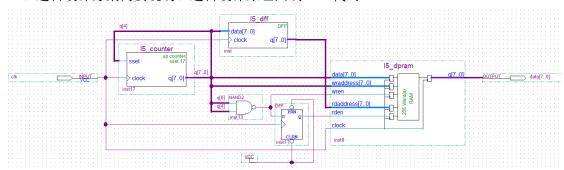
# PLD 实验

## 第五次实验报告

姓名:	董校廷	学号 <b>:</b>	SC21002046	完成时间:	2021-11-14

#### 一、实验要求

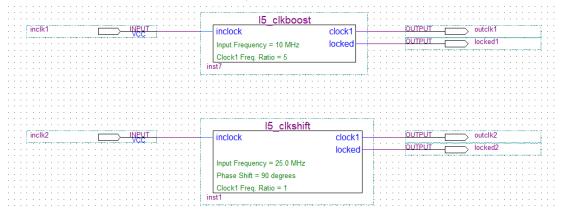
- 1. 选用 APEX20K300EQC240-1X,设计一个同步 DPRAM:数据宽度 8bit,地址线宽度 8bit。用一个 8bit 计数器的输出驱动 DPRAM 的 write address 和 read address。将 00H—10H 写入 DPRAM,在写的同时将其读出(pipeline)。仿真上述逻辑,将仿真波形打印出来。思考:如何避免读写冲突?
- 2. 选用 APEX20K300EQC240-1X,设计仿真 altclklock 的 clockboost 和 clockshift 特性。
- 3. 选用 APEX20K300EQC240-1X,设计仿真 CAM 的写和搜索特性,包括 don't care bit。
- 二、逻辑设计方案简要说明、逻辑设计原理图或 HDL 代码



利用计数器自动写入和读取的 DPRAM 模块如图,计数器产生 0x00~0x10 的值,作为写入地址和数据;在写入的同时进行读取,如果读写地址指向同一个地址就会造成冲突,因此考虑通过 D 触发器将计数器输出延迟一个周期发送到 DPRAM 读地址端口,这样可以先写后读,可以避免冲突问题。

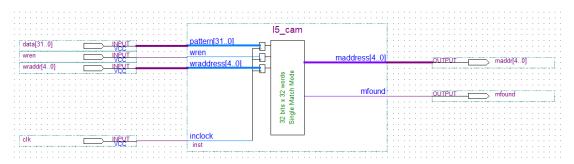
由于 0x00~0x10 都需要写入,因此当计数器在这个范围内写使能都有效,当计数器输出为 0x11 时写使能无效,因此将计数器输出的第 0 位和第 4 位与非之后作为写使能即可,同时读使能即为该信号通过 dff 延迟后的输出。

因此计数器在计数到 0x11 时保持该值即可,这里选用了计数器同步置数为 0x11,当计数器输出为 0x10 将 sset 置高,只需要 q[4]信号连接到计数器的同步置数端口即可。



上图为 altclklock 产生的两个模块;第一个是 clockboost,输入时钟频率为 10Mhz,比例

参数为 5,模块将输出倍频后频率为 50Mhz 的时钟;第二个是 clockshift,Phase Shift = 90 degrees,输出后时钟为输入信号的相位偏移 90 度后的时钟信号。



上图为 CAM 的使用,通过写地址和写使能,配合数据端口可以将数据写入 CAM,当写使能无效,会对数据端口的数据进行查找,如果查找到对应数值,将会返回其地址,mfound 指示是否查找到对应数据。

#### 三、仿真图【仿真截图及仿真时序波形文件】

	ion Waveforms tion mode: Timing												^ ~
Ŋ Ma	ster Time Bar:	0 ps	<ul> <li>Pointer:</li> </ul>	163.78 n	s	Interval	163.78 ns	Sta	t		End		
A Æ	Name	0 ps 40.0 ns	80.0 ns	120. <sub>0</sub> ns	160.0 ns	200. 0 ns	240.0 ns	280.0 ns	320.0 ns	360. 0 ns	400. <sub>0</sub> ns	440.0 ns	480.0 ns
		00 W FF	₩ 00 \ 01 \	(02 ) (03 )()	04 \ 05	XX 06 X 07 XX	08 X 09 XX	OA OB XX	OC X OB XX	OE OF X		10	

DPRAM 仿真是自动读写的,因此这里提供 50Mhz 的时钟信号进行仿真。数据会通过数据输出端口自动读出。

		Vaveforms node: Timing												^
														V
D <sub>6</sub> M	aster T	Time Bar:	0 ps	Pointer:	66.01 ns		Interval:	66.01 ns	Sta	t		End		
A	$\neg$	_	0 ps 80.0 ns	160. <sub>0</sub> ns	240.0 ns	320. 0 ns	400.0 ns	480.0 ns	560.0 ns	640.0 ns	720.0 ns	800.0 ns	880.0 ns	960.0 ns
€		Hane	0 ps											
(A)	<b>№</b> 0	inelk1							ш					
44 0 0 0, 0	<b>∌</b> 1	outclk1	*************	⋙⋙										
44	<b>∌</b> 2	locked1												
99 E	<b>№</b> 3	inclk2				лшп								
m, €	∌4	outclk2				பாப			யாயா				шпшг	
→ <u>«</u>	<b>∌</b> 5	locked2												

CLOCKLOCK 仿真是输入两个时钟信号,会产生倍频、相移后的时钟。

Sim	nulation \	Waveforms								
Sin	nulation	node: Timing								^
										V
De	Master 1	Time Bar:	0 ps Poi	nter: 1.65 ns	Interval:	1.65 ns	Start:		End	
A				0 ns 240.0 ns 320.0	ns 400.0 ns	480.0 ns	560.0 ns 640.0 ns	720. 0 ns 80	0.0 ns 880.0 ns	960.0 ns
€		Nane	D bz							
ⓐ,		clk								
Be	<u>i</u> 1 <u>i</u> 2	wren								
	<b>■</b> 2	₩ wraddr	01 X	02 X 03	Х			00		
44	<b>₽</b> 8	<b>⊞</b> data	11111111 X 222222	22 X222222FX 33333333	X 22222220 X 222222	21   22222222   222222	223   22222224   22222225   222	22226   22222227   2222222	8 (22222229 (11111111 )(33	33333 (00000000)
m,		⊞ naddr	00 X 01 X00	02 (00) 02 ( 0	) X (83 X (	00 X	02 X 00	X 02 X	00 X 01	(03 )(50)
-	€ 47	mfound						ш		

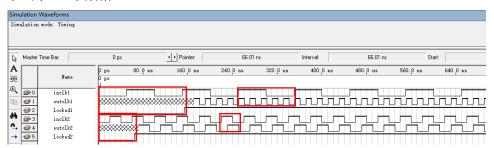
CAM 仿真先写入一些数据,再对一些数据进行查询、匹配地址。

### 四、仿真结果分析

Simulation Waveforms											
Simulation mode: Timir	ag										^
											~
Master Time Bar:	0 ps		<ul> <li>Pointer:</li> </ul>	163.78 ns	Interval	163.78 ns	Start		End		
200	Ops Ops	40.0 ns	80.0 ns	120. <sub>0</sub> ns 16	0.0 ns 200.0 ns	240.0 ns	280.0 ns 320.0 ns	360.0 ns	400.0 ns	440.0 ns	480.0 ns
el clk	( ** **)	YF.	( 00 ) 01 )X	02 X 03 XX 04	X 05 XX 06 X 07	<b>X</b> (X 08 X 09 XX	OA X OB XX OC X OD	X OE X OF XX	7-7	10	

DPRAM 实验,根据设计的避免冲突的自动读写方法,可以看到输出在初始启动 0xff 的无效

输出之后,开始按顺序读取 RAM 的数据,从 0x00~0x10, RAM 中写入了 17 个数据,顺序读取出了 17 个数据。



CLOCKLOCK 实验,可以看到输出时钟 outclk1 成功输出了五倍频后的时钟,输出 outclk2 对输入信号做了 90 度的相移,图中可以看出满足要求。

locked 信号检测输入时钟是否锁定,根据仿真图可以看出,当相同频率时钟信号的第二个上升沿到来即为锁定,locked 会被拉高。



CAM 实验,向地址 0x01 写入 0x11111111,向地址 0x03 写入 0x33333333,这些都需要两个时钟周期,向地址 0x02 写入 0x22222222,同时写入 don't care bit,此时需要三个时钟周期,第三个周期的数据为 0x2222222f,通过比较可知,don't care bit 为第 0、2、3 位(0b0010-0b1111)。

接着进行数据查询,首先对 0x22222220~0x2222229 进行查询,可知,只有最低四位为 0bxx1x 才可以查询到,即 0x2、0x3、0x6、0x7,图中可以看到查询结果与预期一致;接着对 0x11111111 和 0x33333333 进行查询,得到地址 0x01 和 0x03,同时 mfound 也指示匹配成功。