

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
INTRODUÇÃO A MICROELETRONICA



RELATÓRIO I

THIAGO ALVES DE ARAUJO
MATRICULA: 2016019787

JOÃO PESSOA
2018

Sumário

1. Spice Opus	02
a. Criação dos arquivos .spi	03
b. Criação dos arquivos .cir	04
2. Constante de tempo	05
3. Tempo de subida e descida	08
4. Tempo de propagação LH e HL	11
5. Corrente no Diodo	14

1 – Spice Opus

O Spice Opus é uma ferramenta para simulação de circuitos em geral. Ele é uma recompilação do código-fonte original de Berkeley para sistemas operacionais Windows e Linux.

Para a simulação aqui demonstrada, foi utilizado a versão do programa para a plataforma Windows que pode ser adquirida gratuitamente o site oficial do programa.

Para o uso dos circuitos descritos nos arquivos de código, foi necessário salvá-los na pasta raiz do programa. Para ter acesso a esta pasta, basta clicar em cima do ícone de atalho do programa com o botão direito do mouse e ir em “Abrir Local do Arquivo”. Com a pasta aberta, basta depositar os arquivos de código com suas devidas extensões e executar o programa normalmente.

1.1 - Criação dos arquivos .spi

Para criar os arquivos .spi foi utilizado o bloco de notas do Windows. Os arquivos com essa extensão são criados para representar os componentes utilizados no circuito (resistores, capacitores, diodos, etc.) e ele que vai possuir informações como nome do componente, nó e valor atribuído a ele.

Para a simulação, foi utilizado os seguintes componentes:

resistencia1.spi

.subckt resistencia1 1 2

R1 1 2 1k

.ends resistencia1

resistencia2.spi

.subckt resistencia2 1 2

R1 1 2 2k

.ends resistencia2

capacitor2.spi

.subckt capacitor2 1 2

C1 1 2 2u

.ends capacitor2

capacitor1.spi

.subckt capacitor1 1 2

C1 1 2 1u

.ends capacitor1

Lembrando que o nome em verde é referente ao nome que o arquivo vai ser salvo. Caso se deseje alterar esse nome, os “includes” no arquivo .cir deve ser alterado também. Os arquivos aqui utilizados estão disponíveis em <https://github.com/thiagoalves98/Spice-Opus>

1.2 - Criação dos arquivos .cir

Analogamente a criação dos arquivos anteriores, o arquivo .cir foi criado com o bloco de notas do Windows e salvo com sua devida extensão. Este é o arquivo responsável por representar o circuito (onde estão os nós, os componentes, os valores e tipos de fontes, etc.).

Para a simulação foi utilizado o seguinte circuito:

circuito_rc.cir

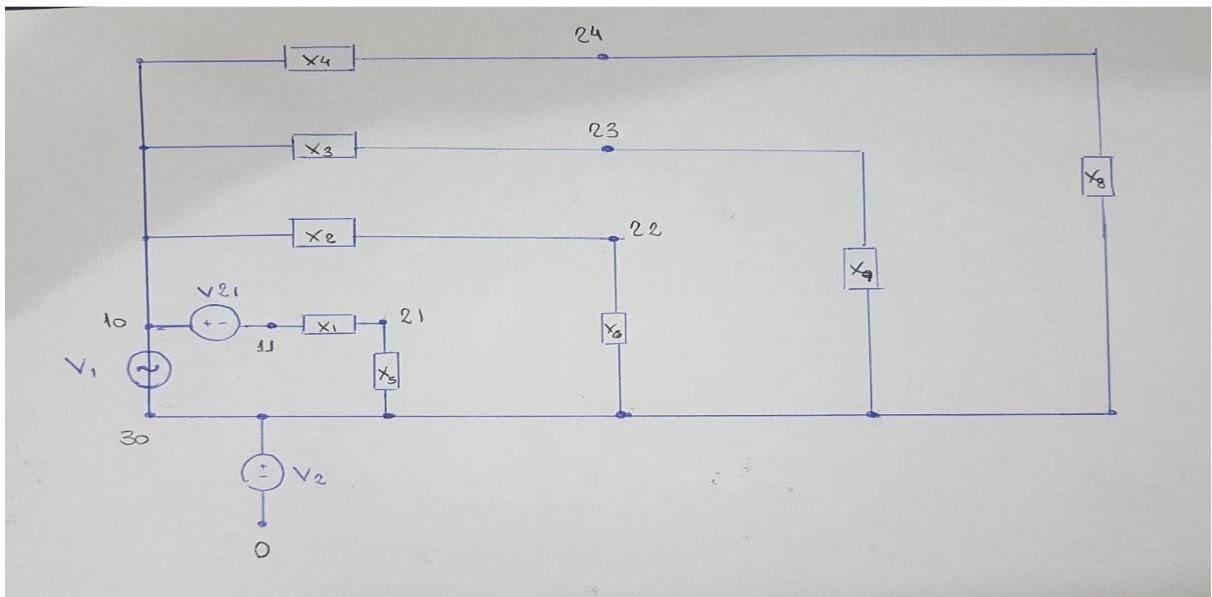
```
.include resistencia1.spi  
.include resistencia2.spi  
.include capacitor1.spi  
.include capacitor2.spi
```

```
x1 11 21 resistencia1  
x2 10 22 resistencia1  
x3 10 23 capacitor1  
x4 10 24 capacitor1  
x5 21 30 capacitor1  
x6 22 30 capacitor2  
x7 23 30 resistencia1  
x8 24 30 resistencia2
```

```
V1 10 30 pulse(-5 5 0 1ns 1ns 10ms 20ms)  
v21 10 11 0v  
V2 30 0 0V
```

```
.tran 0.0001ms 20ms  
.end
```

O circuito equivalente a este código está representado na imagem abaixo



Após isso, para rodar o código, basta abrir o Spice Opus e digitar os seguintes comandos para que seu circuito esteja pronto para ser utilizado

“circuito_rc.cir”

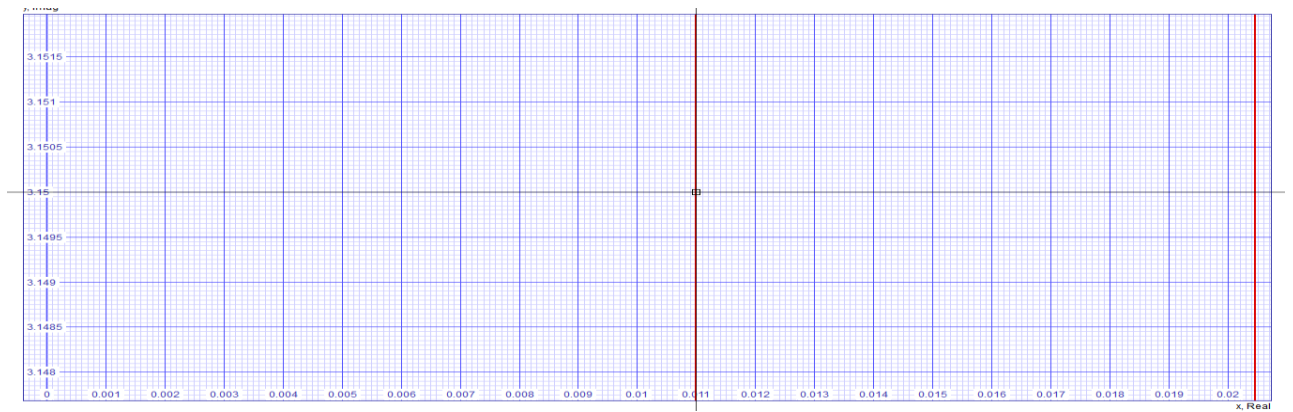
“run”

2 – Constante de tempo

A constante de tempo de um circuito RC é o intervalo de tempo necessário tanto para a carga do capacitor via resistor R até 63,2% da carga total como para a descarga até 37,8% da carga. Para o circuito simulado, segue as seguintes análises

“R1 x C2”



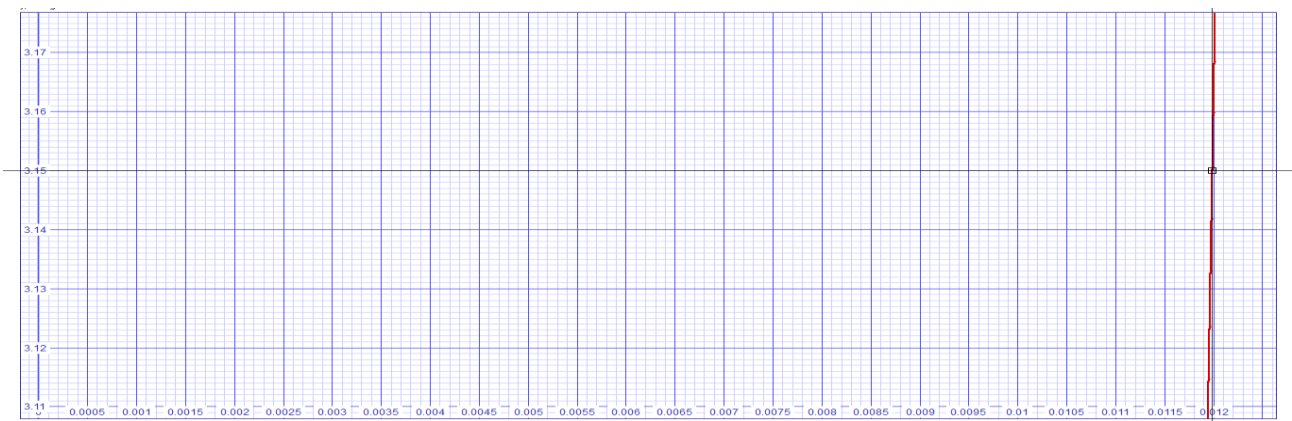


```

Press <space> to identify nearest curve.
x-y grid displaying real vs default.
Marker: x =1.099276832535010e-02   y =3.149998847520531e+00
Cursor: x =1.099276832535010e-02   y =3.149998847520531e+00
Delta : dx=0.000000000000000e+00   dy=0.000000000000000e+00
Slope : dy/dx=+/-INF      dx/dy=+/-INF
  
```

Fazendo “plot v(21)” podemos observar que a constante de tempo esta aproximadamente em 10,1ms (milissegundos)

“R2 x C2”



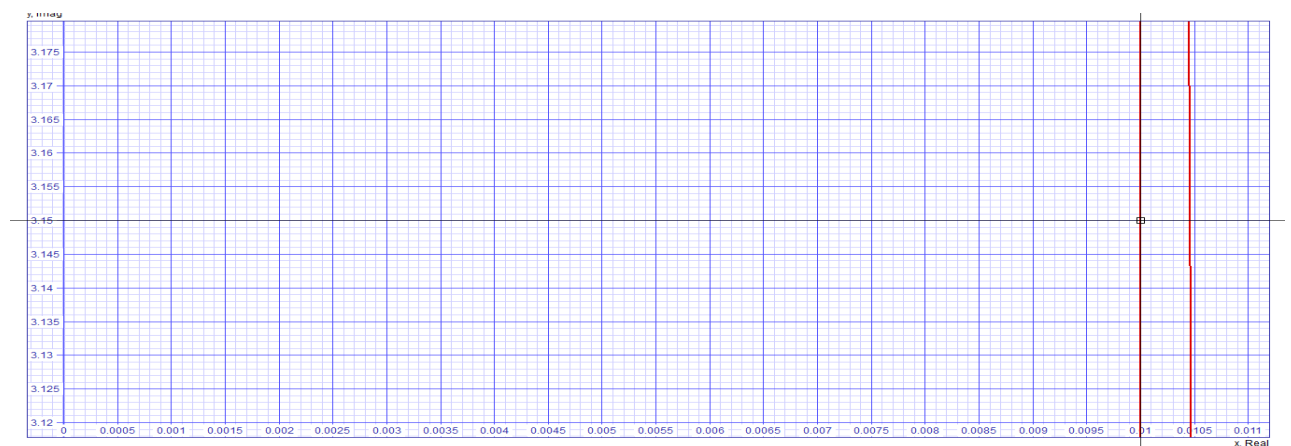
```

Press <space> to identify nearest curve.
x-y grid displaying real vs default.
Marker: x =1.198423718442090e-02   y =3.150012448548824e+00
Cursor: x =1.198423718442090e-02   y =3.150012448548824e+00
Delta : dx=0.000000000000000e+00   dy=0.000000000000000e+00
Slope : dy/dx=+/-INF               dx/dy=+/-INF

```

Exibindo o gráfico do circuito com “plot v(22)” observamos que a constante de tempo é aproximadamente 11,9ms (milissegundos)

“R1 x C1”



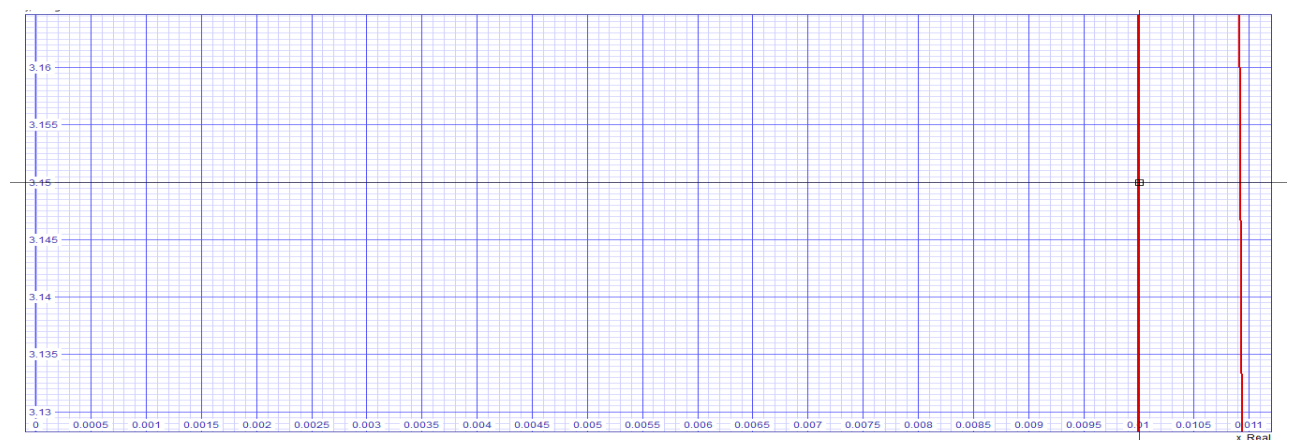
```

Press <space> to identify nearest curve.
x-y grid displaying real vs default.
Marker: x =1.000043738928550e-02   y =3.150009842907537e+00
Cursor: x =1.000043738928550e-02   y =3.150009842907537e+00
Delta : dx=0.000000000000000e+00   dy=0.000000000000000e+00
Slope : dy/dx=+/-INF               dx/dy=+/-INF

```

Exibindo o gráfico do circuito com “plot v(23)” observamos que a constante de tempo é aproximadamente 10,0ms (milissegundos)

“R2 x C1”



```

Press <space> to identify nearest curve.
x-y grid displaying real vs default.
Marker: x =1.000481820057597e-02  y =3.149978464950028e+00
Cursor: x =1.000481820057597e-02  y =3.149978464950028e+00
Delta : dx=0.000000000000000e+00  dy=0.000000000000000e+00
Slope : dy/dx=+/-INF      dx/dy=+/-INF

```

Exibindo o gráfico do circuito com “plot v(24)” observamos que a constante de tempo também é aproximadamente 10,0ms (milissegundos)

3 – Tempo de subida e tempo de descida

Após a simulação podemos obter o tempo de subida e de descida de cada um dos circuitos. Para calcular o tempo de subida, analisaremos o tempo gasto de 10% à 90% da variação máxima de tensão (0% ~ 100%) e para calcular o tempo de descida, fazemos o inverso, analisemos o tempo gasto de 90% à 10%.

Abaixo estão os gráficos com os valores exatos.

Tempo de subida de v(21)



Press <space> to identify nearest curve.
 x-y grid displaying real vs default.
 Marker: x = 1.009593679458239e-02 y = 4.957401988110985e-01
 Cursor: x = 1.228103837471783e-02 y = 4.504032728292289e+00
 Delta : dx=2.185101580135441e-03 dy=4.008292531481191e+00
 Slope : dy/dx=1.834373544882403e+03 dx/dy=5.451452365249343e-04

Tempo de descida de v(21)



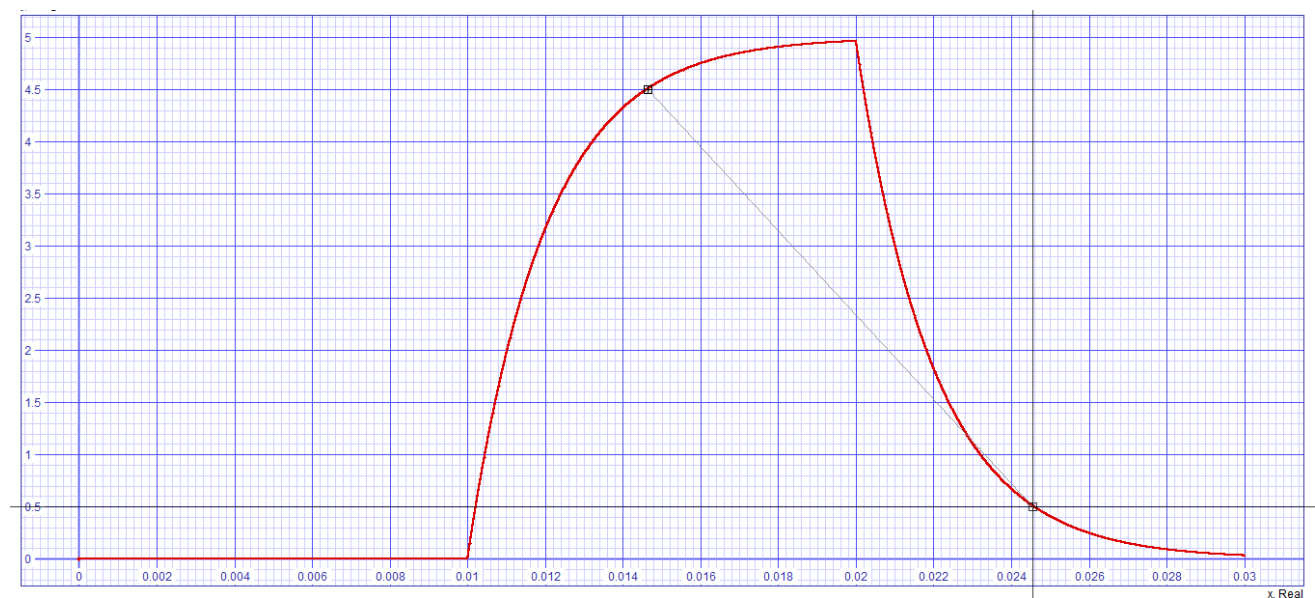
Press <space> to identify nearest curve.
 x-y grid displaying real vs default.
 Marker: x = 1.233069977426637e-02 y = 4.504032728292289e+00
 Cursor: x = 2.231284108352144e-02 y = 4.957401988110985e-01
 Delta : dx=9.981941309255079e-03 dy=-4.008292531481191e+00
 Slope : dy/dx=-4.015544078349542e+02 dx/dy=-2.490322557761630e-03

Tempo de subida de v(22)



Press <space> to identify nearest curve.
 x-y grid displaying real vs default.
 Marker: x =1.463995485327314e-02 y =4.501680580439626e+00
 Cursor: x =1.019525959367946e-02 y =5.016809472591655e-01
 Delta : dx=-4.444695259593678e-03 dy=-3.999979633180461e+00
 Slope : dy/dx=8.999448305225721e+02 dx/dy=1.111179472696369e-03

Tempo de descida de v(22)

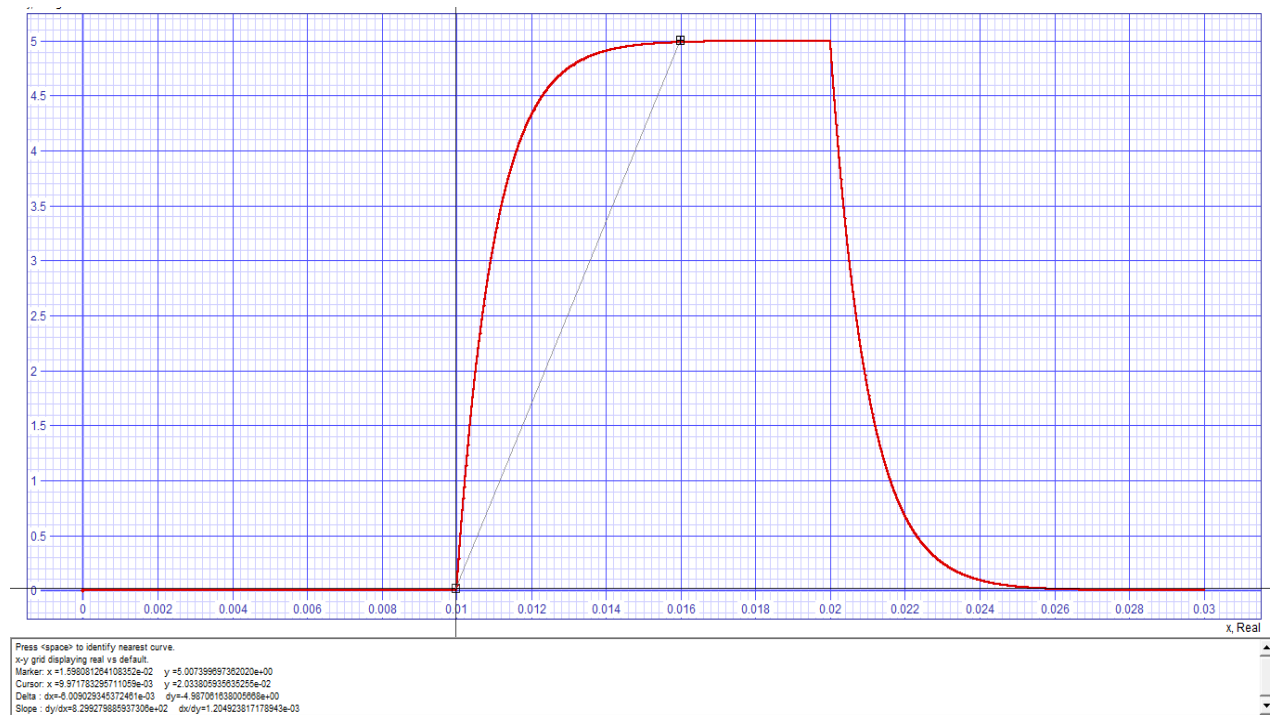


Press <space> to identify nearest curve.
 x-y grid displaying real vs default.
 Marker: x =1.463995485327314e-02 y =4.501680580439626e+00
 Cursor: x =2.457223476297968e-02 y =5.016809472591655e-01
 Delta : dx=9.932279909706548e-03 dy=-3.999979633180461e+00
 Slope : dy/dx=-4.027252221588508e+02 dx/dy=-2.483082620550546e-03

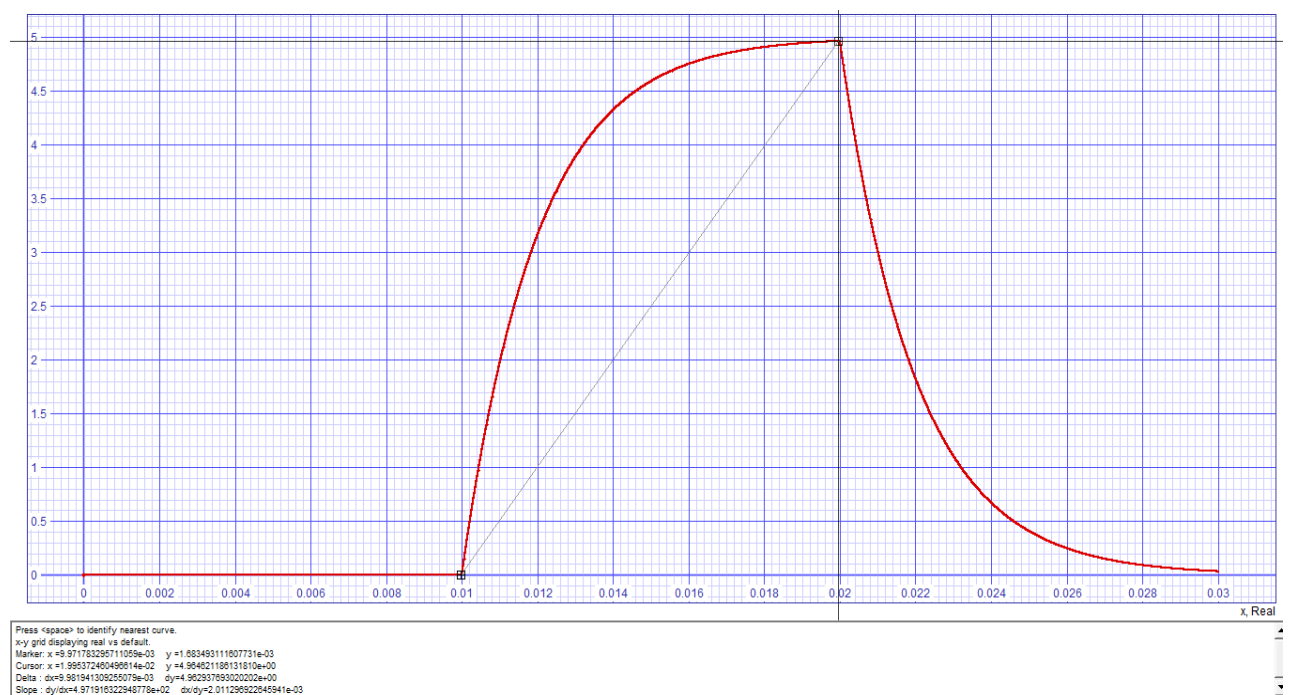
4 – Tempo de propagação LH e HL

É o tempo que determinado circuito leva para atingir o valor médio de toda a tensão que este recebe. As variações são de Low-High (Lo-Hi) e High-Low (Hi-Lo).

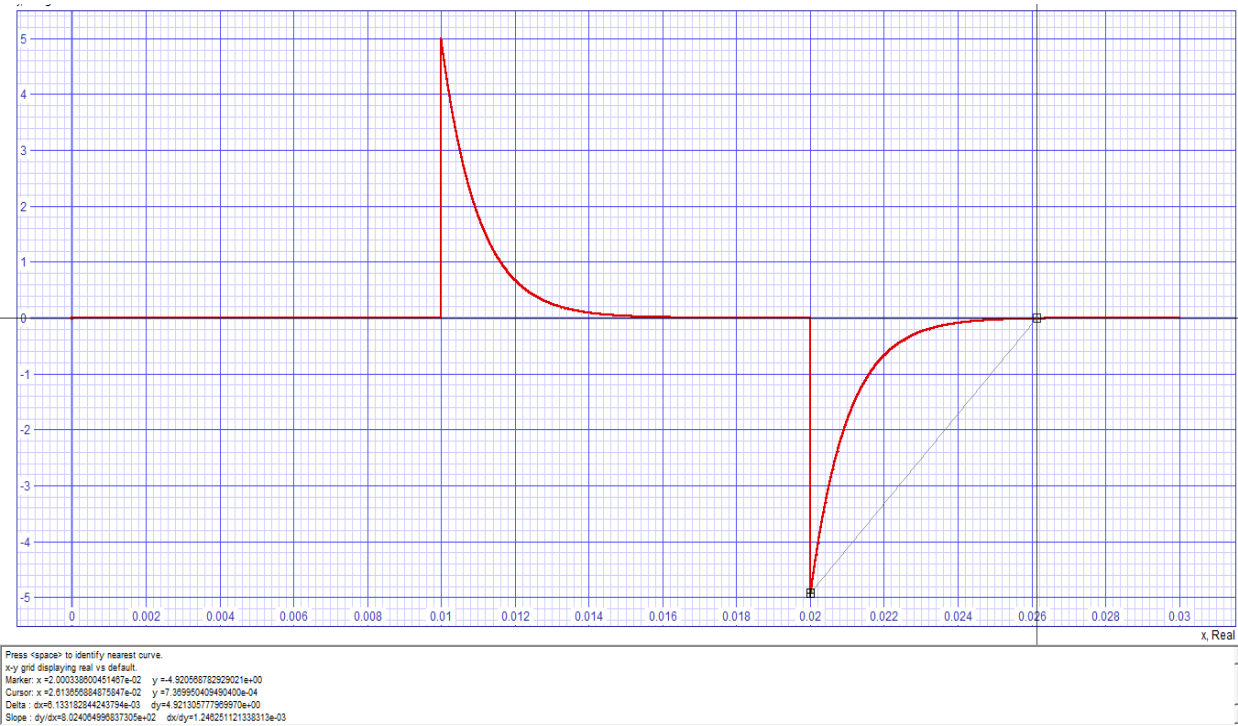
LH de v(21)



LH de v(22)



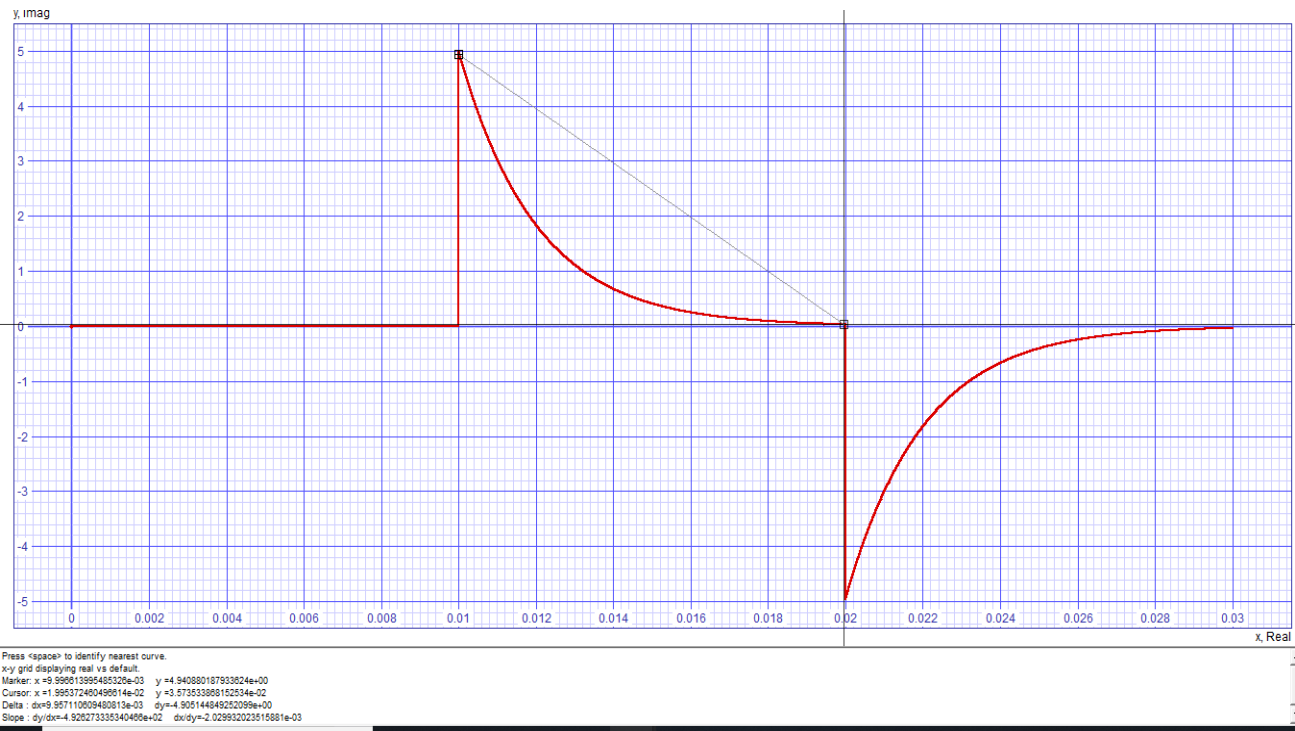
HL de v(23)



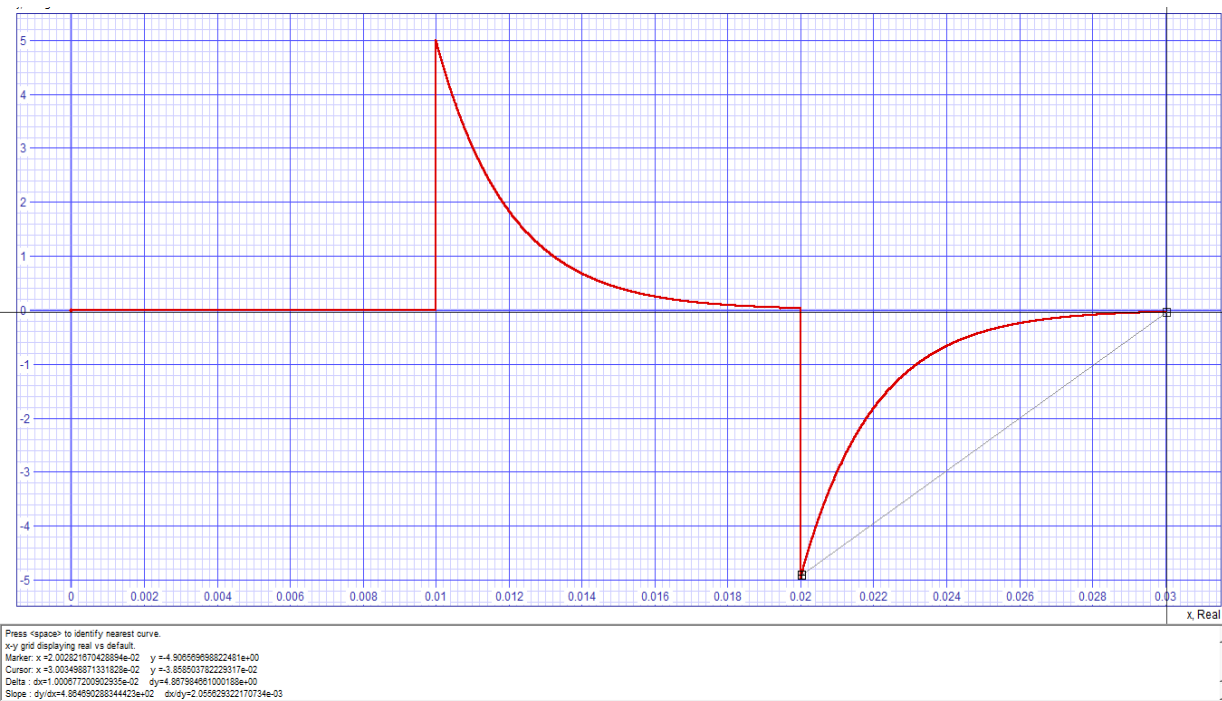
LH de v(23)



HL de $v(24)$



LH de $v(24)$



5 – Corrente no diodo

