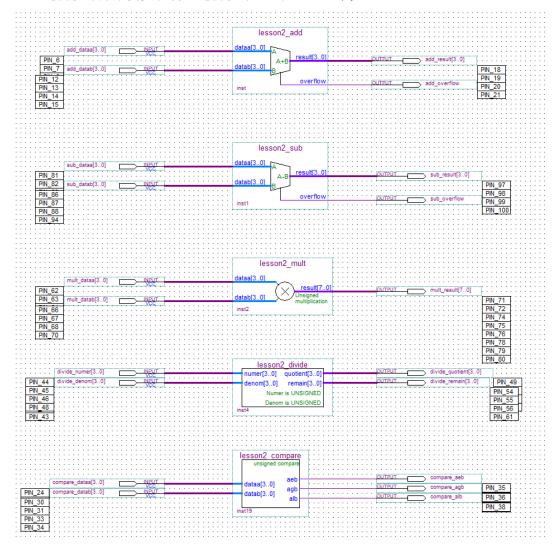
# PLD 实验

## 第二次实验报告

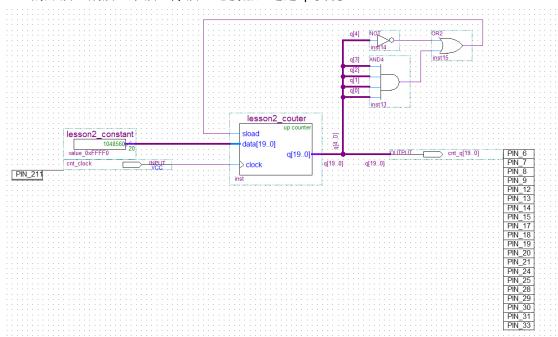
姓名:	<b>蓄</b> 校狂	24. 口.	CC2400204C	今出品词	2024 40 40	
<i>u</i> + 2	THE N'Y 7T.	<b>字</b> 写•	SC21002046	完成时间:	2021.10.18	

#### 一、实验要求

- 1.利用 QuartusII 的"MegaWizard Plug-In Manager",设计输入数据宽度是 4bit 的 ADD、SUB、MULT、DIVIDE、COMPAR,把它们作为一个 project,DEVICE 选用 EPF10K70RC240-4,对它们进行时序仿真。
- 2.利用 QuartusII 的"MegaWizard Plug-In Manager"中的 LPM\_COUNTER,设计一个 20bit 的 up\_only COUNTER,要求该 COUNTER 在 FFFF0H 和 FFFFFH 之间自动循环计数。对该计数器进行时序仿真;分析该 COUNTER 在 EPM7128SLC84-7、EPM7128SLC84-10 和 EPF10K70RC240-2、EPF10K70RC240-4 几种芯片中的最大工作频率;
- 二、逻辑设计方案简要说明、逻辑设计原理图或 HDL 代码



4bit 的加法、减法、乘法、除法、比较器,通过 ip 实现。



循环计数器采用同步 load 实现,同步 load 的数据端接入 lpm\_constant 的常量值 0xFFFF0,由于是同步 load,因此在输出为 0xFFFFF 时 sload 信号即需要有效,这样在下一个时钟沿到来的时刻可以成功装载 0xFFFF0,以达到循环计数的效果,我们可以使用 q[3..0]想与来表示是否计数到 0xFFFFF。

由于启动时刻初始值为 0x00000, 为了能够进入循环状态,引入了 q[4]信号控制,该信号在启动时为 0,而在循环中持续为 1,将其反向输出后接入 sload 控制信号,就可以完成由启动态进入循环状态的功能。

将上述两个控制信号相或再接入 sload 即可完成循环计数器的设计。

三、仿真图【仿真截图及仿真时序波形文件】



上图中所有的输入输出信号显示格式都为 Unsigned Decimal。

在 EPM7128SLC84-7 芯片下,循环计数器的输出波形,周期 8ns。



在 EPM7128SLC84-10 芯片下,循环计数器的输出波形,周期 10ns。

Simulation Waveforms Similation neds: Tining										
Master T	ime Bar:	690.0 ns	Pointer:	29.77 ns	1	nterval	-660.23 ns	Start:		
A <del>M</del> (B)	Напе	0 ps 20.0 ns	40.0 ns	60.0 ns	80.0 ns	100.0 ns	120.0 ns	140.0 ns	160.0 ns	180.0 ns
⊕ ⊕ ⊕ 1	ent_clock m ent_q	FEFFO FFFF1 FFFF2	X FFFF3 X FFFF4 X	FFFFS X FFFF6 X	FFFF7 FFFF8	FFFF9 FFFFA	( PFFFB ( PFFFC )	FFFFD X FFFFE X	TFFFF FFFFO X	FFFF1 FFFF2

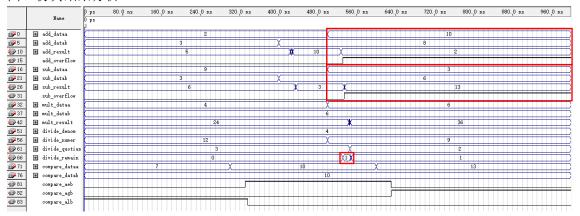
在 EPF10K70RC240-2 芯片下,循环计数器的输出波形,周期 13ns。

	Waveforms node: Tining								
D Maste	Time Bar:	690.0 ns	4 Pointer:	31.59 ns	Interval:	-658,41 ns	Start		End
A 96	Нап-е	0 ps 20.0 ns	40.0 ns 60.0 ns	80. Q ms	100.0 ns 120.0 ns	140.0 ns 160.0 ns	180.0 ns 2	220,0 ns 220,0 ns	240.0 ns
(a) (a) (a) (b) (b) (c) (c) (c) (c) (c) (c) (c) (c) (c) (c	ont_clock  mont_q	00000 )((\$)() 77770	) H111   H112   H113	X FFFF4 FFFF5	77776 77777 X 77776	Y FIFTS X FIFTA Y FIFT	FB X FFFFC X FFFFF	mni mm	FFFF0 FFFF1

在 EPF10K70RC240-4 芯片下,循环计数器的输出波形,周期 18ns。



四、仿真结果分析



加法和减法功能由图中可以看出正常运行,需要注意的是溢出现象。

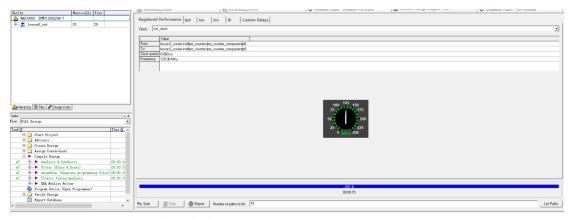
加法的两个 4bit 输入,如果相加得到的值大于 15,即 4bit 输出无法表示,就会产生溢出现象,如图中示例,10(0b1010) + 8(0b1000) = 18(0b10010),由于输出只有 4bit,所以输出为 0b0010 即无符号十进制的 2,溢出指示会被拉高。

减法的溢出可能出现在两个输入数据同号的情况下,如果两个无符号数相减得到一个小于 0的值,即有符号数,就会发生溢出。如图示例: 3(0b0011)-6(0b0110)=0b10011-0b00110 (减法借位) = 0b1101,即无符号十进制的 13;同时溢出指示拉高。

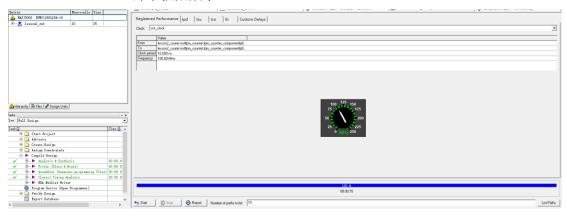
除法器会输出商和余数,图中由于输入数据变化竞争冒险产生了毛刺。

比较器我输出了三种指示, a>b, a=b, a<b; 在输入信号为某一种状态时对应的输出指示会被拉高。

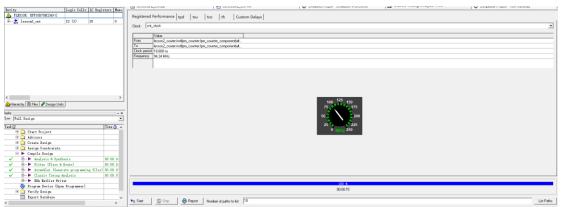
EPM7128SLC84-7 芯片下极限频率:



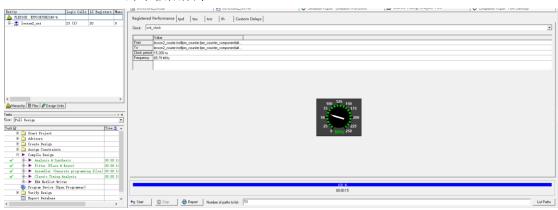
#### EPM7128SLC84-10 芯片下极限频率:



#### EPF10K70RC240-2 芯片下极限频率:



### EPF10K70RC240-4 芯片下极限频率:

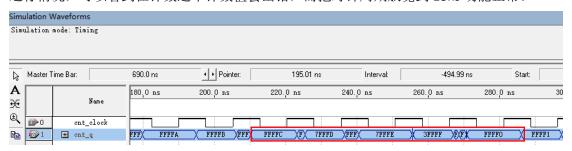


在同一系列芯片内部, MAX7000S 系列的 EPM7128SLC84-7 频率高于 EPM7128SLC84-10,

FLEX10K 系列的 EPF10K70RC240-2 频率高于 EPF10K70RC240-4; 这种情况是符合速度指标的。

在不同系列芯片间,MAX7000S 系列频率普遍高于 FLEX10K 系列,FLEX10K 系列速度指标-2 的芯片速率和 MAX7000S 速度指标-10 的芯片频率相近;但资源占用率是 MAX7000S 系列远高于 FLEX10K 系列,这反映了 FLEX10K 高密度的特点,但是从性能上 MAX7000S 更强。

从波形图来分析,MAX7000S 系列能够都能够运行到极限频率,并且波形完美。而 FLEX10K 系列芯片在极限频率下的仿真会出错,如下图是 EPF10K70RC240-4 在 15.2ns 周期的 运行情况,可以看到在计数途中计数值会出错,而把时钟周期放宽到 18ns 功能正常。



并且 FLEX10K 系列在有效输出数据之间会产生毛刺,这是由于数据变化时刻产生的竞争冒险现象导致输出波形有毛刺,这就要求取值的时刻要在 data valid 期间保证输出的可靠性。

