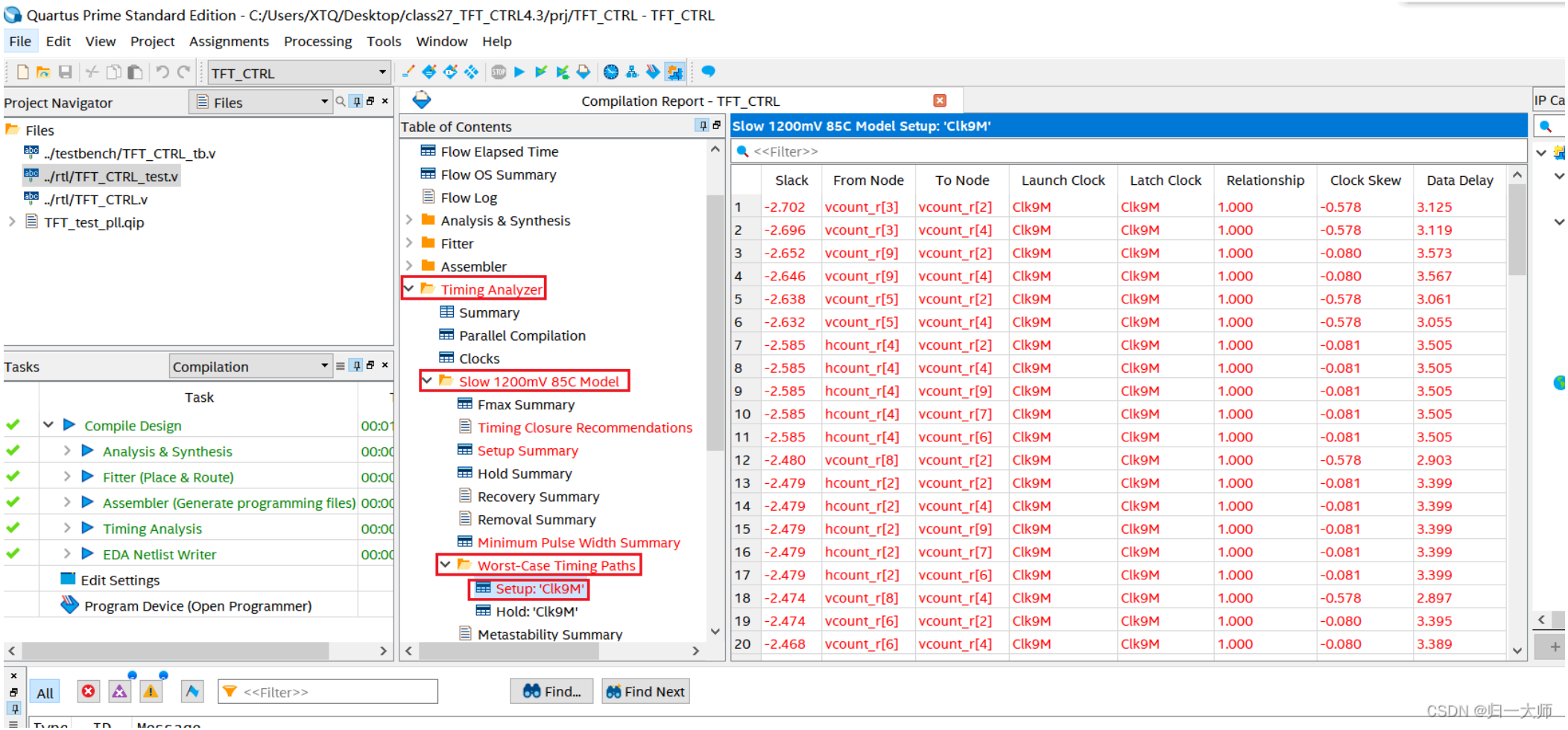
TimeQuest时序路径详解

基于TimeQuest软件来查看时序报告和分析时序路径。

分析最坏传输路径

根据[FPGA　典型时序路径的分析可知，最坏传输路径对应的建立时间（setup time）余量最小。所以，查看最坏传输路径也就是查看建立时间余量最小的路径。如果不对设计加入约束，软件会自动寻找设计中的时钟，给默认的约束，1GHZ；可以直接在 Quartus 的报告界面查看，也可以在 TimeQuest Timing Analyzer 中查看，先介绍如何在 Quartus 报告界面查看。

以小梅哥 TFT　屏为案例，进行全编译后，在 Compilation Report 中，依次展开 TimeQuest Timing Analyzer -> Slow 1200mV 85C Mode–> Worst-Case Timing Paths，点击 Setup “Clk9M”。然后右侧就会自动安装建立时间余量从小到大的顺序排列出来，排在第一位的就是建立时间余量最小的那个信号，如下图所示：

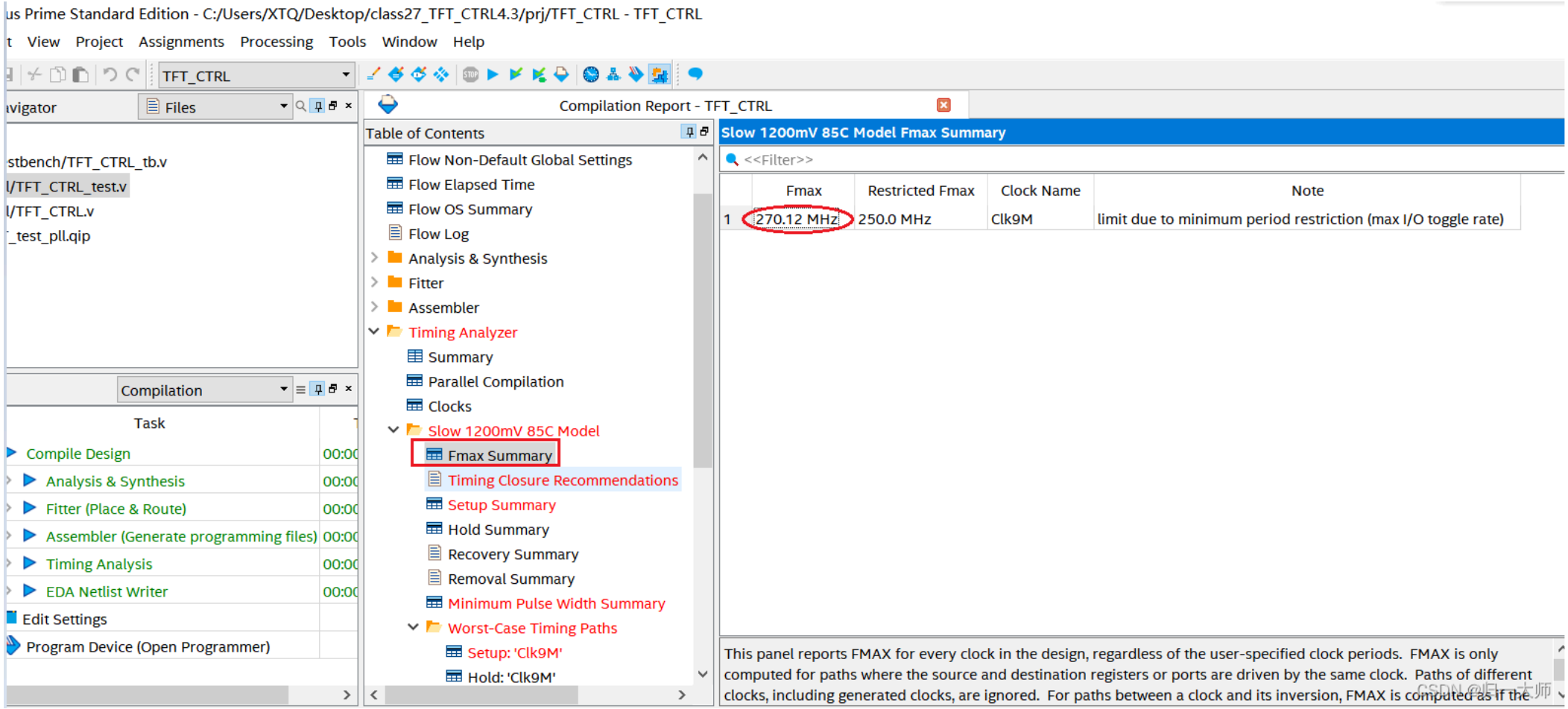


由下图可以查看最坏路径的建立时间余量（Slack），最坏路径的源寄存器（From Node），目的寄存器（To Node），源寄存器发射数据的时钟（Lauch Clock），目的寄存器接收数据的时钟（Lach Clock），Lauch Clock egde与Lach Clock egde的时间差为Relationship（当Lauch Clock和Lach Clock为同一个时钟时，Relationship值为时钟周期，否则不是时钟周期），时钟偏斜Clock Skew，数据传输延迟Data Delay。

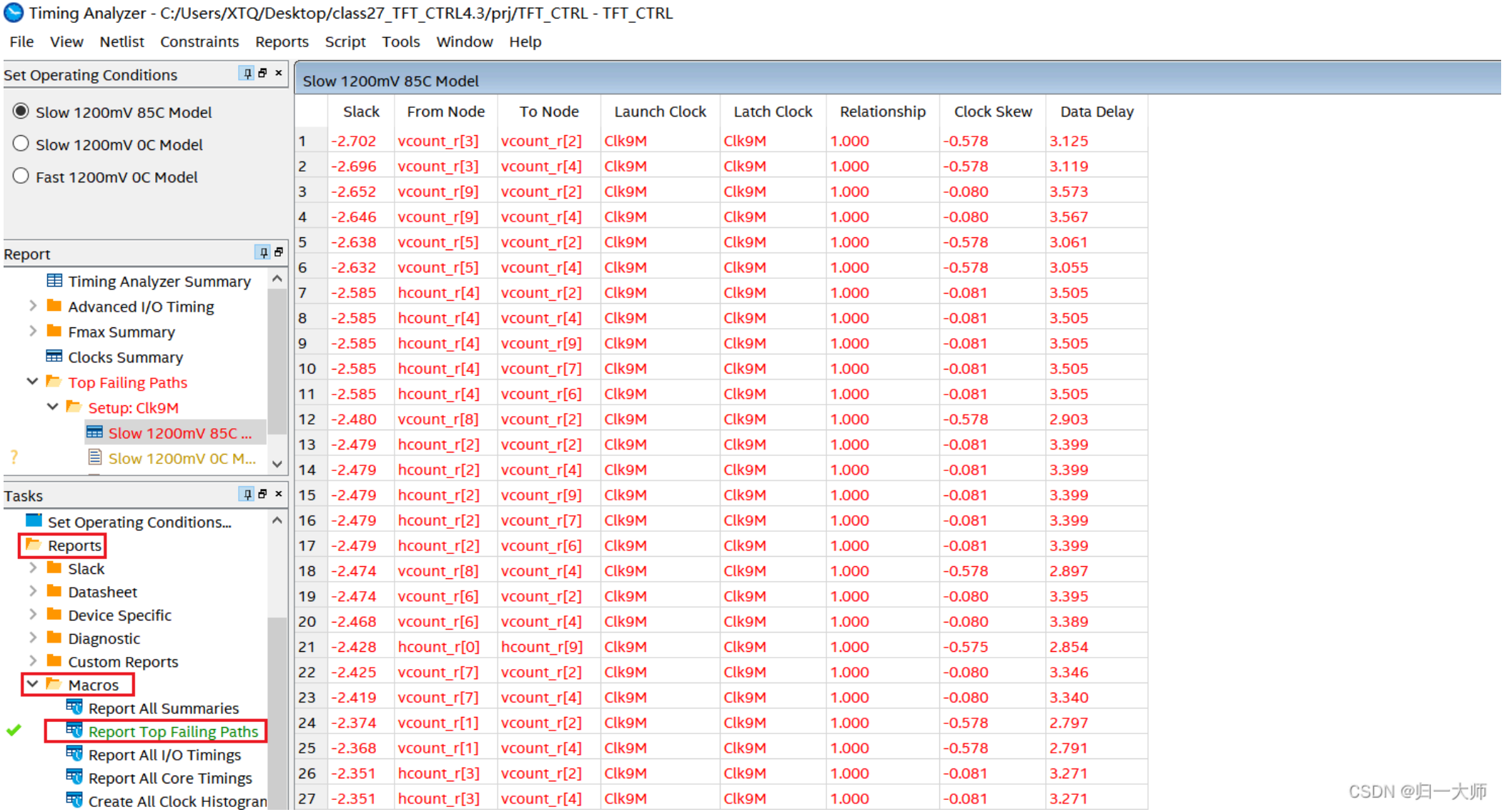
Slow 1200mV 85C Model

	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	-2.702	vcount_r[3]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.578	3.125
2	-2.696	vcount_r[3]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.578	3.119
3	-2.652	vcount_r[9]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.080	3.573
4	-2.646	vcount_r[9]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.080	3.567
5	-2.638	vcount_r[5]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.578	3.061
6	-2.632	vcount_r[5]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.578	3.055
7	-2.585	hcount_r[4]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.081	3.505
8	-2.585	hcount_r[4]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.081	3.505
9	-2.585	hcount_r[4]	vcount_r[9]	Clk9M	Clk9M	1.000	-0.081	3.505
10	-2.585	hcount_r[4]	vcount_r[7]	Clk9M	Clk9M	1.000	-0.081	3.505
11	-2.585	hcount_r[4]	vcount_r[6]	Clk9M	Clk9M	1.000	-0.081	3.505
12	-2.480	vcount_r[8]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.578	2.903
13	-2.479	hcount_r[2]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.081	3.399
14	-2.479	hcount_r[2]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.081	3.399
15	-2.479	hcount_r[2]	vcount_r[9]	Clk9M	Clk9M	1.000	-0.081	3.399
16	-2.479	hcount_r[2]	vcount_r[7]	Clk9M	Clk9M	1.000	-0.081	3.399
17	-2.479	hcount_r[2]	vcount_r[6]	Clk9M	Clk9M	1.000	-0.081	3.399
18	-2.474	vcount_r[8]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.578	2.897
19	-2.474	vcount_r[6]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.080	3.395
20	-2.468	vcount_r[6]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.080	3.389
21	-2.428	hcount_r[0]	hcount_r[9]	Clk9M	Clk9M	1.000	-0.575	2.854
22	-2.425	vcount_r[7]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.080	3.346
23	-2.419	vcount_r[7]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.080	3.340
24	-2.374	vcount_r[1]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.578	2.797
25	-2.368	vcount_r[1]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.578	2.791
26	-2.351	hcount_r[3]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.081	3.271
27	-2.351	hcount_r[3]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.081	3.271
28	-2.351	hcount_r[3]	vcount_r[9]	Clk9M	Clk9M	1.000	-0.081	3.271
29	-2.351	hcount_r[3]	vcount_r[7]	Clk9M	Clk9M	1.000	-0.081	3.271

查看建立时间余量，可以看到，建立时间余量最小的路径出现在 hcount_r[8]和 vcount_r[1]之间，建立时间余量为-2.182，根据 Fmax 的计算方法，计算出Fmax = 1 / (Tclk - Tslack) = 1 / (1– (-2.702)) = 270.1243MHZ，与系统时序报告的最大时钟频率一致；



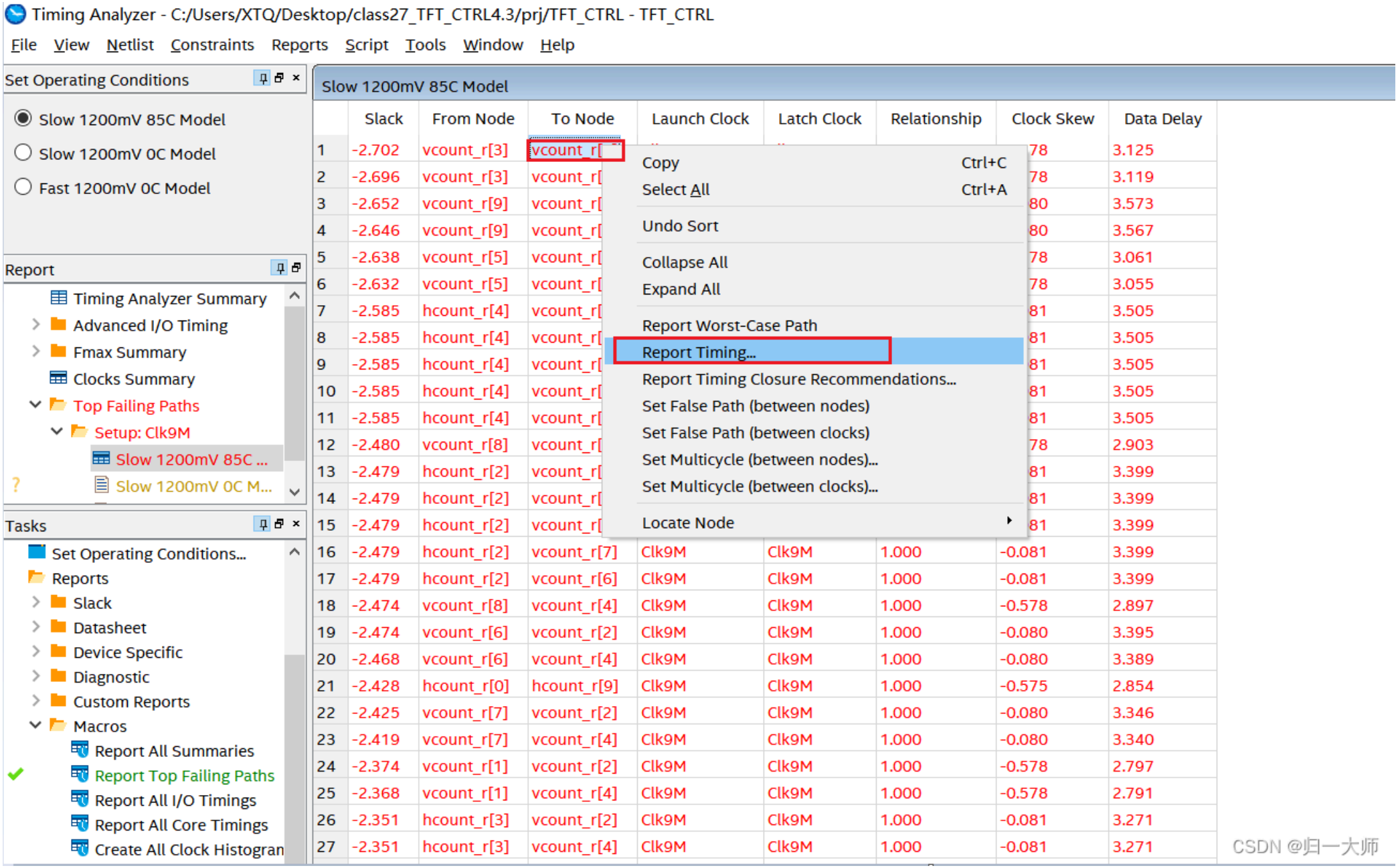
在TimeQuest Timing Analyzer 中查看时，创建网表，READ SDC之后，通过Reports→Macros→ Report Top Failling Paths查看最坏路径，如下图所示：



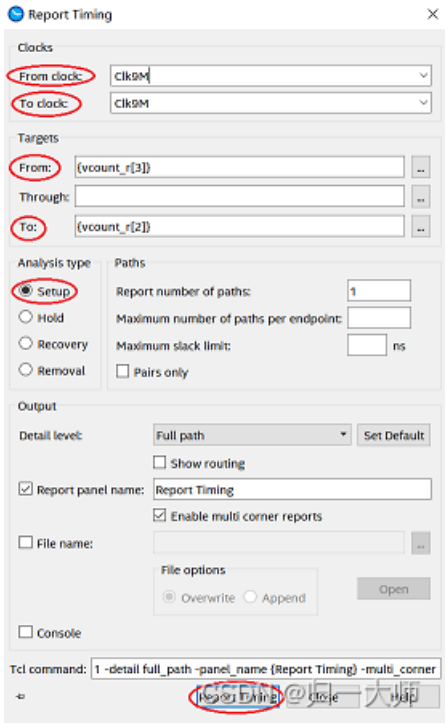
	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	-2.702	vcount_r[3]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.578	3.125
2	-2.696	vcount_r[3]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.578	3.119
3	-2.652	vcount_r[9]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.080	3.573
4	-2.646	vcount_r[9]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.080	3.567
5	-2.638	vcount_r[5]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.578	3.061
6	-2.632	vcount_r[5]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.578	3.055
7	-2.585	hcount_r[4]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.081	3.505
8	-2.585	hcount_r[4]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.081	3.505
9	-2.585	hcount_r[4]	vcount_r[9]	Clk9M	Clk9M	1.000	-0.081	3.505
10	-2.585	hcount_r[4]	vcount_r[7]	Clk9M	Clk9M	1.000	-0.081	3.505
11	-2.585	hcount_r[4]	vcount_r[6]	Clk9M	Clk9M	1.000	-0.081	3.505
12	-2.480	vcount_r[8]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.578	2.903
13	-2.479	hcount_r[2]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.081	3.399
14	-2.479	hcount_r[2]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.081	3.399
15	-2.479	hcount_r[2]	vcount_r[9]	Clk9M	Clk9M	1.000	-0.081	3.399
16	-2.479	hcount_r[2]	vcount_r[7]	Clk9M	Clk9M	1.000	-0.081	3.399
17	-2.479	hcount_r[2]	vcount_r[6]	Clk9M	Clk9M	1.000	-0.081	3.399
18	-2.474	vcount_r[8]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.578	2.897
19	-2.474	vcount_r[6]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.080	3.395
20	-2.468	vcount_r[6]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.080	3.389
21	-2.428	hcount_r[0]	hcount_r[9]	Clk9M	Clk9M	1.000	-0.575	2.854
22	-2.425	vcount_r[7]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.080	3.346
23	-2.419	vcount_r[7]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.080	3.340
24	-2.374	vcount_r[1]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.578	2.797
25	-2.368	vcount_r[1]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.578	2.791
26	-2.351	hcount_r[3]	vcount_r[2]	Clk9M	Clk9M	1.000	-0.081	3.271
27	-2.351	hcount_r[3]	vcount_r[4]	Clk9M	Clk9M	1.000	-0.081	3.271

已经找到了设计中影响整个设计的最高运行时钟频率的路径，那么，这个路径究竟是对应着代码中的哪个部分呢？又该如何来优化这段逻辑呢？

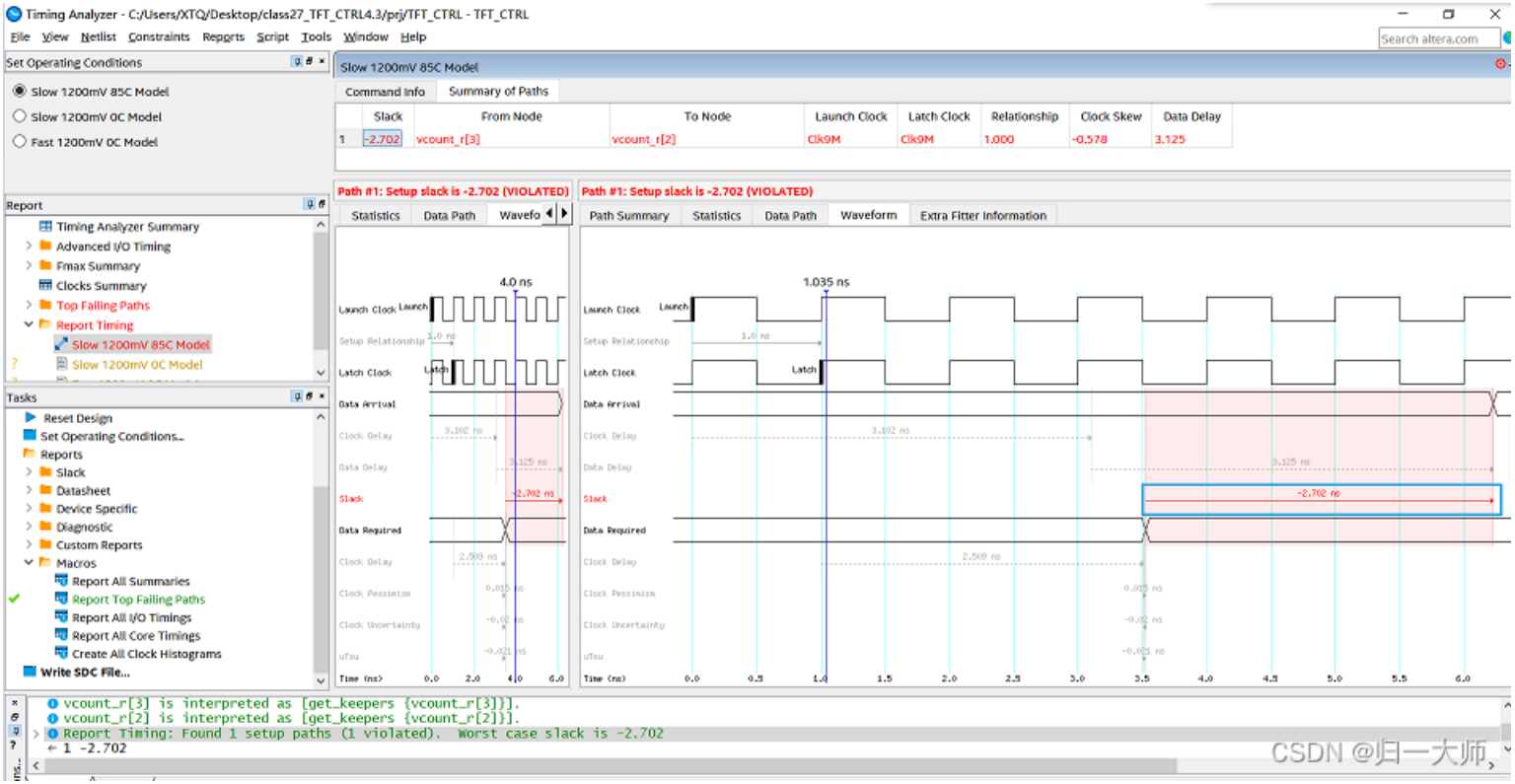
选中最坏路径的任意一个属性，右击鼠标，选择Report Timing…，如下图所示：



弹出以下页面，可知最坏路径的源寄存器时钟是clk9M，目的寄存器的时钟也是clk9M，源寄存器是vcount_r[3]，目的寄存器是 vcount_r[2]，分析的是建立时间Setup，之后点击Report Timing；



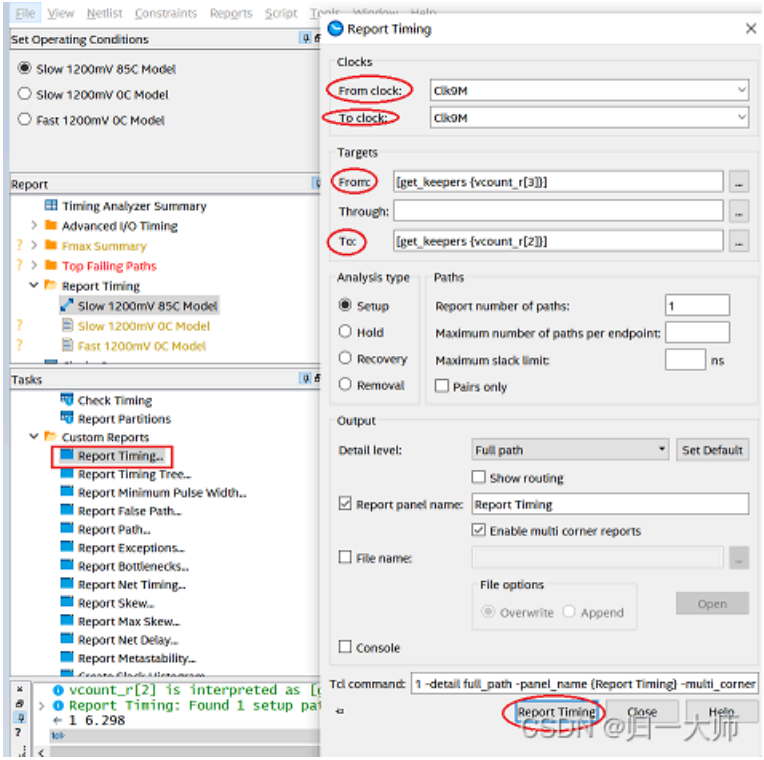
会弹出如下页面，此时得知建立时间是不满足的，由于系统没办法达到1GHZ，所以不能进行优化。



将时钟约束改为100MHZ，之后重新Read SDC，查看报告，发现全部满足时序要求，用同样的方法可以打开之前的最坏路径，得到下图所示的时序分析图：

- 部分版本quartus不能在直接查看之前对应最坏路径的时序分析图，可以手动添加路径，方法如下：

点击Custom Reports→Reprt Timing…，将源寄存器、目的寄存器的时钟以及寄存器名填写后选择Report Timing，也可以打开之前的路径分析时序了。



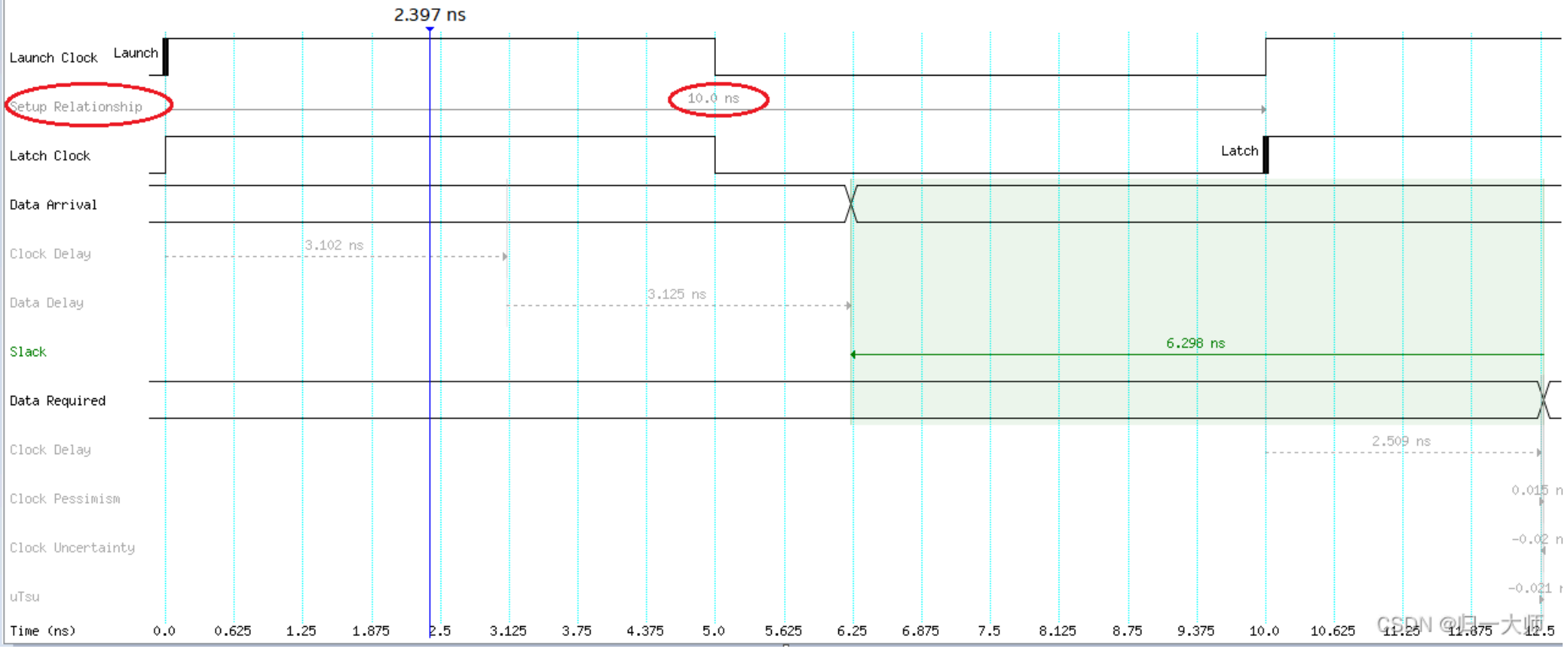
具体分析

Waveform页面进行分析

	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	6.298	vcount_r[3]	vcount_r[2]	Clk9M	Clk9M	10.000	-0.578	3.125

Path #1: Setup slack is 6.298

Path Summary | Statistics | Data Path | Waveform | Extra Fitter Information



因为此路径源寄存器和目的寄存器共用同一个时钟，所以RelationShip与时钟周期相等，由上图知RelationShip与为10ns，即Tclk = 10ns，与100MHZ的时钟周期结果一致；

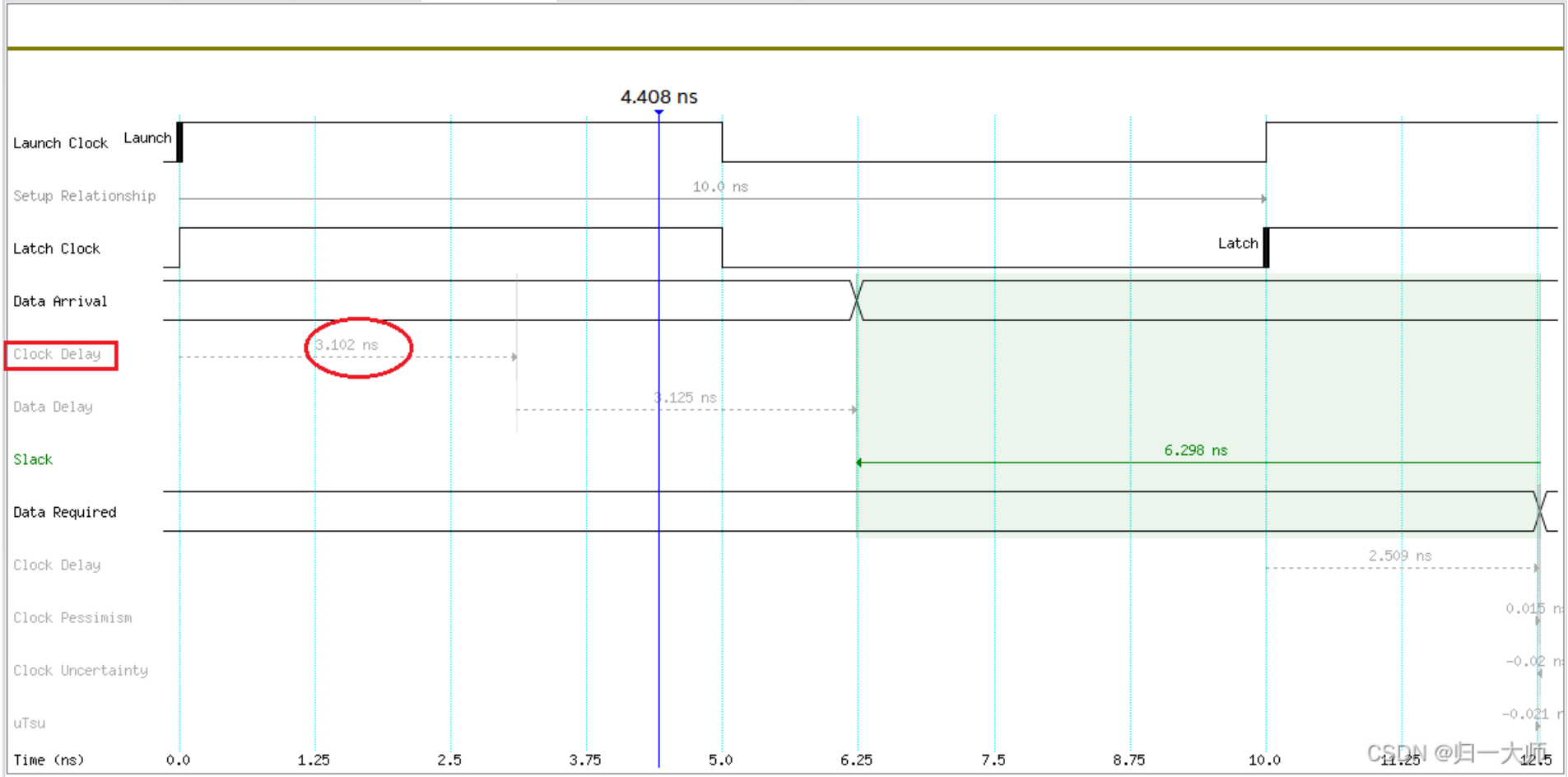
Slow 1200mV 85C Model

Command Info | Summary of Paths

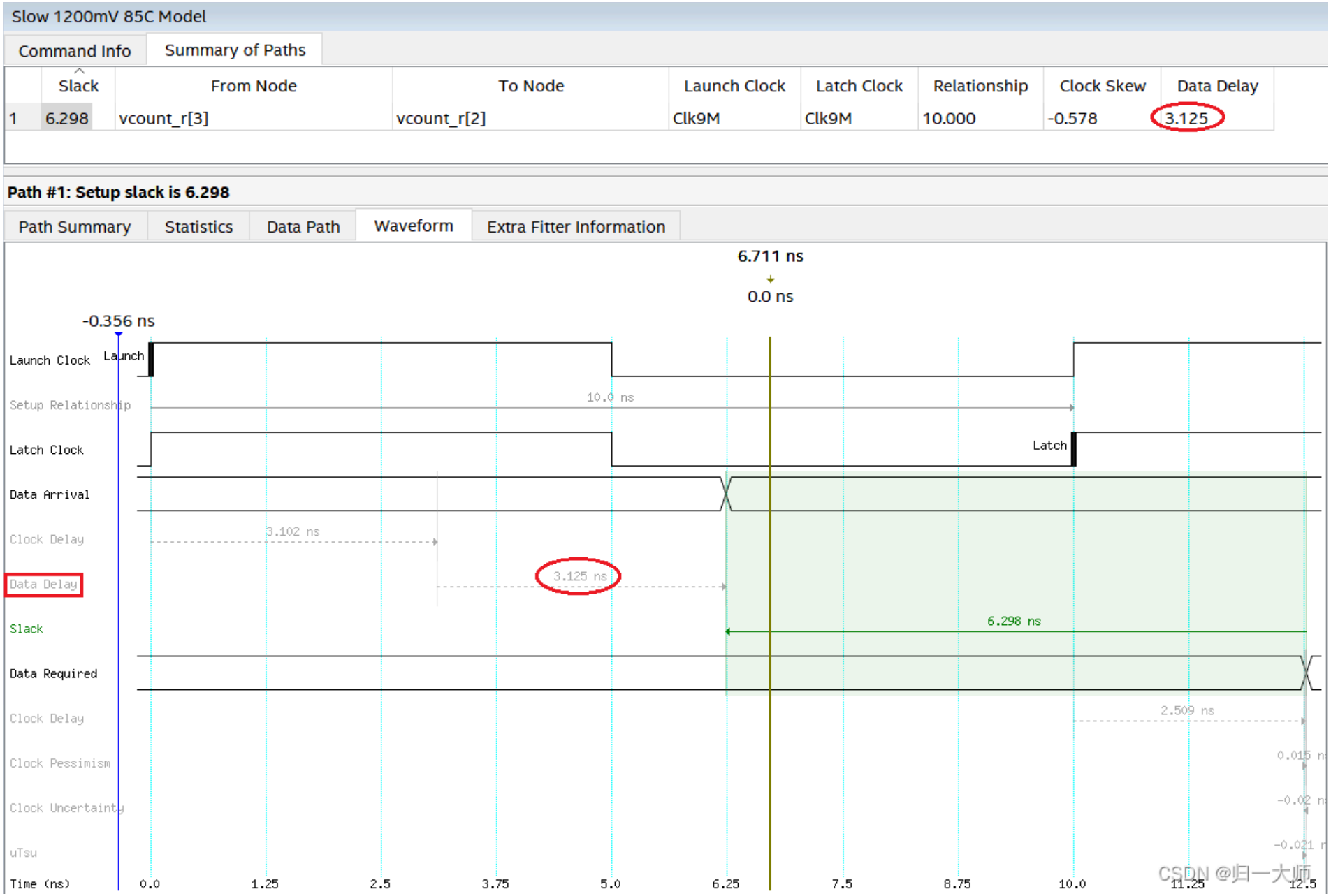
	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	6.298	vcount_r[3]	vcount_r[2]	Clk9M	Clk9M	10.000	-0.578	3.125

Path #1: Setup slack is 6.298

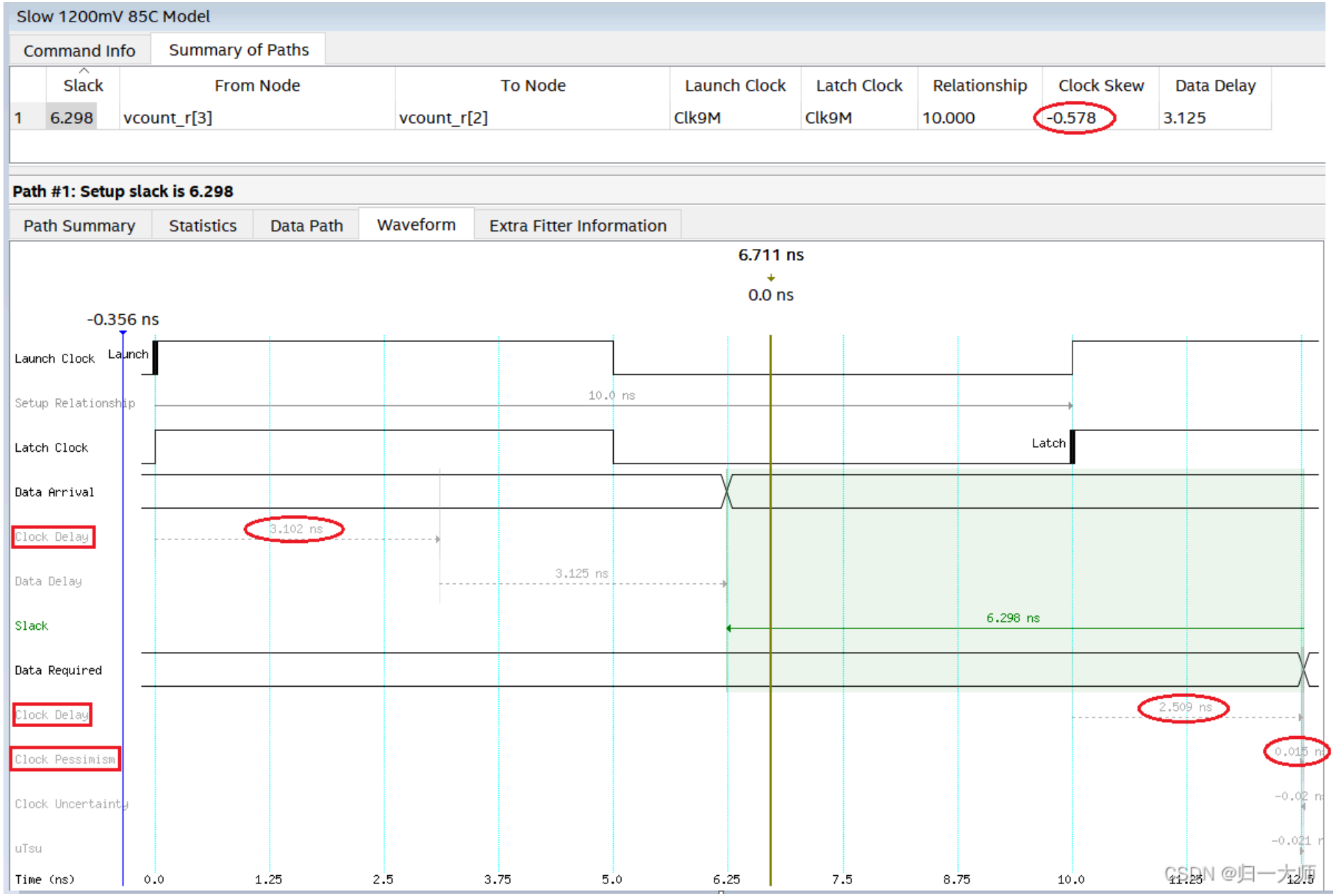
Path Summary | Statistics | Data Path | Waveform | Extra Fitter Information



由上图知，Clock Delay是源寄存器时钟相对于时钟Clock的延迟，即Tclk1 = 3.102ns。



Data Delay为数据传输延迟，Altera把寄存器数据延迟（Tco）和数据传输延迟（Tdata）统称为数据传输延迟Data Delay，所以Data Delay = Tco + Tdata = 3.125ns；



时钟偏斜：Tskew = Clock skew = -0.578ns；

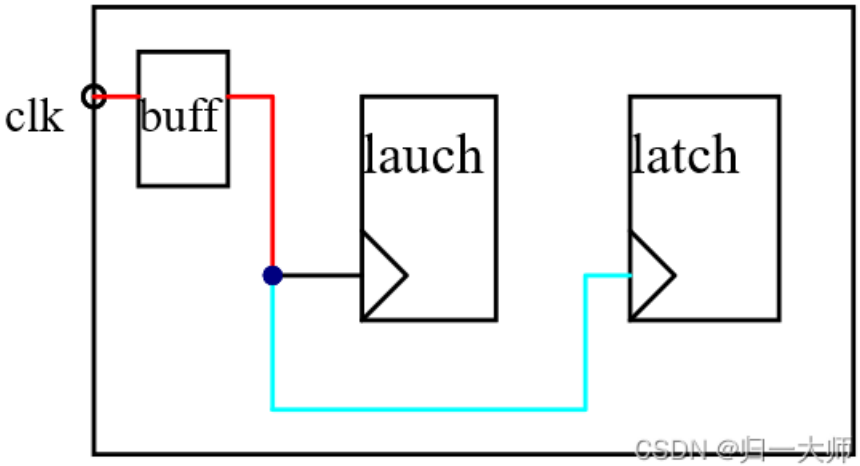
目的寄存器时钟延迟：Tclk2 = Clock Delay = 2.509ns；

时钟悲观：Clock Pessimism = 0.015ns；

Tskew = Tclk2 - Tclk1 + Clock Pessimism = 2.509 - 3.102 + 0.015 = -0.578ns；

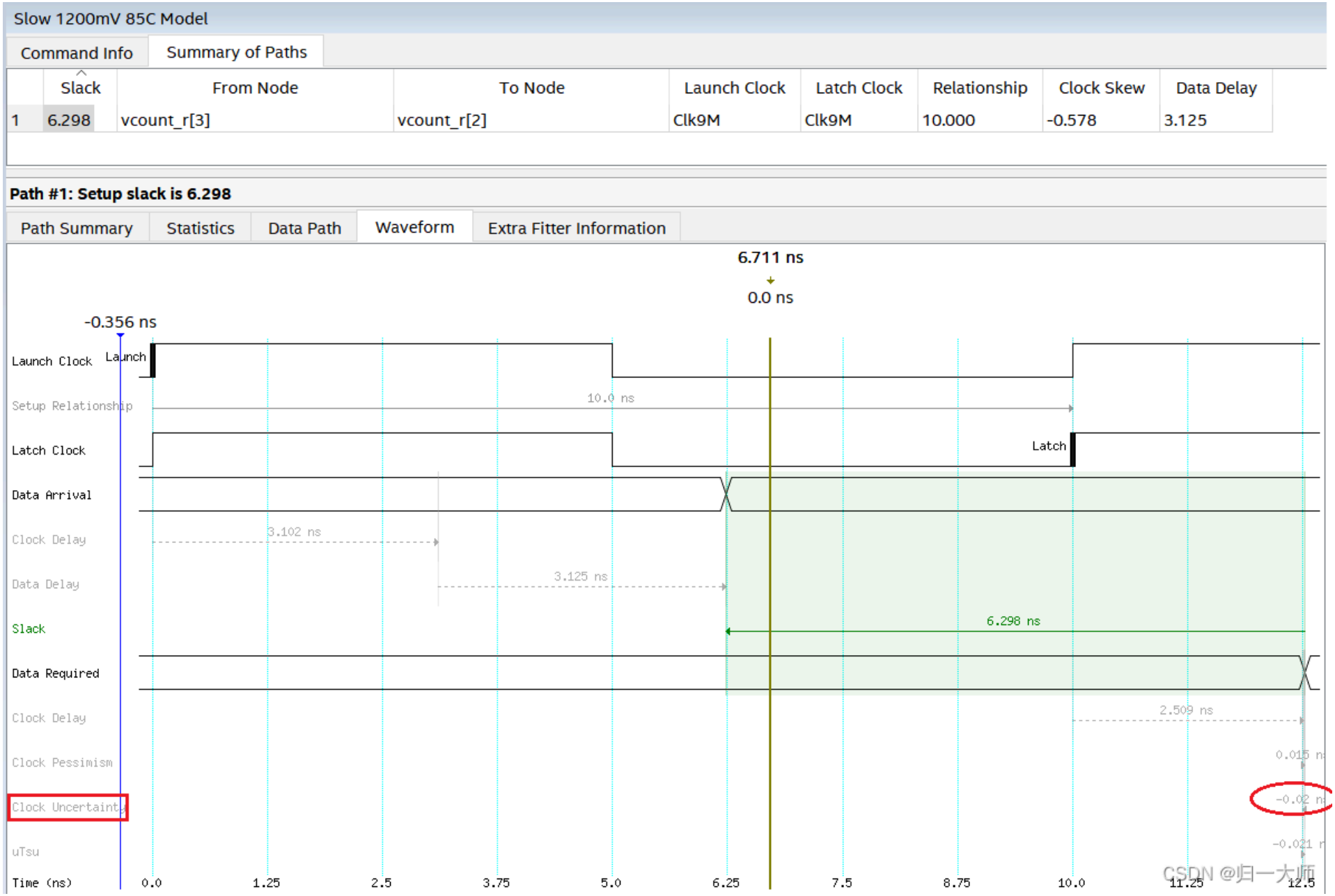
时钟悲观（Clock Pessimism）

此路径是最坏路径，虽然将系统时钟改变，此设计能满足当前时钟，但是并没有把此设计进行重新编译，网表文件并没有改变，所以此路径还是此网表的最坏路径，即建立时间余量Slack最小，而Slack = Tclk + Tskew - Tsu - Tco - Tdata，由于时钟周期固定，所以Tclk是定值，当器件固定时，Tsu和Tco也是定值，当逻辑设计固定时，Tdata也是定值，只有时钟偏斜Tskew可以改变，Tskew = Tclk2 - Tclk1，两个寄存器是时钟如下图所示：



要得到建立时间余量Slack为最小值，Tskew的值就必须为最小，所以分析计算Tclk2的时候，EDA软件会认为时钟从源端口出发按照最快的速度到达目的寄存器时钟端口（如上图时钟以最快速度通过红色和蓝色路径）。分析计算Tclk1的时候，EDA软件会认为时钟从源端口出发按照最慢的速度到达源寄存器的时钟端口（上图中时钟以最慢的速度通过红色和黑色路径）。由于时钟从源端口到达源寄存器时钟端口和目的寄存器时钟端口过程中，会经过一段相同的路径，同一信号经过同一路径，不可能会是两种速度，所以通过Tclk2 - Tclk1计算出的时钟偏斜是偏小的；

时钟通过同一段路径按照不同的速率计算出的时间在真实情况下是不存在的，EDA软件在分析时钟的时候过分悲观，导致时钟偏斜Skew值偏小，所以需要将一段路径过分悲观的值补偿回来，加上一个时钟悲观值（Clock Pessimism），此参数会被算入数据需求时间（data require time ）；



由上图知时钟不确定参数TClock Uncertainty = -0.02ns；

时钟不确定参数（TClock Uncertainty）

一个周期为10ns，占空比为50%的系统时钟，高电平和低电平持续的时间应该都是5ns，但是实际的时钟都会存在误差，几乎不可能刚好5.000000ns，这就会使得上升沿提前或者滞后一点到达，从而影响建立时间比标准的建立时间提前或者滞后，所以就引入时钟不确定参数Clock Uncertainty对其进行补偿，该参数会被算入数据需求时间（data require time）；

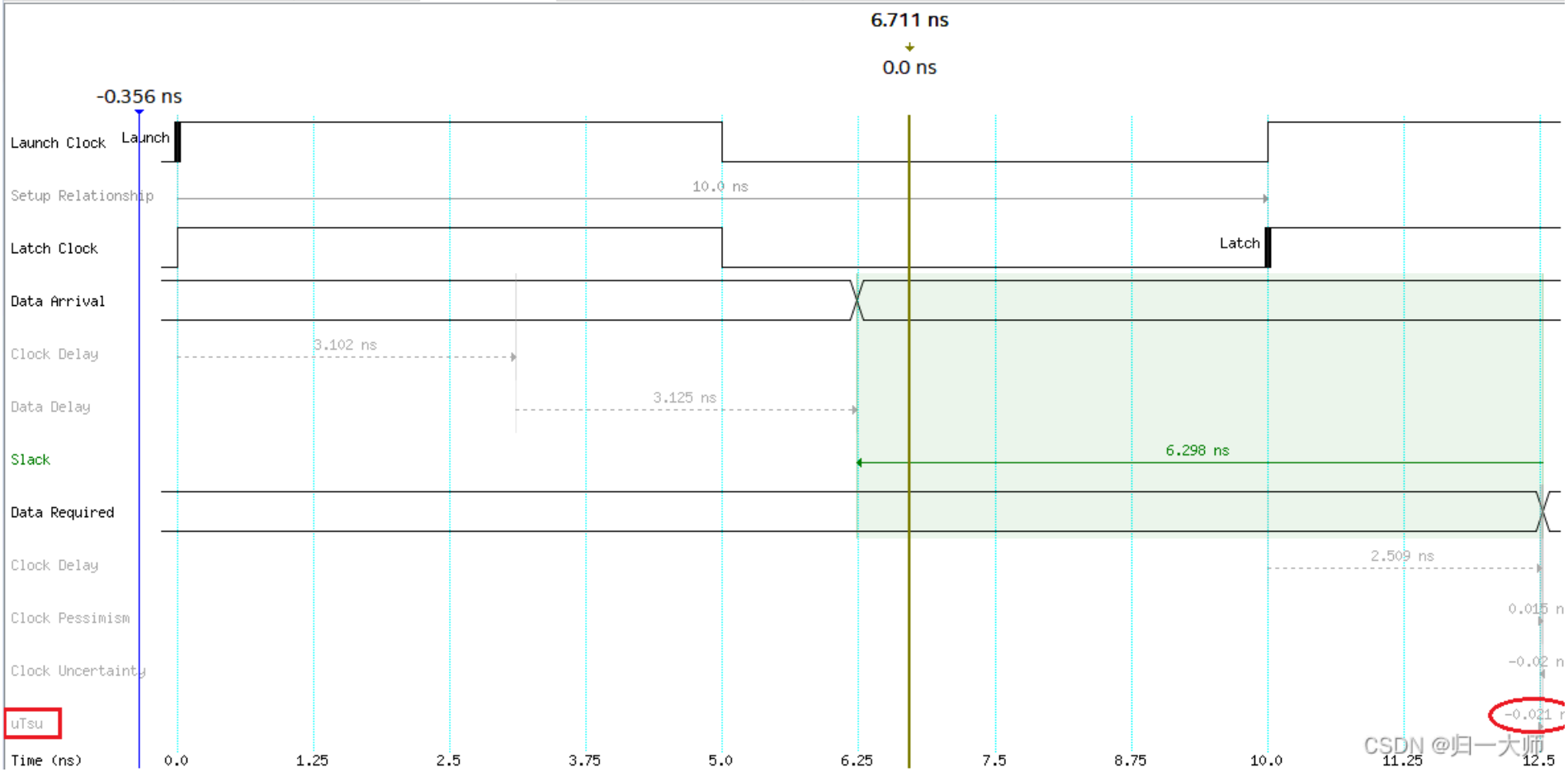
Slow 1200mV 85C Model

Command Info | Summary of Paths

	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	6.298	vcount_r[3]	vcount_r[2]	Clk9M	Clk9M	10.000	-0.578	3.125

Path #1: Setup slack is 6.298

Path Summary | Statistics | Data Path | Waveform | Extra Fitter Information



建立时间：Tsu = uTsu = -0.021ns；Altera官方给出，该器件的建立时间可以是负值，在正常范围之内；

由上述参数可以计算：

数据到达时间（data arrival time）：Tclk1 + Tco + Tdata；

数据需求时间（data require time）：Tclk + Tclk2 - Tsu = Tclk + Tclk2 -Tsu + TClock Pessimism + TClock Uncertainty（时钟不确定参数）；

建立时间余量：Slack = Tclk + Tskew - Tco - Tdata - Tsu

= Tclk + Tclk2 +TClock Pessimism + TClock Uncertainty - Tsu - Tclk1 - Tco - Tdata

= 10 + 2.509 + 0.015+ (-0.02)- (-0.021) - 3.102- 3.125

= 6.298ns

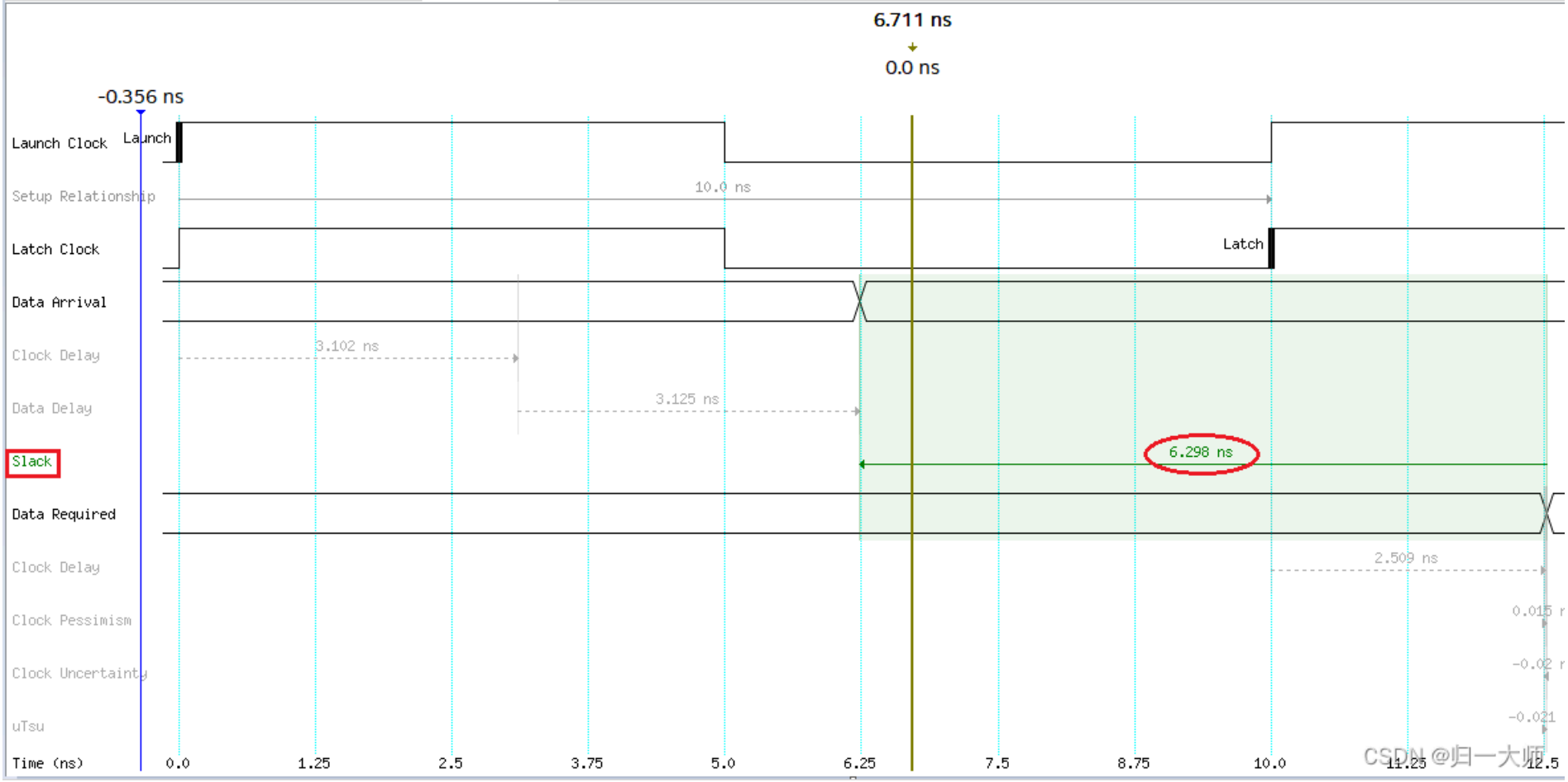
Slow 1200mV 85C Model

Command Info | Summary of Paths

	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	6.298	vcount_r[3]	vcount_r[2]	Clk9M	Clk9M	10.000	-0.578	3.125

Path #1: Setup slack is 6.298

Path Summary | Statistics | Data Path | Waveform | Extra Fitter Information



通过上图知，建立时间余量计算正确；

通过数据路径（Data Path）进行分析

前面使用waveform波形对路径进行分析，后面使用数据路径（Data Path）进行分析，点击Data Path打开下图页面：

Slow 1200mV 85C Model

Command Info | Summary of Paths

	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	6.298	vcount_r[3]	vcount_r[2]	Clk9M	Clk9M	10.000	-0.578	3.125

Path #1: Setup slack is 6.298

Path Summary | Statistics | Data Path | Waveform | Extra Fitter Information

Data Arrival Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	3.102	3.102					clock path
3	6.227	3.125					data path

Data Required Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	10.000	10.000					latch edge time
2	12.524	2.524					clock path
3	12.504	-0.020					clock uncertainty
4	12.525	0.021		uTsu	1	FF_X28_Y8_N27	vcount_r[2]

由下图知EDA软件是对数据到达时间（data arrival path）和数据需求时间（data required path）分别进行分析的，数据到达时间（data arrival path）是对源寄存器发送时钟路径（Clock path）和数据传输路径（data path）进行分析，数据需求路径（data required path）主要对目的寄存器接收时钟路径（Clock Path）进行分析；

Slow 1200mV 85C Model

Command Info | Summary of Paths

	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	6.298	vcount_r[3]	vcount_r[2]	Clk9M	Clk9M	10.000	-0.578	3.125

Path #1: Setup slack is 6.298

Path Summary | Statistics | Data Path | Waveform | Extra Fitter Information

Data Arrival Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	3.102	3.102					clock path
3	6.227	3.125					data path

Data Required Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	10.000	10.000					latch edge time
2	12.524	2.524					clock path
3	12.504	-0.020					clock uncertainty
4	12.525	0.021		uTsu	1	FF_X28_Y8_N27	vcount_r[2]

分析数据传输路径（data arrival path）（源寄存器时钟传输路径Clock Path）

如下图所示，Total表示路径积累延迟时间，Incr表示经过所在路径的延时，Type表示传输过程中经过发路径类型，其中IC表示经过的是片内互连线，CELL表示经过发是触发器或者查找表（LUT），Location表示到达的器件管脚名或者路径节点，Element表示寄存器名或者元素名；

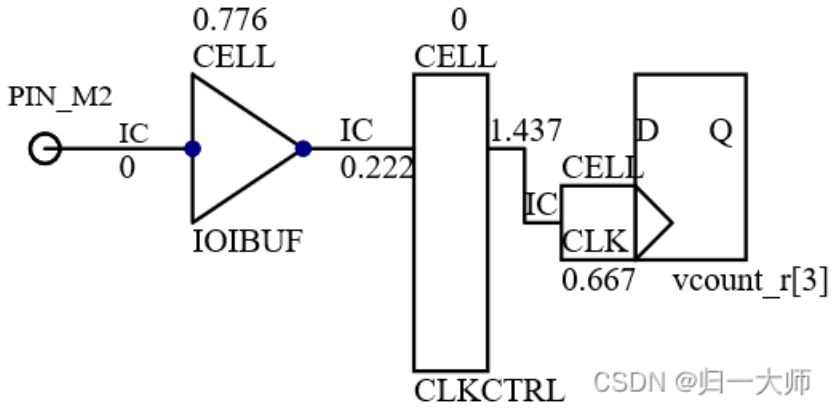
Path #1: Setup slack is 6.298

Path Summary | Statistics | Data Path | Waveform | Extra Fitter Information

Data Arrival Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	3.102	3.102					clock path
1	0.000	0.000					source latency
2	0.000	0.000			1	PIN_M2	Clk9M
3	0.000	0.000	RR	IC	1	IOIBUF_X0_Y11_N15	Clk9M~input|i
4	0.776	0.776	RR	CELL	2	IOIBUF_X0_Y11_N15	Clk9M~input|o
5	0.998	0.222	RR	IC	1	CLKCTRL_G4	Clk9M~inputclkctrl|inclk[0]
6	0.998	0.000	RR	CELL	20	CLKCTRL_G4	Clk9M~inputclkctrl|outclk
7	2.435	1.437	RR	IC	1	FF_X28_Y8_N1	vcount_r[3]|clk
8	3.102	0.667	RR	CELL	1	FF_X28_Y8_N1	vcount_r[3]
3	6.227	3.125					data path

上述路径可以简化为下图传输路径：



从源寄存器发射时钟沿开始计算延迟时间（launch edge time），时钟从时钟引脚M2进入FPGA，经过片内互连线（IC）进入IO_BUF，需要0.776ns才能输出到时钟树上，经过0.222ns的片内互连线延迟到达CLKCTRL，没有延迟，直接输出，再次经过1.437ns的片内互联线延迟到达源寄存器（Vcount_r [3]）的时钟端，在经过0.667ns才能被源触发器使用；

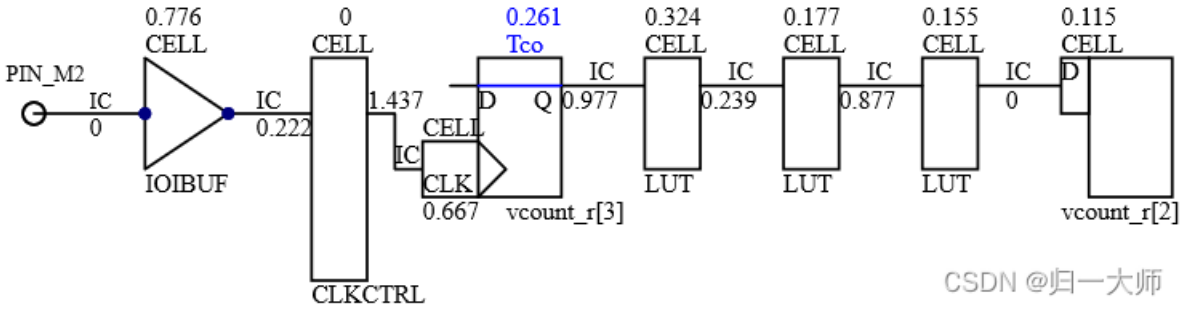
分析数据传输路径（data arrival path）（源寄存器数据传输路径Data Path）

如下图框中所示：

Data Arrival Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	> 3.102	3.102					clock path
3	v 6.227	3.125					data path
1	3.363	0.261		uTco	1	FF_X28_Y8_N1	vcount_r[3]
2	3.363	0.000	RR	CELL	5	FF_X28_Y8_N1	vcount_r[3]|q
3	4.340	0.977	RR	IC	1	LCCOMB_X28_Y7_N28	Equal1~2|datac
4	4.664	0.324	RR	CELL	1	LCCOMB_X28_Y7_N28	Equal1~2|combout
5	4.903	0.239	RR	IC	1	LCCOMB_X28_Y7_N10	Equal1~3|datad
6	5.080	0.177	RR	CELL	5	LCCOMB_X28_Y7_N10	Equal1~3|combout
7	5.957	0.877	RR	IC	1	LCCOMB_X28_Y8_N26	vcount_r~1|datad
8	6.112	0.155	RF	CELL	1	LCCOMB_X28_Y8_N26	vcount_r~1|combout
9	6.112	0.000	FF	IC	1	FF_X28_Y8_N27	vcount_r[2]|d
10	6.227	0.115	FF	CELL	1	FF_X28_Y8_N27	vcount_r[2]

延迟时间从源寄存器时钟延迟时间结束开始计算，所以此处从3.102ns开始计算，源寄存器有0.261ns的延迟时间（Tco），经过0.977ns的片内互连线延迟到达查找表Equal端口datac，经过0.324ns后输出，经过0.239ns的片内互连线延迟到达查找表Equal端口datad，经过0.177ns之后输出，在经过0.877ns的片内互连线延迟到达查找表端口datad，经过0.155ns延迟后输出，之后到达目的寄存器vcount_r[2]的数据端口，经过0.115ns延迟后才能在vcount_r[2]输入端口起作用。



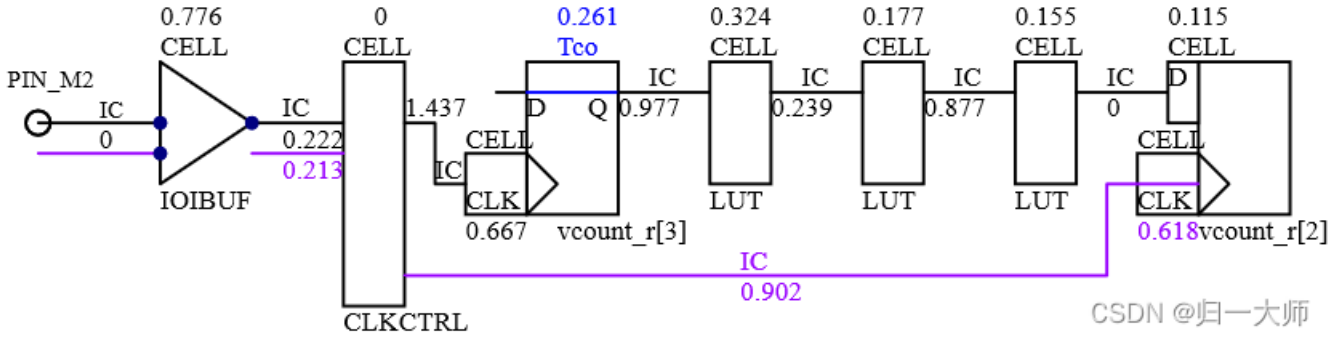
分析数据需求路径（data required path）（目的寄存器时钟传输路径Clock Path）

![在这里插入图片描述](https://img-blog.csdnimg.cn/218a039edb5145f599461841e69df02e.png#pic_center

数据需求时间是针对目的寄存器的，所以起始时间应该以接收时钟沿（latch edge time）为基础开始计算；

上图可知TimeQuest将时钟传输延迟Tclk2，悲观时间，时钟不确定，建立时间均归纳在数据需求路径之中；

目的寄存器时钟路径可以简化为下图中紫色连线部分



从引脚M2到达IO_BUF没有延迟，经过0.776ns后输出到全局时钟树上，之后经过0.213ns内部互连线延迟后到达CLKCTRL，直接输出经0.902ns片内互连线延迟到达目的寄存器（vcount_r[2]）的时钟端，需要经过0.618ns后才能被触发器使用；

由上图知时钟从源引脚到达源寄存器时钟端口和到达目的寄存器时钟端口需要经过一段相同的路径，并且通过这段路径的时间延迟是不同的，所以时钟悲观参数是存在的；

总结

通过Wavefrom可以分析每个路段的延迟时间，查看源寄存器时钟延迟时间Tclk1、寄存器延迟时间Tco、数据传输延迟Tdata、目的寄存器时钟延迟时间Tclk2、建立时间Tsu、时钟偏斜Tskew、建立时间余量Slack、时钟悲观、时钟不确定等参数；通过Data Path查看具体路径每个环节的延迟，片内互连线（IC）延迟，查找表、触发器、buf等CELL延迟；

您的支持是我更新的最大动力！将持续更新工程，如果本文对您有帮助，还请多多点赞👍、评论💬和收藏⭐！