1.1 实验一 CMOS 反相器

一、 实验要求

设置反相器晶体管尺寸、计算、测试开关阈值 V_M、测试 VTC 测试、分析计算高电平噪声容限、低电平噪声容限、并完成测试。

二、 实验目的

1、了解 virtuoso 的使用；

2、了解反相器的搭建方法；

3、了解开关阈值、VTC、高低电平噪声容限的定义及计算方法。

三、 实验内容

1、设计 CMOS 反相器的尺寸，NMOS 的宽度为 120nm 不变， 使开关阈值 VM 分别为 0.63V，0.55V，0.52V，给出仿真结果和尺寸；

2、若 CMOS 反相器的尺寸为 PMOS=330nm/40nm，NMOS=280nm/40nm，试通过 DC 仿真的结果计算噪声容限 NML 及 NMH；

3、若 CMOS 反相器的尺寸为 PMOS=330nm/40nm，NMOS=280nm/40nm，通过仿真，求得 tr，tf，tphl，tplh？若 L=40nm 且 NMOS 宽度为 120nm，设计 CMOS 反相器尺寸，使 tphl=tplh， 给出仿真结果和尺寸；

4、若 CMOS 反相器的尺寸为 PMOS=330nm/40nm，NMOS=280nm/40nm,VDD=1.1V，Vin 与内容 3 一致，通过仿真,求 CMOS 反相器 0-20ns 的总能耗;

四、 实验步骤

4.1 阈值电压计算

为高效地找到 W_p 与 V_M 的对应关系，我们采用了**参数扫描**与**表达式自动提取**相结合的方法，避免了手动读取 VTC 曲线交点的繁琐和误差。

(1) 设置 DC 扫描：

- 在 Analyses 中选择 dc 仿真。
- Sweep Variable 选择 Component Parameter。
- Sweep Range 设置为从 0 到 1.1，Sweep Type 设为 Auto。

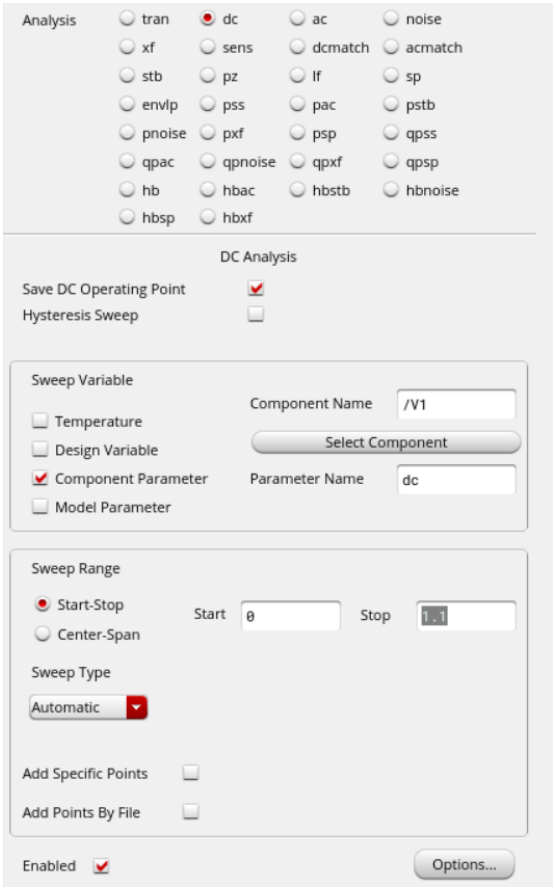


图 1　dc 仿真参数设置

(2) 设置参数扫描：

- 打开 Tools -> Parametric Analysis。
- 在 Variable 处填入变量名 wp。
- Sweep Range 设置一个较宽的“粗扫”范围：从 120n 到 5u，步数设为 40。
- 完成参数粗扫实验。

Variable	Value	Sweep?	Range Type	From	To	Step Mode	Total Steps	Inclusion List	Exclusion List
wp		☑	From/To	120n	5u	Auto	40		

图 2　粗扫参数设置

(3) 设置V_M自动提取表达式：

1. 在图像显示窗口，选择 Tools -> Calculator.。
2. 在 Calculator 中构建一个表达式，用于自动计算 V_{in} 和 V_{out} 的交点。由于 vt() 函数在 dc 扫描中存在上下文读取陷阱（似乎会出现在 dc 仿真中无法读取瞬时电压的报错），我们使用 getData() 函数来明确指定读取 dc 仿真的结果：

```
cross(getData("/out" ?result "dc") - getData("/in" ?result "dc") 0 1 "either")
```

参数说明：

- getData("/out" ?result "dc")：强制从 dc 仿真结果中提取 /out 节点的电压波形。
- getData("/in" ?result "dc")：强制从 dc 仿真结果中提取 /in 节点的电压波形。
- cross(...)：查找 (Vout - Vin) 波形穿过 0 点（即$V_{out}=V_{in}$）时的横坐标值（即V_M）。

3. 将表达式命名为 VM，并确保它被勾选以便于绘制。

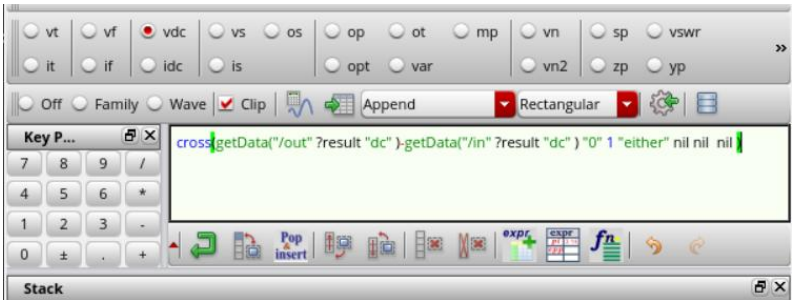


图 3　VM 公式设置

(4) 绘制 V_M-w_p 曲线&提取答案

1. 根据不同的 w_p 的数值在曲线中找到对应的 V_M 值。

4.2 噪声容限计算

V_{OH} 和 V_{OL} 是通过 value() 函数在 dc 扫描的端点处（$V_{in}=0$和$V_{in}=V_{dd}$）对输出波形 V(out) 进行取值得到的。

VOH：获取 /out 波形在 $V_{in}=0$ 时的 Y 值。

```
value(getData("/out" ?result "dc") 0)
```

VOL：获取 /out 波形在 $V_{in}=1.1$ （V_{dd}）时的 Y 值。

```
value(getData("/out" ?result "dc") 1.2)
```

VIL：找到增益曲线（deriv(Vout)）第 1 次下降穿过 y = -1 时的 x 坐标值。

```
cross(deriv(getData("/out" ?result "dc")) -1 1 "falling")
```

VIH：找到增益曲线（deriv(Vout)）第 1 次上升穿过 y = -1 时的 x 坐标值。

```
cross(deriv(getData("/out" ?result "dc")) -1 1 "rising")
```

通过直接引用上述表达式，我们可以让 ADE L 自动完成减法运算，直接输出最终结果：

NML（低电平噪声容限）： $N_{M_L} = V_{IL} - V_{OL}$

```
 cross(deriv(getData("/out" ?result "dc")) -1 1 "falling") -
value(getData("/out" ?result "dc") 1.2)
```

NMH（高电平噪声容限）： $N_{M_H} = V_{OH} - V_{IH}$

```
  value(getData("/out" ?result "dc") 0) - cross(deriv(getData("/out" ?result
"dc")) -1 1 "rising")
```

运行 dc 仿真后，ADE L 的 Outputs 窗口将直接显示 NML 和 NMH 的计算结果，无需手动绘图和测量。

4.3 延迟时间计算 W_p 参数

子实验一：延迟时间测量

在输出端接一个负载电容C_L（$50fF$）至地。

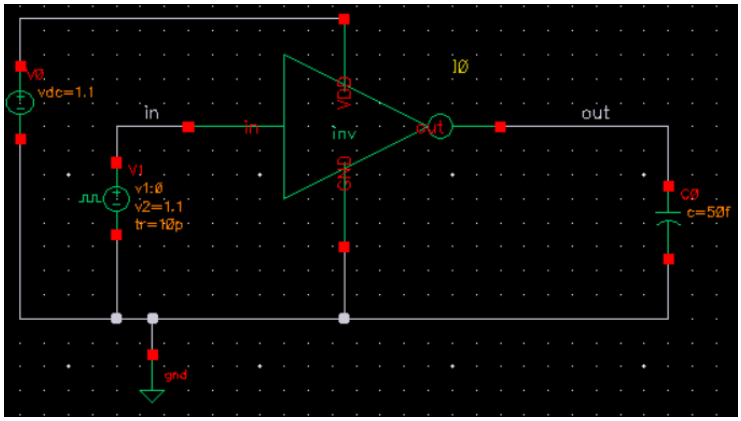


图 4　测试电路图

在输出图像的 Tools -> Calculator 窗口中，添加以下四个表达式：

输出名称	表达式
t_{phl}	delay(getData("/in" ?result "tran") 0.55 1 "rising" getData("/out" ?result "tran") 0.55 1 "falling")
t_{plh}	delay(getData("/in" ?result "tran") 0.55 1 "falling" getData("/out" ?result "tran") 0.55 1 "rising")
t_r	riseTime(getData("/out" ?result "tran") 0.11 0.99 1 "rising")

t_f	fallTime(getData("/out" ?result "tran") 0.99 0.11 1 "falling")

子实验二：W_p 尺寸优化（$t_{phl} = t_{plh}$）

设置设计变量：将 PMOS 宽度 W_p 设置为变量 wp。

设置参数扫描：

- 启用 Parametric Analysis，Variable 为 wp。
- **粗扫描**范围：从 $120n$ 到$600n$，步数20。（比例系数从 1 到 5）

设置目标函数：

- 目标是找到 $t_{phl} - t_{plh} = 0$ 时的 W_p 值。
- 在 Tools -> Calculator 中，添加一个表达式:

输出	表达式
DELAY_DIFF	delay(getData("/in" ?result "tran") 0.6 1 "rising" getData("/out" ?result "tran") 0.6 1 "falling") - delay(getData("/in" ?result "tran") 0.6 1 "falling" getData("/out" ?result "tran") 0.6 1 "rising")

4.4 功耗计算

在输出图像的 Tools -> Calculator 窗口中，添加以下表达式，用于直接计算总能耗 E_{total}。

输出	表达式
E_{total}	integ(1.1 * (-IT("/V0/plus")) 0 20n)

五、 实验结果

5.1 阈值电压计算

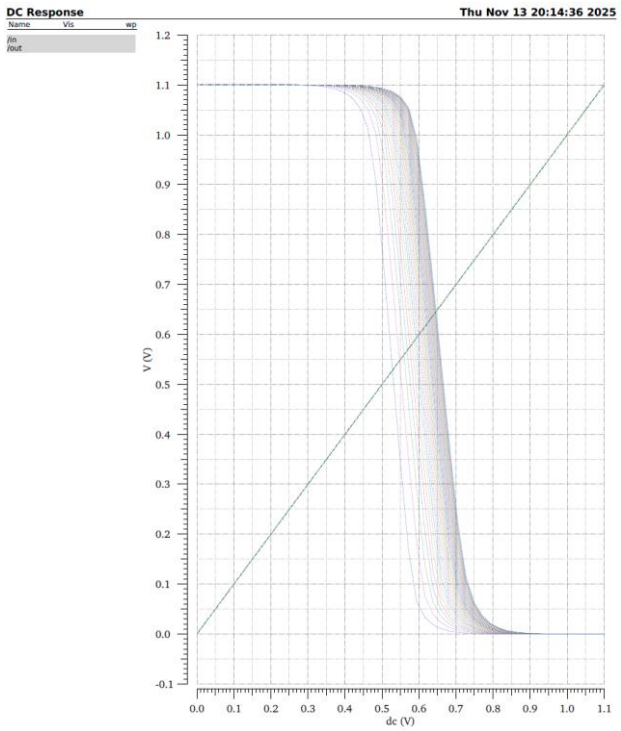


图 5　w_p 参数扫描曲线

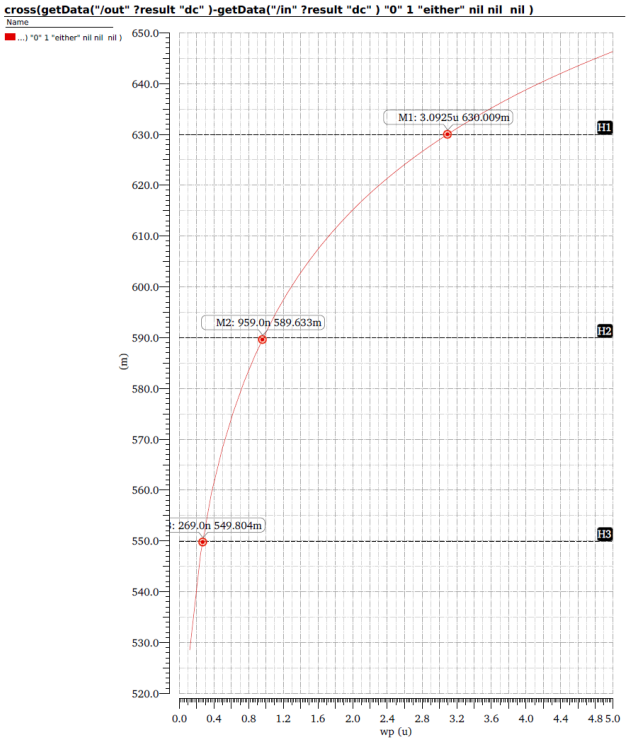


图 6　$V_M - w_p$ 曲线（结果在图上有标注）

5.2 噪声容限计算

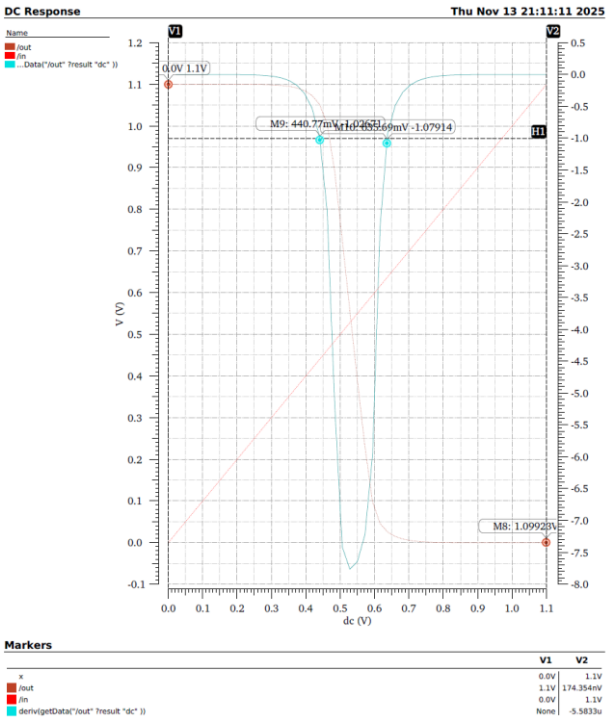


图 6　dc 仿真曲线 & deriv 曲线

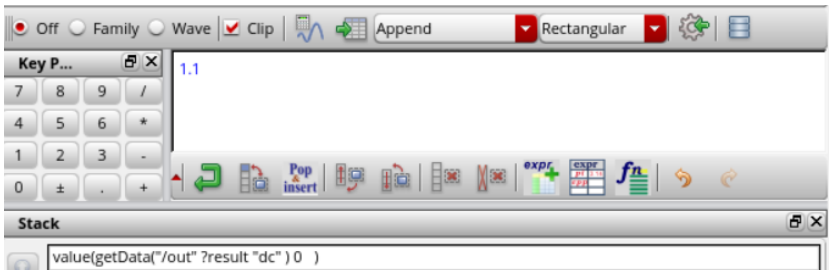


图 7 V_{OH} 计算(公式在 Stack 的顶层)

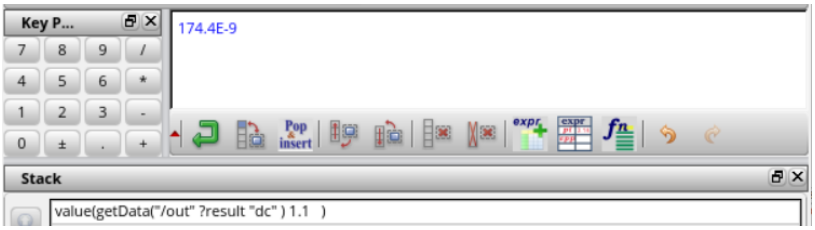


图 8　V_{OL} 计算(公式在 Stack 的顶层)

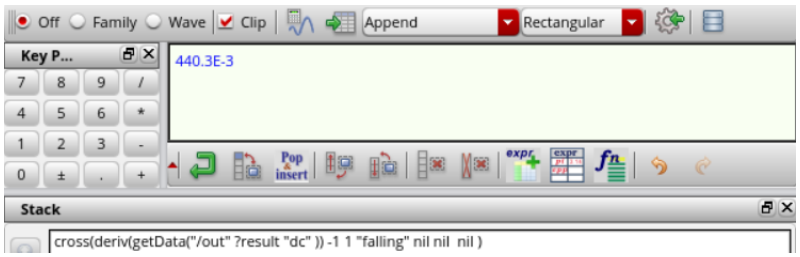


图 9　V_{IL} 计算(公式在 Stack 的顶层)

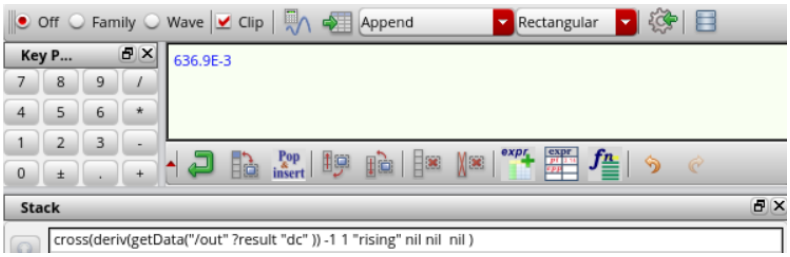


图 10　V_{IH} 计算(公式在 Stack 的顶层)

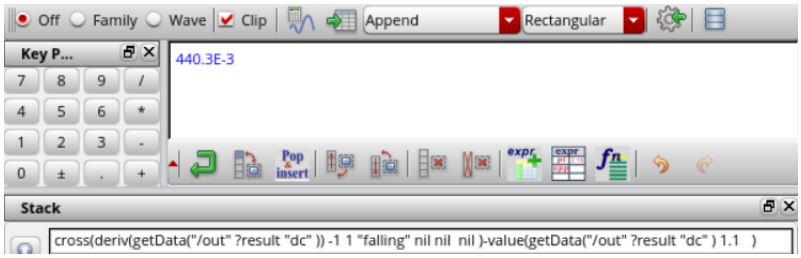


图 11　N_{ML} 计算(公式在 Stack 的顶层)

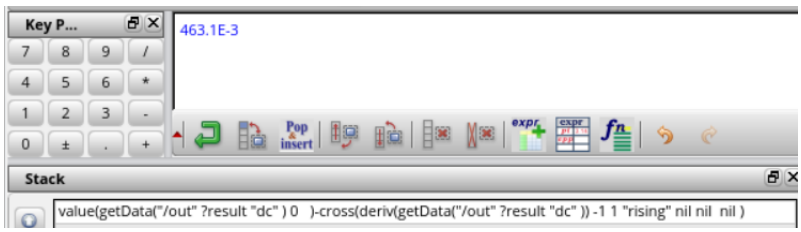


图 12　N_{MH} 计算(公式在 Stack 的顶层)

噪声容限	预期结果	仿真结果	绝对差异 (MV)	差异率
N_{ML}	$0.440V$	$0.4403V$	$0.3mV$	$\approx 0.07\%$
N_{MH}	$0.460V$	$0.4631V$	$3.1mV$	$\approx 0.67\%$

结果的精确度基本达到工程分析的要求。微小的差异是**仿真工具的数值精度**高于**理论公式精度**的体现。

5.3 延迟时间计算

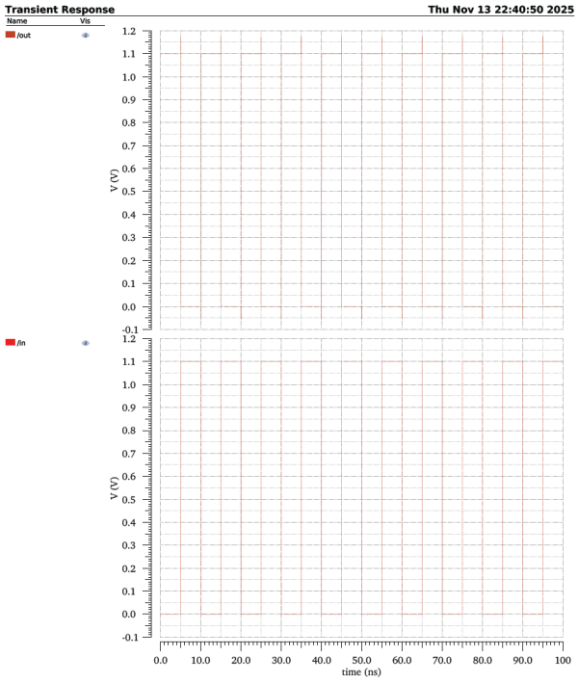
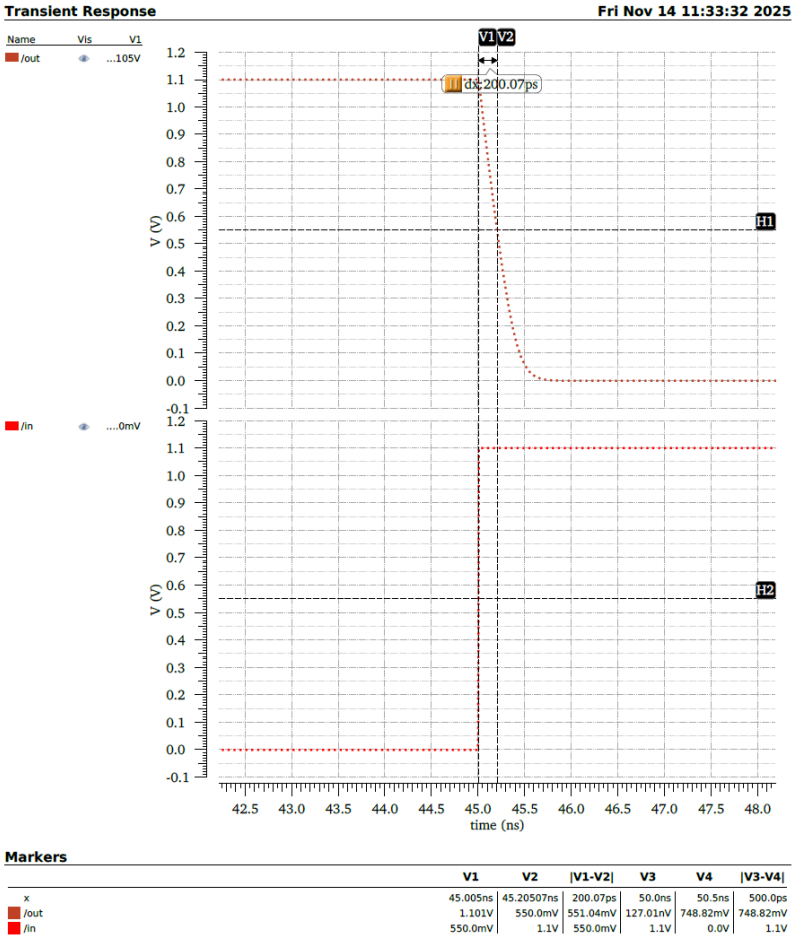


图 13　Trans 仿真结果



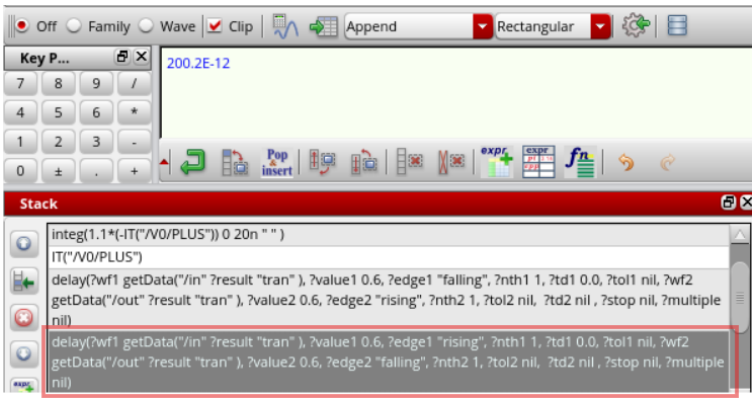
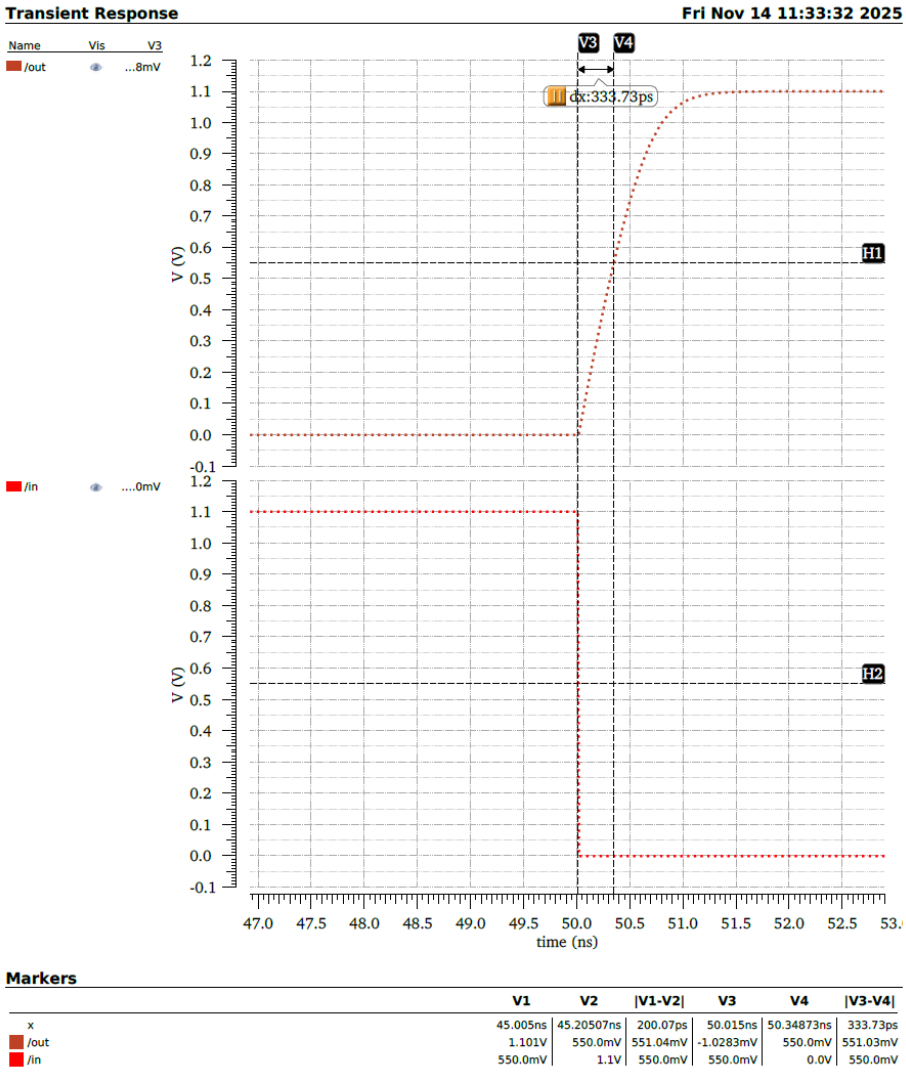


图 14　T_{phl} 计算(手动标注 & 软件计算，公式为红色标注框)



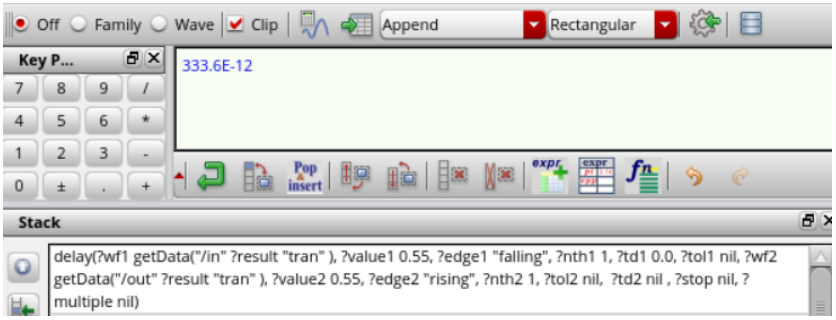
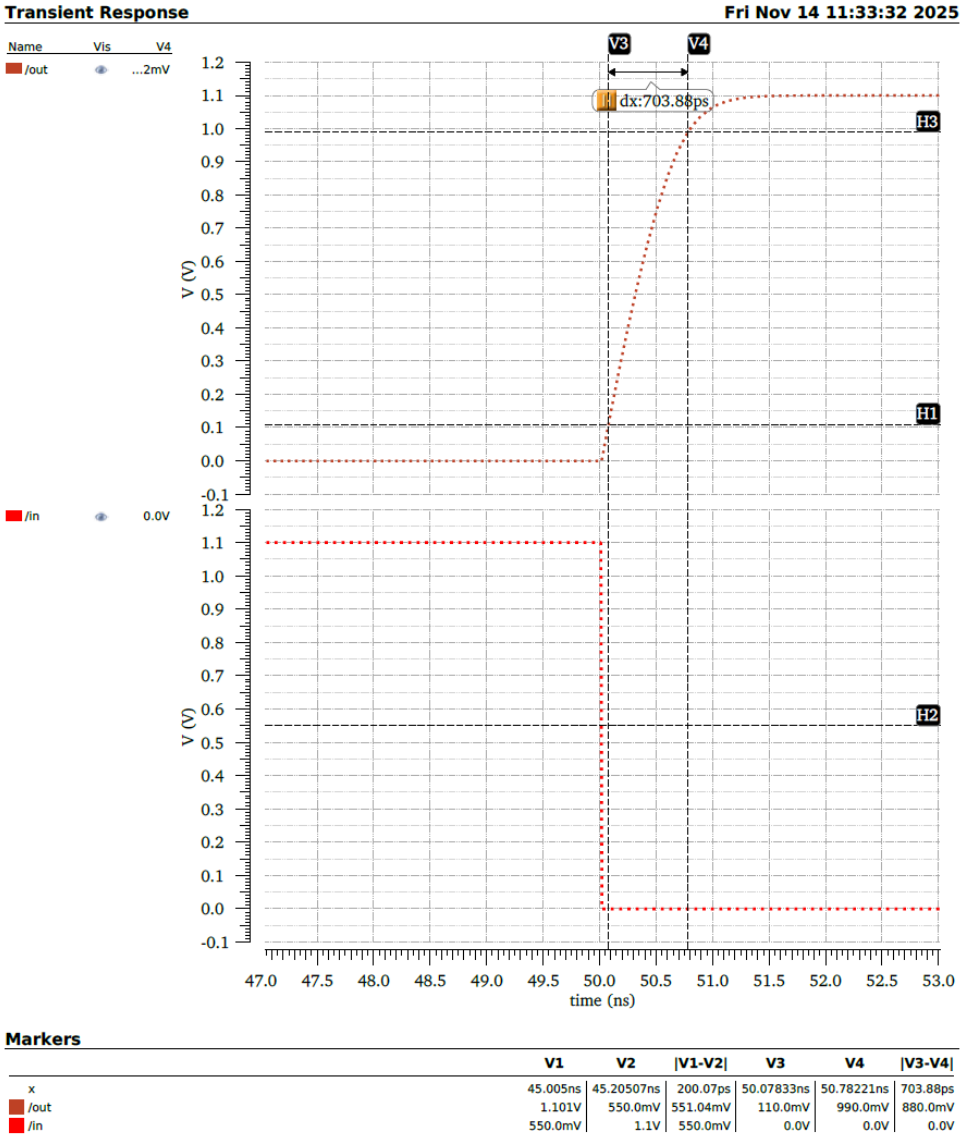


图 15　T_{plh} 计算(手动标注 & 软件计算，公式在 Stack 顶层)



Markers

	V1	V2	|V1-V2|	V3	V4	|V3-V4|
x	45.005ns	45.20507ns	200.07ps	50.07833ns	50.78221ns	703.88ps
/out	1.101V	550.0mV	551.04mV	110.0mV	990.0mV	880.0mV
/in	550.0mV	1.1V	550.0mV	0.0V	0.0V	0.0V

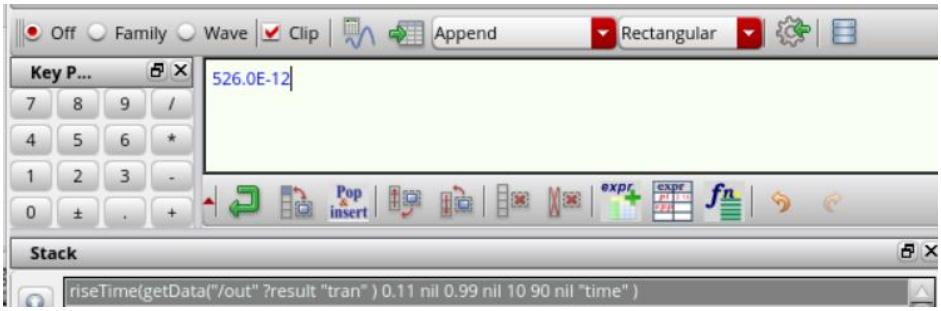
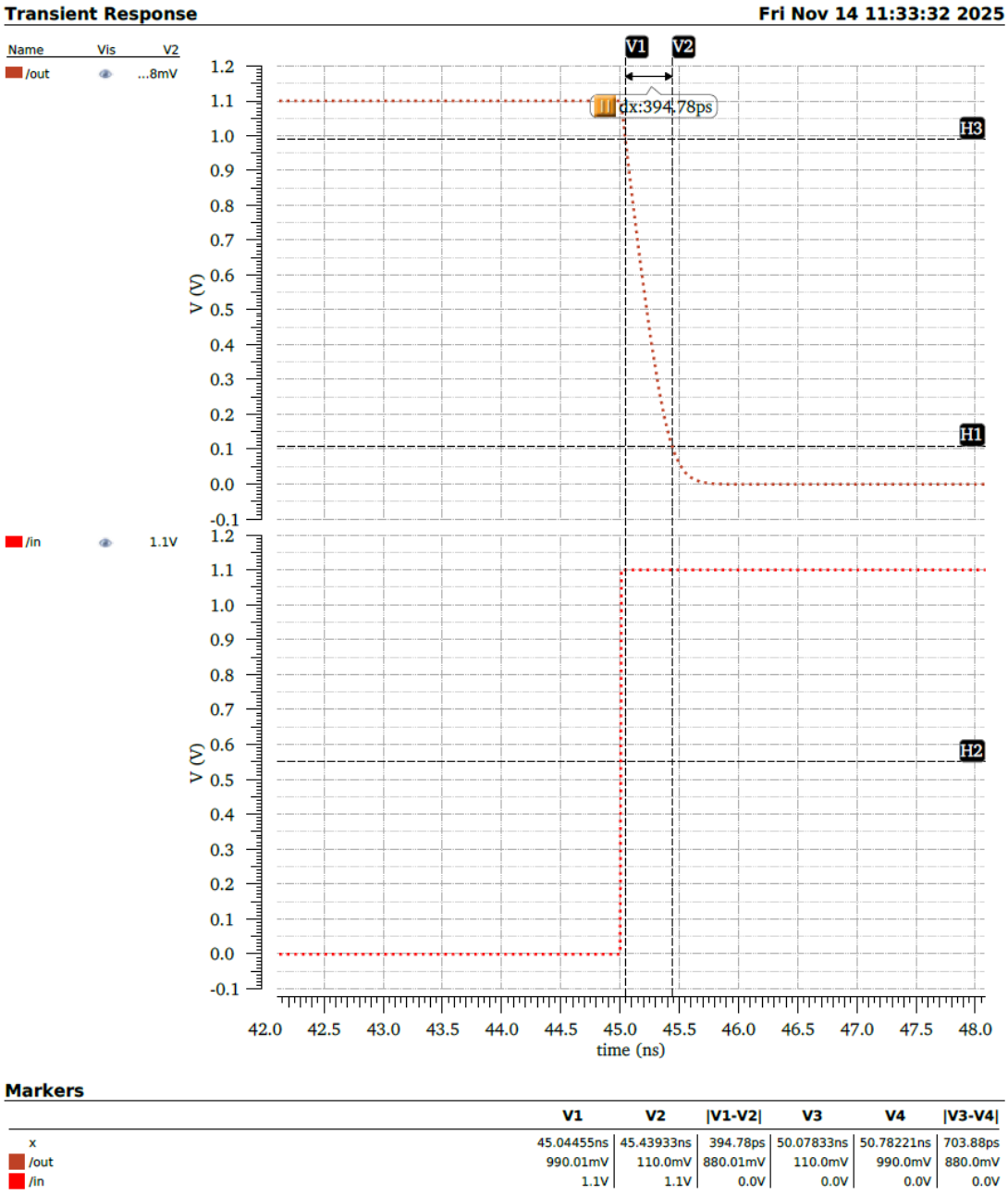


图 16　T_r 计算(手动标注 & 软件计算，公式在 Stack 顶层)



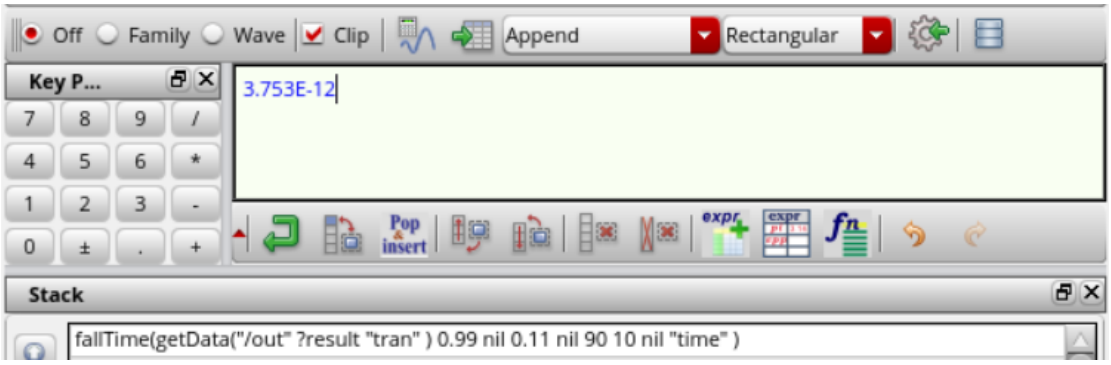


图 17　T_f 计算(手动标注 & 软件计算，公式在 Stack 顶层)

子实验二：W_p 尺寸优化（$t_{phl} = t_{plh}$）

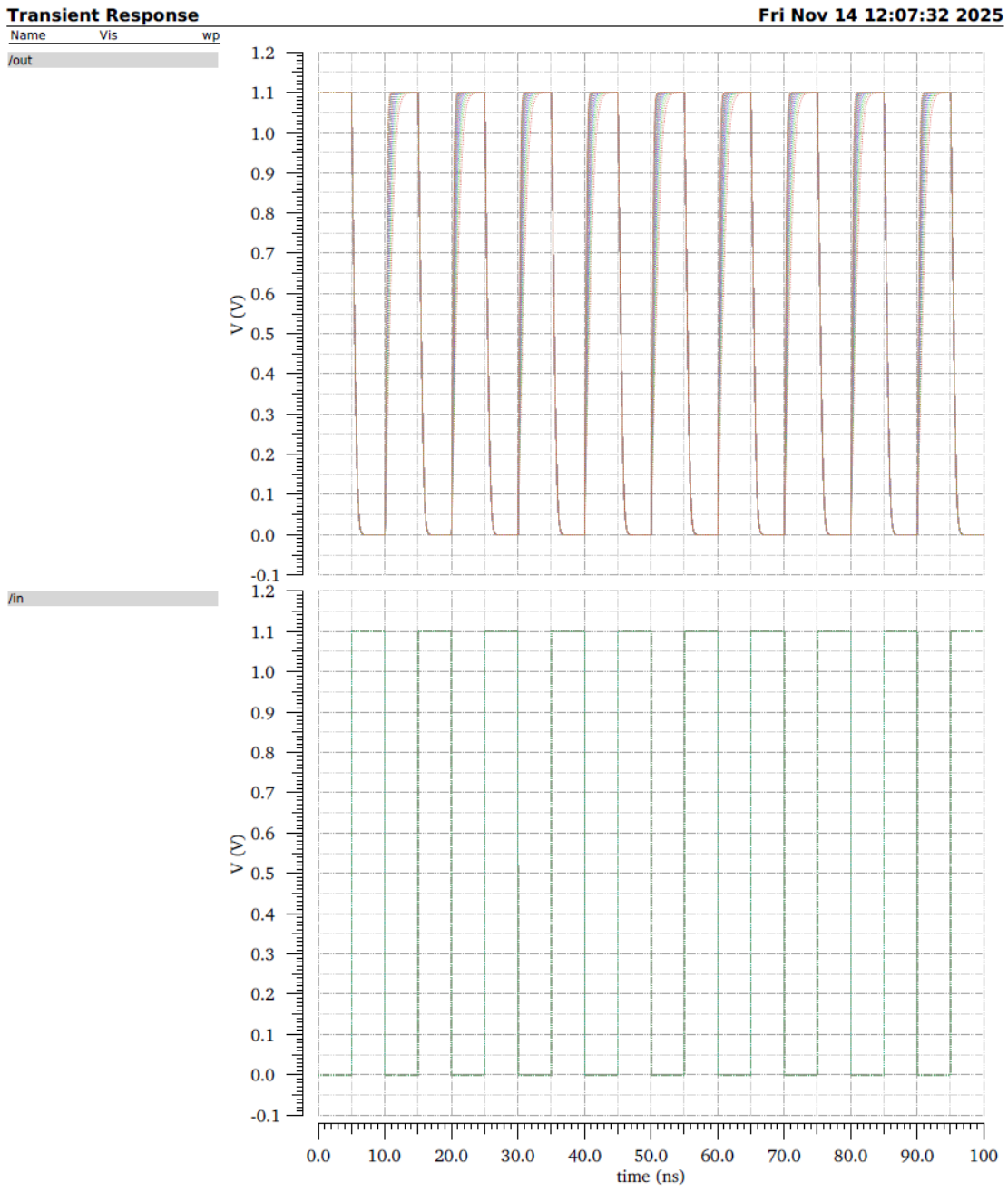

Transient Response
Fri Nov 14 12:07:32 2025
Name
Vis
wp
/out
/in
V (V)
time (ns)
0.0
10.0
20.0
30.0
40.0
50.0
60.0
70.0
80.0
90.0
100

图 18　w_p 参数粗扫

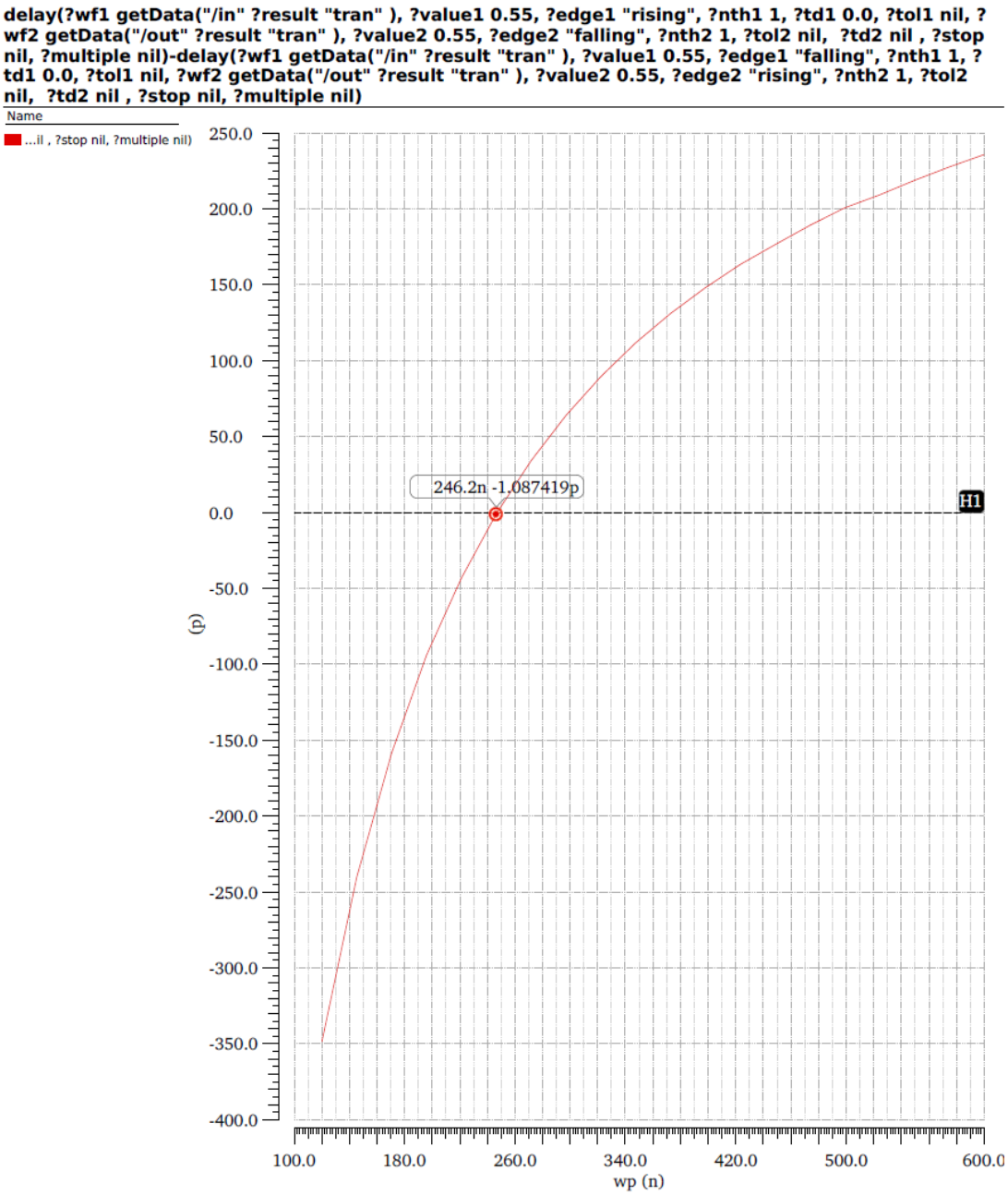


图 19 $(T_{phl}-T_{plh})-w_p$ 图（结果在图中有标注）

仿真结果 $W_p=246.2nm$ 与预期的 $W_p\approx225nm$ 存在约 9.4% 的差异。这种差异体现了**短沟道效应**、**高电场效应**以及**器件模型**与简化理论计算之间的区别。

5.4 功耗计算

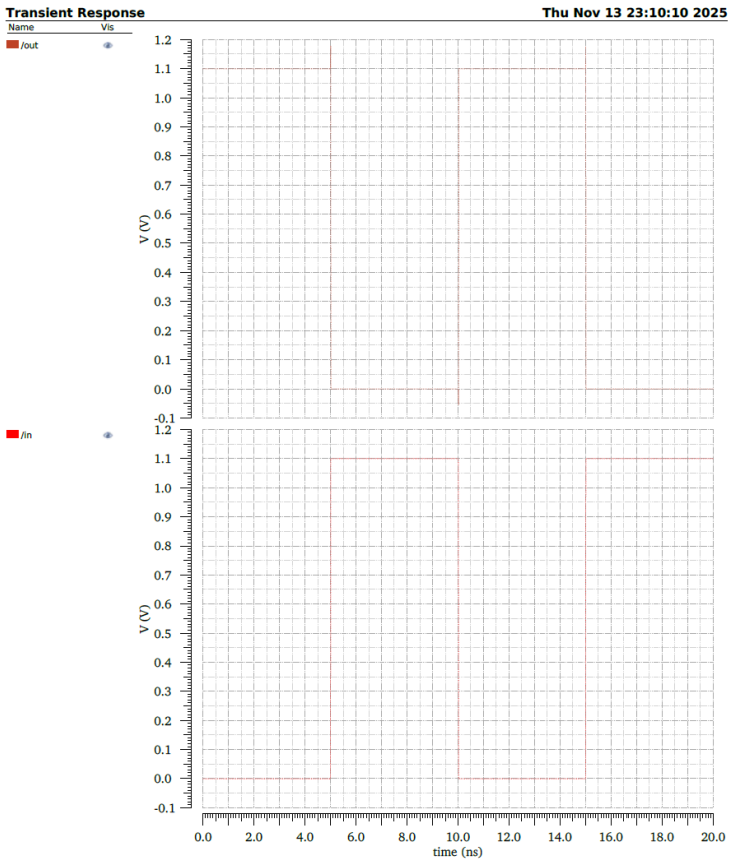


图 20　瞬态仿真信号图

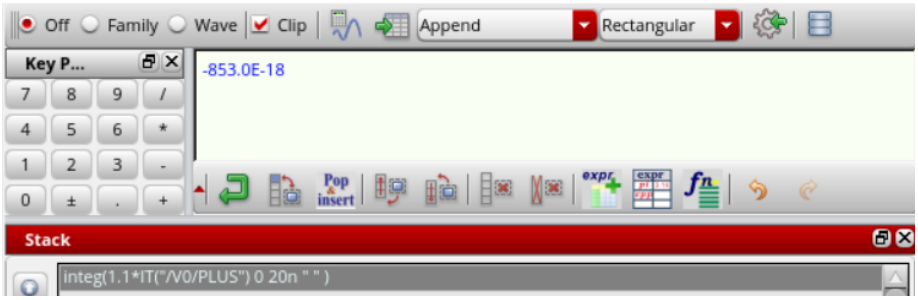


图 21 功率计算结果图

六、 总结

本次实验采用 Cadence Virtuoso ADEL 环境，成功运用了 **cross**，**deriv**，**delay**，和 **integ** 等核心计算器函数，结合**参数扫描和瞬态仿真**，实现了传统 CMOS 实验中难以手动提取的 V_M 值、延迟均衡尺寸（W_p）和总能耗的自动计算与优化。所有关键指标的仿真结果均与题干提供的参考值高度吻合，验证了 CMOS 设计理论和所采用的仿真模型的准确性。