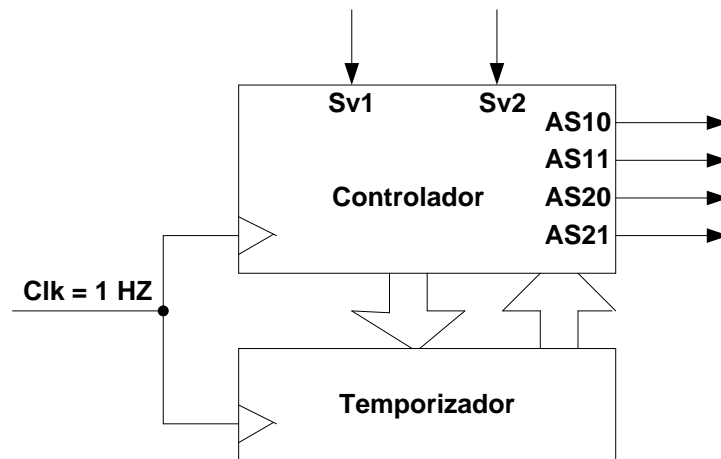


Figura 6. Estrutura geral: circuito

Exercícios Resolvidos: Prova P4

6Q: *Obter* o diagrama de estados modelo Moore minimizado de um controlador de semáforo de um cruzamento de duas avenidas de sentido único. Os **sensores** Sv1 e Sv2 respectivamente detectam a presença de automóveis nas avenidas 1 e 2. A prioridade das duas avenidas é a mesma. A sequência em cada semáforo segue o tradicional (Verde → Amarelo → vermelho → Verde). Os dois semáforos nas duas avenidas são ativados por (AS10, AS11) e (AS20, AS21). O semáforo tem o seguinte comportamento: O semáforo no verde permanece 15 segundos, no termino deste tempo a máquina consulta o sensor contrário (semáforo em vermelho) se ele estiver em alta o semáforo irá para o amarelo e a assim por diante. Caso contrário o semáforo permanecerá mais 15 segundos. Figura 6 mostra a estrutura geral do sistema digital.

Solução:

Definições:

P → pedido de tempo (0→1)

T → T=1 → tempo processado

(Sv1, Sv2) → 1 detecta automóvel

AS11	AS10	Definição
0	0	Amarelo
0	1	Verde
1	0	Vermelho

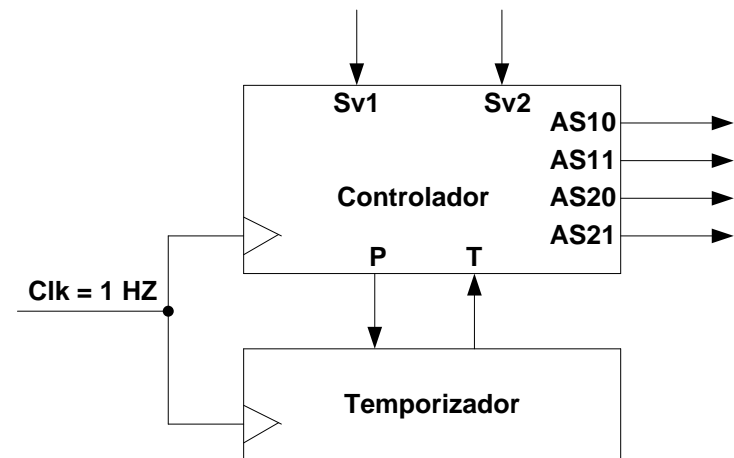
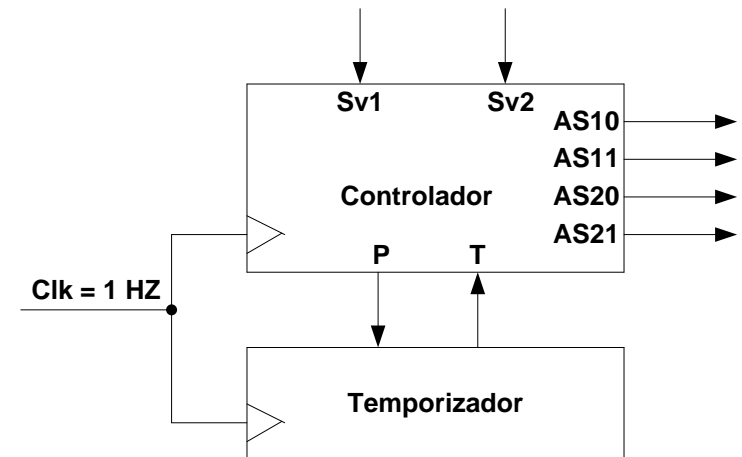
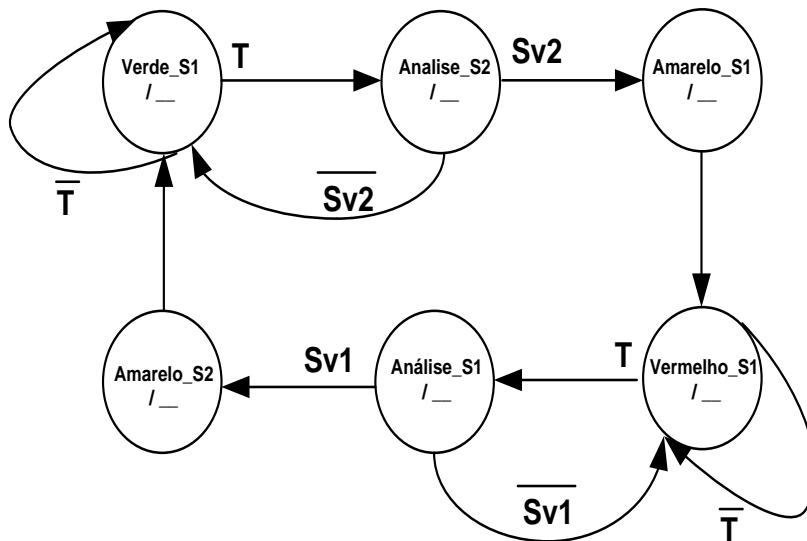


Figura 6. Estrutura geral: circuito

Exercícios Resolvidos: Prova P4

6Q: Obter o diagrama de estados modelo Moore minimizado de um controlador de semáforo de um cruzamento de duas avenidas de sentido único. Os sensores Sv1 e Sv2 respectivamente detectam a presença de automóveis nas avenidas 1 e 2. A prioridade das duas avenidas é a mesma. A sequência em cada semáforo segue o tradicional (Verde → Amarelo → vermelho → Verde). Os dois semáforos nas duas avenidas são ativados por (AS10, AS11) e (AS20, AS21). O semáforo tem o seguinte comportamento: O semáforo no verde permanece 15 segundos, no término deste tempo a máquina consulta o sensor contrário (semáforo em vermelho) se ele estiver em alta o semáforo irá para o amarelo e a assim por diante. Caso contrário o semáforo permanecerá mais 15 segundos. Figura 6 mostra a estrutura geral do sistema digital.

Solução:



Exercícios Resolvidos: Prova P4

6Q: Obter o diagrama de estados modelo Moore minimizado de um controlador de semáforo de um cruzamento de duas avenidas de sentido único. Os sensores Sv1 e Sv2 respectivamente detectam a presença de automóveis nas avenidas 1 e 2. A prioridade das duas avenidas é a mesma. A sequência em cada semáforo segue o tradicional (Verde → Amarelo → vermelho → Verde). Os dois semáforos nas duas avenidas são ativados por (AS10, AS11) e (AS20, AS21). O semáforo tem o seguinte comportamento: O semáforo no verde permanece 15 segundos, no termino deste tempo a máquina consulta o sensor contrário (semáforo em vermelho) se ele estiver em alta o semáforo irá para o amarelo e a assim por diante. Caso contrário o semáforo permanecerá mais 15 segundos. Figura 6 mostra a estrutura geral do sistema digital.

Solução:

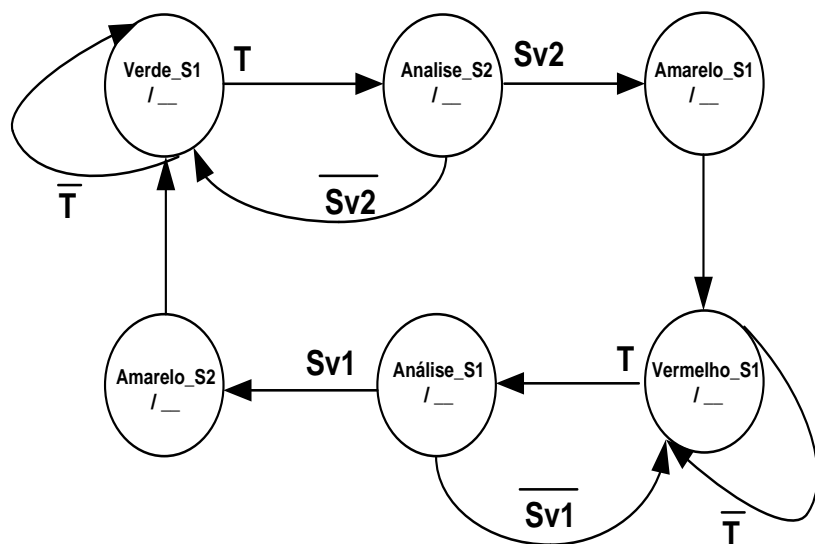


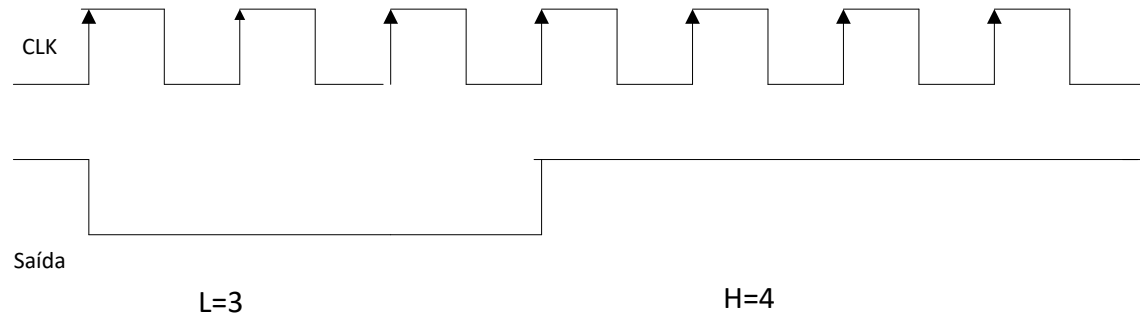
Tabela de saídas

Estados	AS11	AS10	AS21	AS20	P
Verde_S1	0	1	1	0	1
Análise_S2	0	1	1	0	0
Amarelo_S1	0	0	1	0	1
Vermelho_S1	1	0	0	1	1
Análise_S1	1	0	0	1	0
Amarelo_S2	1	0	0	0	1

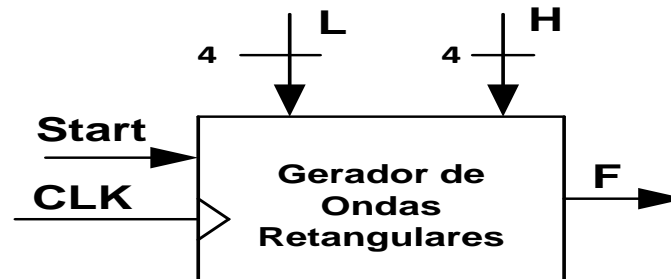
Exercícios Resolvidos: Prova P4

7Q: Projetar um **gerador de ondas retangulares** que, acionado por um sinal de clock periódico gere pulsos, nos quais o **tempo de permanência** no nível 1 seja igual a tantos períodos de clock, quanto o número binário H de 4 bits. De mesma forma a duração do pulso de saída no nível baixo é determinado pela via L de 4 bits. A saída Z é de 1 bit. Obs: use portas, FFs e funções MSI.

**Exemplo: Para
L=3 e H=4**



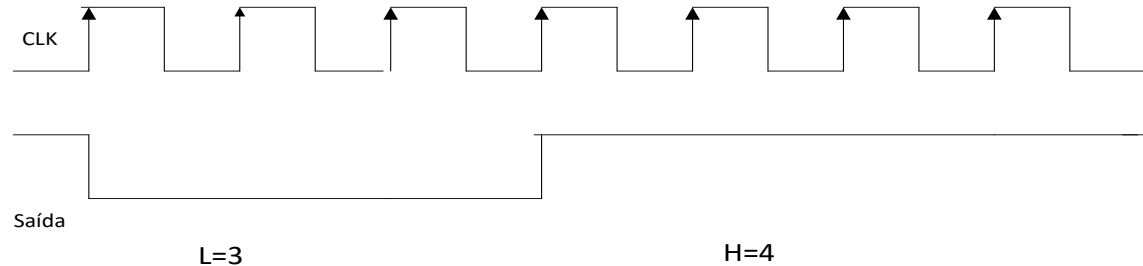
Solução:



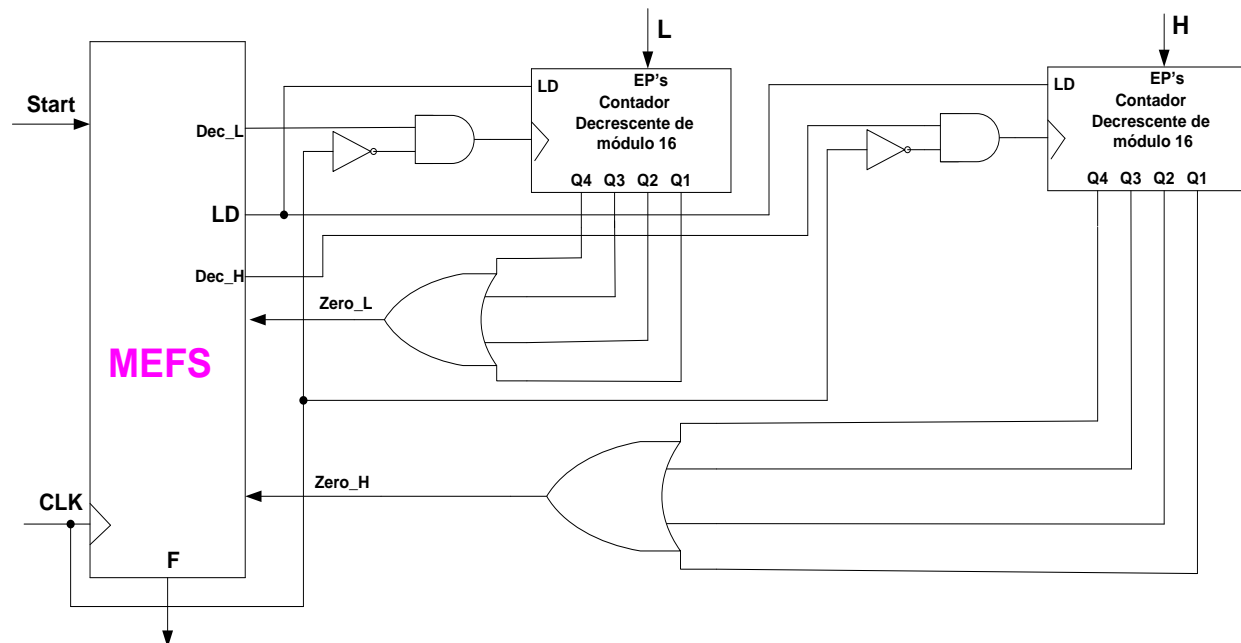
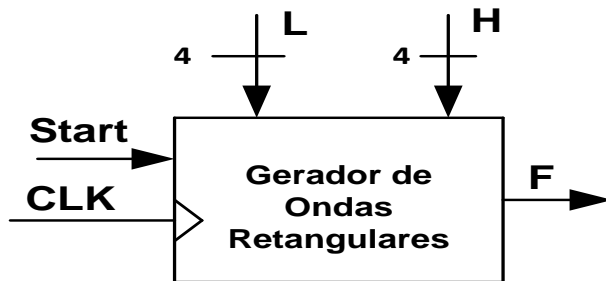
Exercícios Resolvidos: Prova P4

7Q: Projetar um **gerador de ondas retangulares** que, acionado por um sinal de clock periódico gere pulsos, nos quais o **tempo de permanência** no nível 1 seja igual a tantos períodos de clock, quanto o número binário H de 4 bits. De mesma forma a duração do pulso de saída no nível baixo é determinado pela via L de 4 bits. A saída Z é de 1 bit. Obs: use portas, FFs e funções MSI.

**Exemplo: Para
L=3 e H=4**



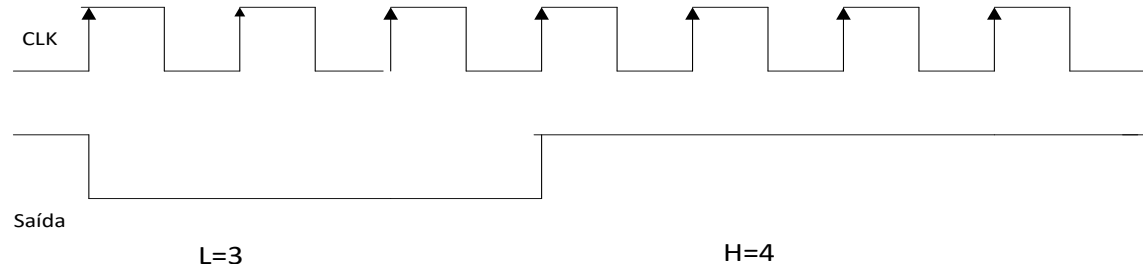
Solução:



Exercícios Resolvidos: Prova P4

7Q: Projetar um **gerador de ondas retangulares** que, acionado por um sinal de clock periódico gere pulsos, nos quais o **tempo de permanência** no nível 1 seja igual a tantos períodos de clock, quanto o número binário H de 4 bits. De mesma forma a duração do pulso de saída no nível baixo é determinado pela via L de 4 bits. A saída Z é de 1 bit. Obs: use portas, FFs e funções MSI.

**Exemplo: Para
L=3 e H=4**



Solução:

