

UNIVERSIDADE DE SÃO PAULO

ESCOLA DE ENGENHARIA DE SÃO CARLOS

PROJETO DE CIRCUITOS INTEGRADOS ANALÓGICOS -  
SEL0618

---

## Projeto 10

---

*Aluno:* Lucas A. M. Magalhães

*Número USP:* 7173347

*Aluno:* Luiz H. G. Patire

*Número USP:* 7126667

11 de agosto de 2014

## Sumário

<b>Relatório</b>	<b>2</b>
Questão 1 . . . . .	2
Questão 2 . . . . .	2
Questão 3 . . . . .	2
Questão 4 . . . . .	2
Questão 5 . . . . .	4
Questão 6 . . . . .	5
Questão 7 . . . . .	7
Questão 8 . . . . .	7
Questão 9 . . . . .	8

---

# Relatório

## Questão 1

Veja os esquemáticos que foram gerados (observe o esquemático com a opção *multi-pages* ou não). Qual é a diferença entre o esquemático associado a EXEMPLAR\_XTR e o esquemático associado a EXEMPLAR.

O esquemático EXEMPLAR\_XTR é o esquemático originalmente gerado pelo programa. O esquemático associado a EXEMPLAR é um esquemático otimizado (nos padrões definidos) e que utiliza células da tecnologia escolhida.

## Questão 2

Verifique o *Critical Path* do esquemático sintetizado e mapeado na tecnologia da AMS. O que significa este *critical path* e como é calculado?

O *Critical Path* mostra o caminho do circuito que possui o maior tempo total de propagação, ou seja, o caminho que limita a máxima frequência em que o circuito pode operar. Ele é calculado com a soma de todos os atrasos possíveis entre a porta de saída e de entrada de dois *Flip-Flops*

## Questão 3

Refaça a otimização alterando as opções de objetivo, área, velocidade, etc. e verifique os resultados. Variando as opções, minimize o *Critical Path*. Qual o valor final obtido e qual a frequência máxima de operação que o circuito sintetizado pode atingir? Apresente no relatório a figura do caminho crítico encontrado.

O atraso final encontrado foi de 3,22 ns, isso produz uma frequência máxima é de 310,5 MHz. A Figura 1 mostra o caminho crítico encontrado pelo "Leonardo Spectrum".

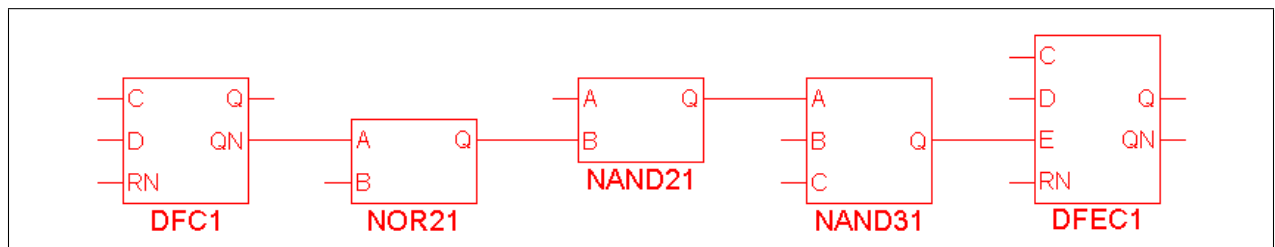


Figura 1: Caminho Crítico Encontrado pelo "Leonardo Spectrum"

## Questão 4

A partir do esquemático (utilize o *viewpoint*) faça a geração do *layout* da **uart**. No roteamento utilize para  $V_{DD}$  e  $V_{SS}$  apenas linhas mais largas do que  $1,8\mu m$ . Para conseguir isso utilize, dentro do menu **ARoute**, a edição dos **net Classes**.

---

Procure nesse circuito fazer o roteamento mas observe que devido ao tamanho, completa-lo é bastante trabalhoso. Após o roteamento passe o LVS e o DRC.

Obs.1: Para se posicionar todas as células da **uart** é necessário se ter as duas páginas do esquemático abertas no **ICStation**. Para isso utilize o comando **\$open\_sheet()**. Posteriormente selecione todas as células de cada esquemático e faça o seu *placement*.

Obs.2: Para terminar *layout* a sugestão é seguir os passos:

1. Faça inicialmente o roteamento das linhas de alimentação ( $V_{DD}$  e  $V_{SS}$ );
2. Execute, sem colocar os *ports*, o roteamento automático do restante dos sinais. Deixe os metais configurados para serem utilizados em apenas uma direção;
3. Quando o número de ligações não feitas estiver em torno de 30, altere a configuração para permitir que os metais sejam utilizados nas duas direções;
4. Agora selecione uma linha de cada vez e mande executar o roteamento automático. Caso a ferramenta não consiga executar um roteamento, apague as ligações que estão atrapalhando, sempre há, ou tente o **RIP**. Não use indiscriminadamente o comando **RIP**, pois algumas vezes ele piora o roteamento;
5. Quando conseguiu realizar todas as ligações execute o LVS (coloque nele a opção “ignorar os ports”);
6. Após o LVS dar resultado correto acrescente os *ports* e termine o roteamento;
7. Passe o DRC e corrija os erros (muitos);
8. Termine o *layout* passando o LVS (agora considerando os *ports*).

Obs.3: o item acima é bastante trabalhoso, mas será um excelente treino para roteamento e DRC.

O *layout* é apresentado na Figura 2.

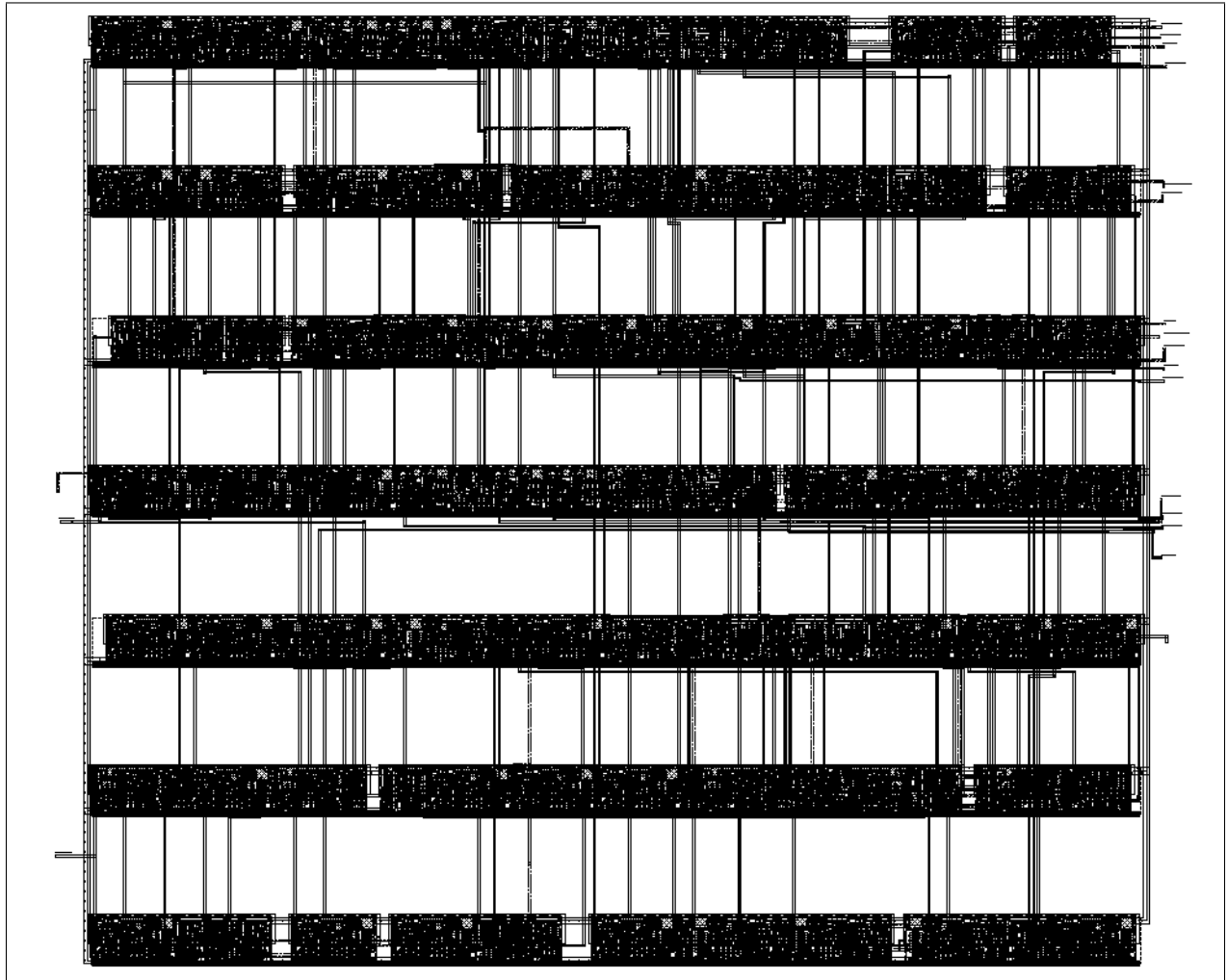


Figura 2: *Layout* Gerado da Uart

### Questão 5

Considere o circuito da Figura 3 (circuito *prescaler*).

O circuito composto pelos blocos hachurados (três D-*flip flops* e duas portas lógicas) compões uma máquina de estados. Considere que:

1. os sinais A, B e C definem o estado da máquina (ex.: o estado 000 é quando  $A=0$ ,  $B=0$  e  $C=0$ );
2. esta máquina tem como entrada o sinal MC que define se o circuito divide o *clock* por 4,  $MC = 1$ , ou por 5,  $MC = 0$ ;
3. a saída é o sinal A.

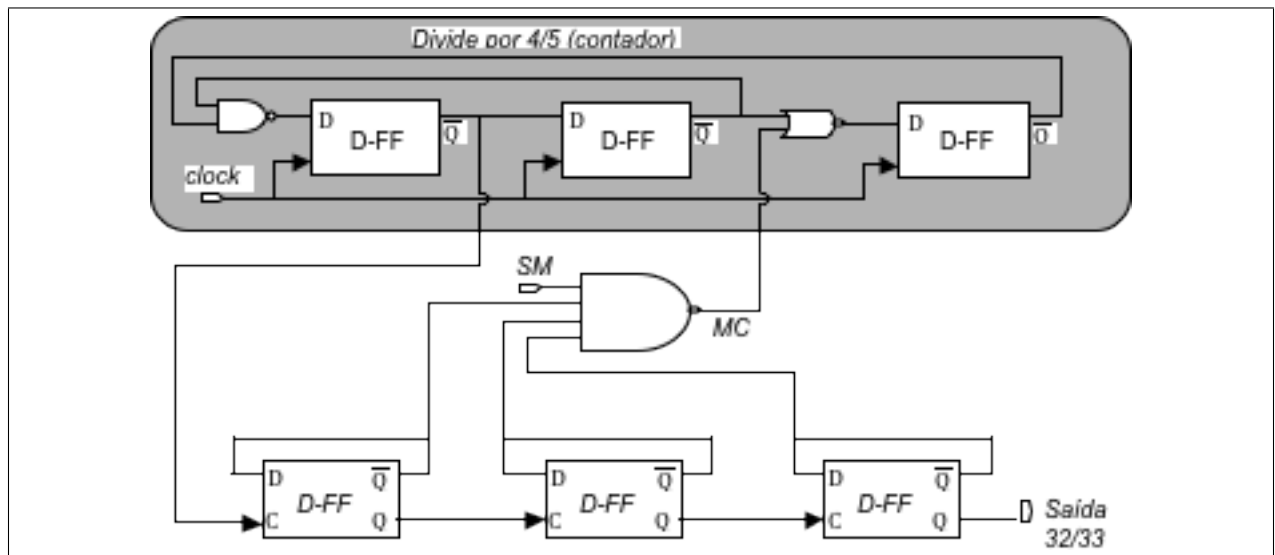


Figura 3: Prescaler 32/33

Desenhe o diagrama de estados que representa o funcionamento do contador 4/5.

O diagrama do contador é apresentado na Figura 4

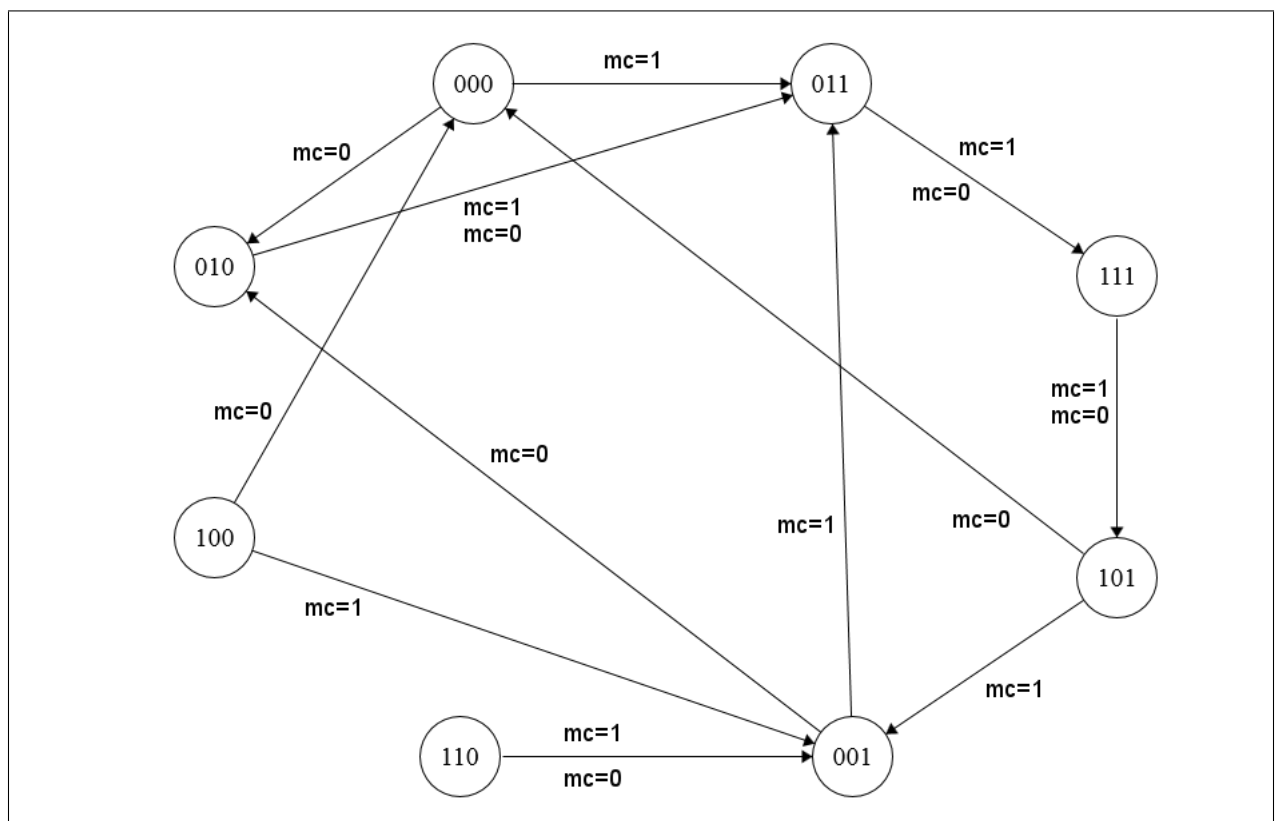


Figura 4: Máquina de Estados do Contador 4/5

## Questão 6

Descreva a máquina de estados em VHDL. Utilize o manual de VHDL do *software* Leonardo (LeonardoSpectrum HDL Synthesis Manual (D:\MGC\LeoSpec\LS2005a\_82\doc\leospec\_hdl.pdf)) para ver modelos de descrição para máquinas de estado. Apresente o VHDL no relatório.

O *Listing* 1 mostra o código VHDL que especifica a máquina de estados mostrada na questão anterior.

Listing 1: "Código em VHDL da Máquina de Estados do Contador 4/5"

```
1  entity state_machine is
      port( clk, mc: IN BIT ;
            qn : OUT BIT) ;
end state_machine;

6  architecture sm of state_machine is
      type state_type is (s0, s1, s2, s3, s4, s5, s6, s7);
      signal present, nt : state_type;

begin
11     update: process ( clk )
        begin
            if clk'event and clk = '1' and clk'last_value = '0' then
                present <= nt;
            end if;
16     end process ;

        states: process (present, mc)
        begin
            case present is
21             when s0 =>
                if (mc = '1') then
                    nt <= s3;
                else
                    nt <= s2;
26             end if;
                qn <= '0';
            when s1 =>
                if (mc = '1') then
                    nt <= s3;
31             else
                    nt <= s2;
                end if;
                qn <= '0';
            when s2 =>
36             nt <= s3;
                qn <= '0';
            when s3 =>
                nt <= s7;
                qn <= '0';
41             when s4 =>
                if (mc = '1') then
                    nt <= s1;
                else
                    nt <= s0;
46             end if;
                qn <= '1';
            when s5 =>
```

```

51         if (mc = '1') then
            nt <= s1;
        else
            nt <= s0;
        end if;
        qn <= '1';
56     when s6 =>
        nt <= s1;
        qn <= '1';
        when s7 =>
            nt <= s5;
            qn <= '1';
61     end case;
end process ;

```

### Questão 7

Feche o Leonardo e abra novamente. Leia o VHDL, mas tome cuidado para utilizar como *Encoding Style* a opção *Binary*. Otimize e veja se o resultado melhorou. Otimize o circuito para obter o menor *Critical Path*. Qual é o valor encontrado? Apresente o esquemático obtido.

O caminho crítico encontrado tem atraso de 0,96 ns, assim a frequência máxima de operação do circuito é de 1,04 GHz. O esquemático é mostrado na Figura 5.

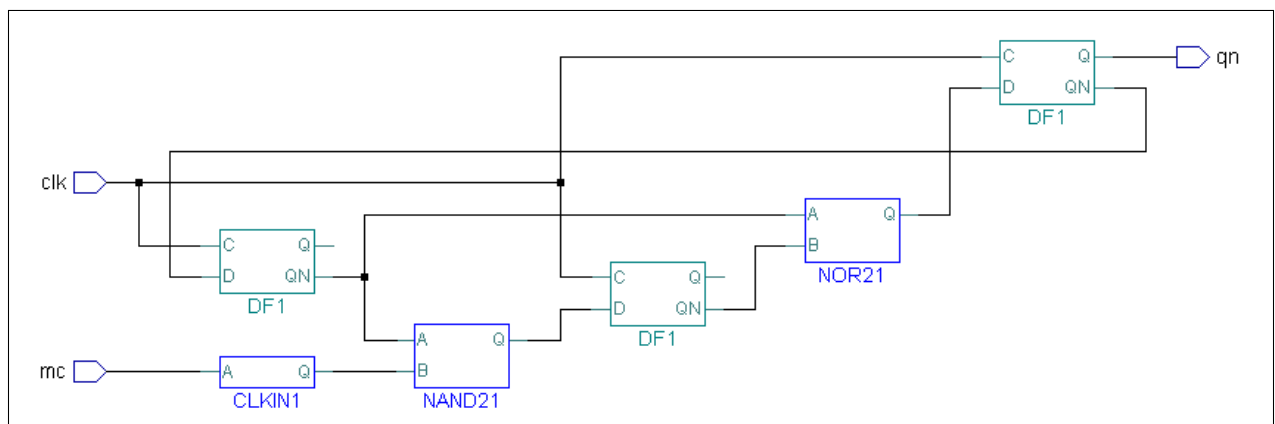


Figura 5: Esquemático do Contador 4/5 criado pelo "Leonardo Spectrum"

### Questão 8

Faça as verificações com DRC e LVS e apresente o *layout* final.

O *layout* do circuito divisor 4/5 é apresentado na figura 6.



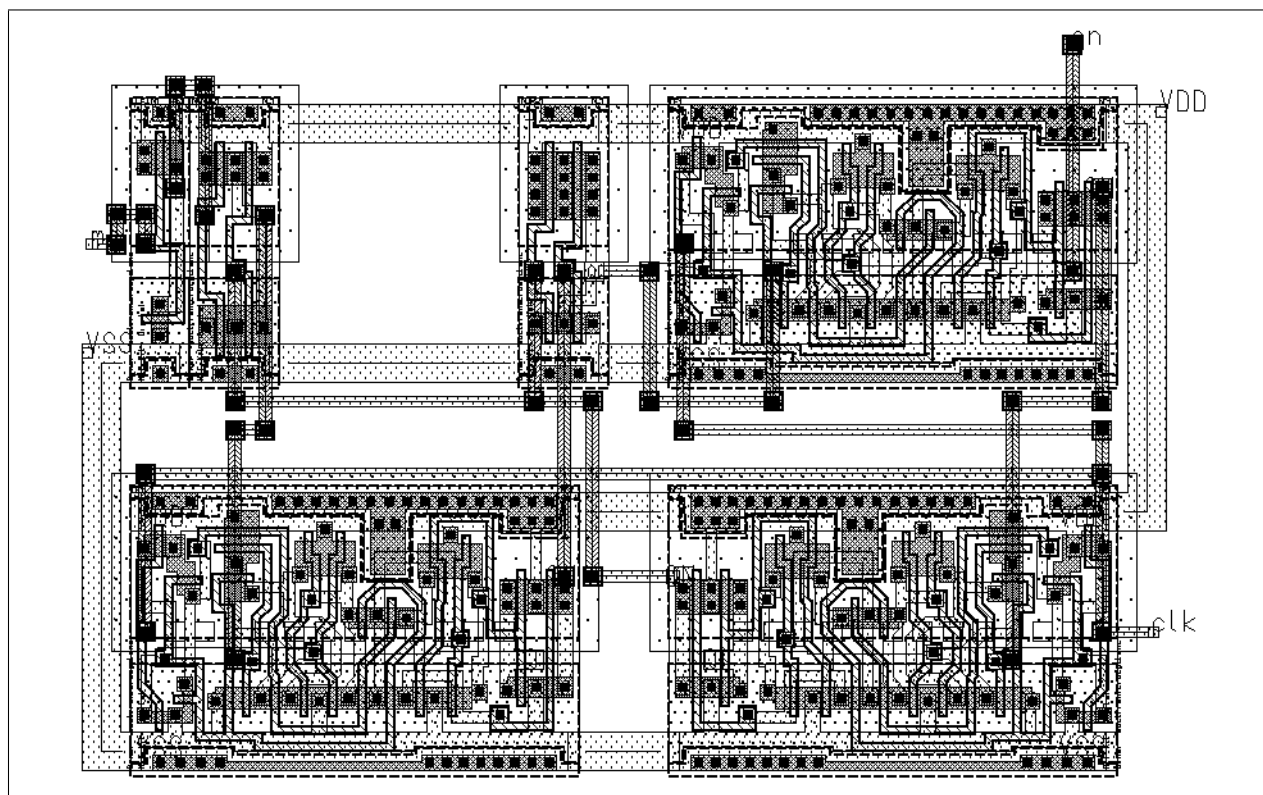


Figura 6: *Layout* do Circuito Divisor 4/5

### Questão 9

Faça a extração do circuito com **R+C+CC** e determine a máxima velocidade que o circuito atinge (teste para MC = “1” e para MC= “0”). Compare o resultado com o encontrado no item 17.

O arquivo padrão para a simulação realizada pode ser encontrada no *Listing 2*. Ele mostra a simulação para MC = '1' (para simular em MC='0', basta mudar a tensão da fonte VM para 0V).

Listing 2: "Arquivo de Simulação do Contador 4/5"

```

*****
3  .include "contador5"

.param f = 500MEG , T='1/f', ts='0.01*T' , td = ts

X1 QN MC CLK VDD VSS STATE_MACHINE

8  VM MC 0 3V
   VD VDD 0 3V
   VS VSS 0 0V
   VCLK CLK 0 PULSE( 0 3V 0 ts td '0.5*T-0.5*ts-0.5*td' T )

13 .tran 0 100n 0 100p SWEEP f 0.7G 1.0G 10MEG
    .meas tran periodo TRIG V(QN) val=1.5 fall=2 targ V(QN) val=1.5 fall=3
    .meas tran outFreq PARAM='1/periodo'

```

```

.meas tran rel PARAM='outFreq/f'
.probe tran V(CLK) V(QN)

*
.include "transistors.mod"
.end

```

Os gráficos de saída das simulações estão na Figura 7 e na Figura 8. A relação entre as frequências foi calculada dividindo a frequência de saída pela frequência de operação de saída.

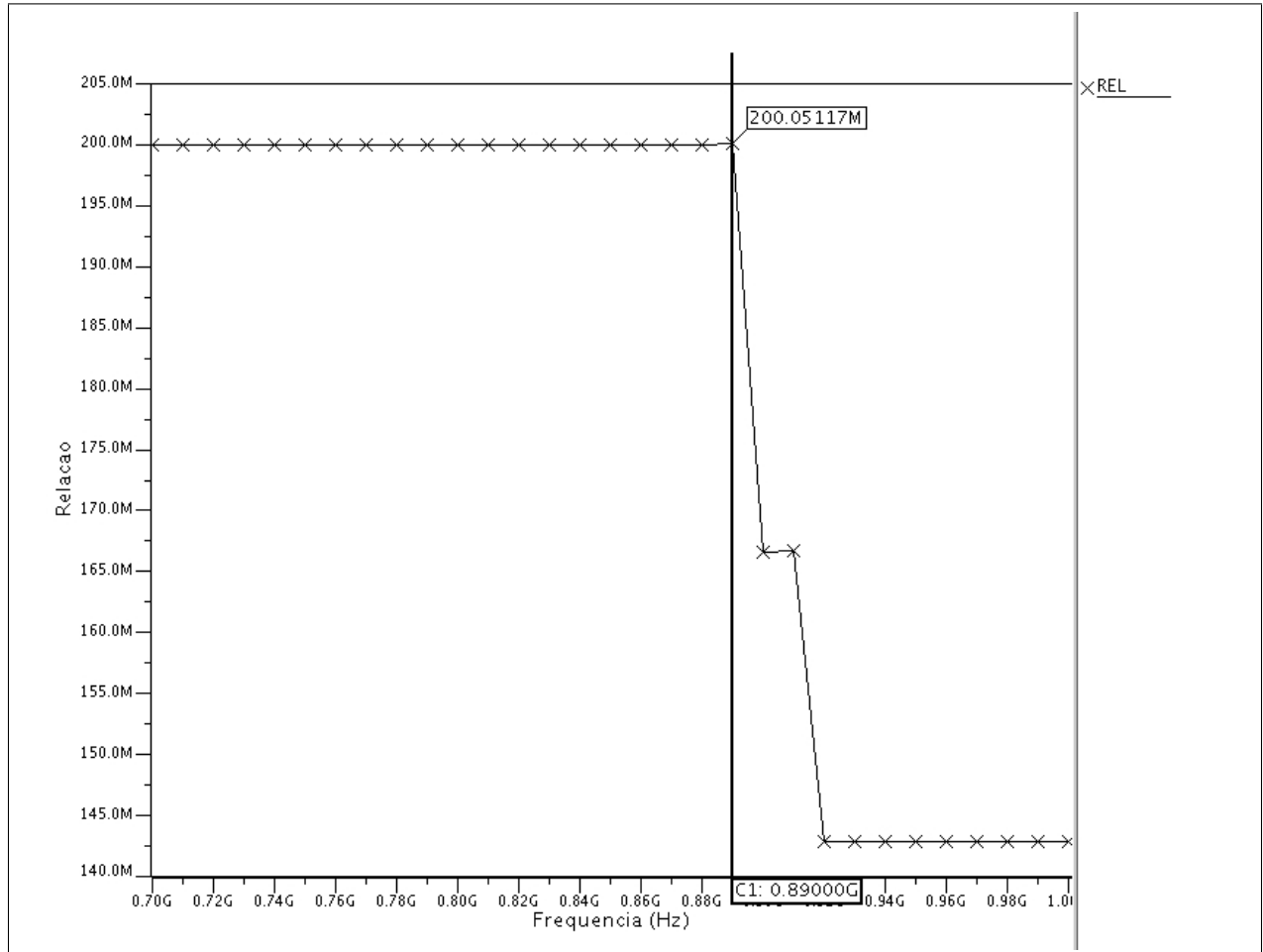


Figura 7: Simulação Contador 4/5 (MC = 0) - Relação entre Frequências x Frequência de Operação

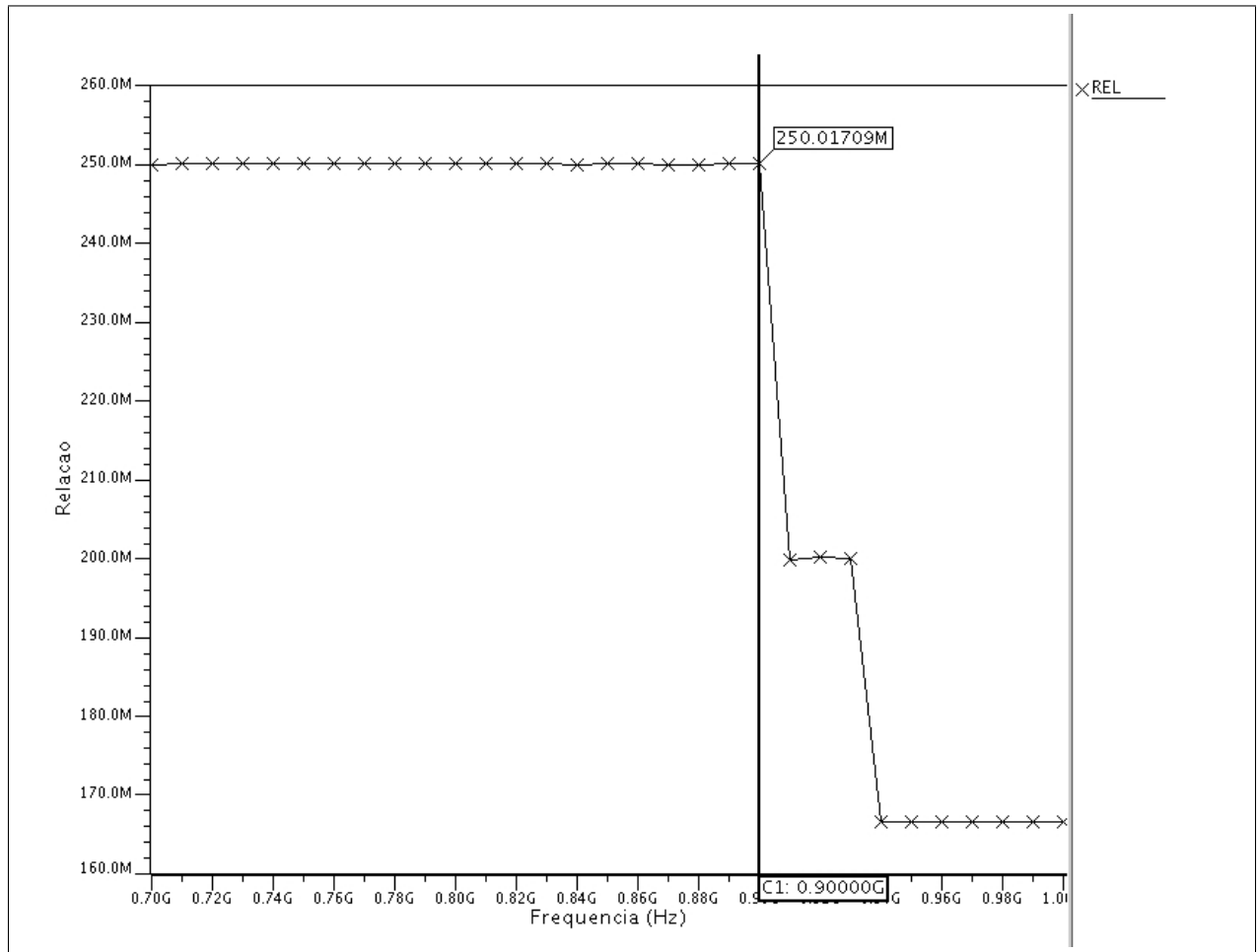


Figura 8: Simulação Contador 4/5 (MC = 1) - Relação entre Frequências x Frequência de Operação

As frequências de operação encontradas podem ser encontradas na Tabela 1:

Tabela 1: Máximas Frequências de Operação

Análise	Frequência
Caminho Crítico - "Leonardo Spectre"	1,04 GHz
Simulação R+C+CC, MC = 0	890 MHz
Simulação R+C+CC, MC = 1	900 MHz

A frequência máxima encontrada pela simulação é significativamente menor do que a frequência encontrada pelo "Leonardo Spectre". Isso se deve ao fato de que a extração R+C+CC (e consequentemente a simulação) leva em consideração elementos parasitas do *layout*, não considerado por análises em alto nível.