

تمرین 7-Segment

فاطمه علی‌ملکی

مهدی حق‌وردی

۲۸ آذر ۱۴۰۲

فهرست مطالب

۱	اتصال چهار 7-Segment به برد FPGA	۱
۲	توضیح فایل‌های VHDL	۲
۲	فایل clock_divide.vhd	۱.۲
۲	فایل decodeSevenSeg.vhd	۲.۲
۲	فایل sevenSegment.vhd	۳.۲
۳	چالش‌ها	۳
۲	چالش اول	۱.۳
۲	چالش دوم	۲.۳
۲	چالش سوم	۳.۳

۱ اتصال چهار 7-Segment به برد FPGA

توضیحات سایت^۱:

آیا ما از ۴ عدد 7-Segment استفاده کنیم و هر 7-Segment، ۹ پایه احتیاج داشته باشد باید برای نمایش مقدار مورد نظر از ۳۶ پایه fpga استفاده کنیم؟

خیر این‌گونه نیست. باید گفت که پایه‌های enable هر کدام به یک پورت متصل شود ولی بر فرض مثال پایه a هر چهار 7-Segment باید تنها به یک پورت متصل شود. باید به این نکته توجه کرد که چشم انسان توانایی دیدن روشن خاموش شدن مداوم یک LED را تا فرکانس (سرعت) مشخصی دارد و اگر سرعت خاموش و روشن شدن آن بر فرض مثال به ۸ میلی ثانیه برسد چشم انسان قادر به تشخیص دقیق آن نخواهد بود. حال فرض کنیم عبارت ۲۰۲۰ قرار است توسط ۴ عدد 7-Segment نمایش داده شود ابتدا در لحظه صفر پایه enable اولی وصل میشود و بقیه enable ها قطع میشوند و پورت های a, b, c, d, e, f, g مقدار صفر را نمایش میدهند و بعد از ۸ میلی ثانیه (دلخواه) enable دوم فعال و بقیه غیر فعال میشوند و پورتهای مقدار دو را نمایش میدهند، اگر همین‌گونه این کار ادامه یابد و تکرار شود چشم ما تنها مقادیر ۲۰۲۰ را خواهد دید تنها با تعداد پورت مصرفی کمتر.

^۱<https://posedge.ir/1399/07/21/connect-7segment-to-fpga/>

”به طور خلاصه در یک لحظه تنها یکی از 7-Segment ها، روشن است اما چون تعویض به سرعت اتفاق می‌افتد، چشم ما توانایی دیدن روشن و خاموش شدن آنها را ندارد.“

۲ توضیح فایل‌های VHDL

۱.۲ فایل clock_divide.vhd

۲.۲ فایل decodeSevenSeg.vhd

۳.۲ فایل sevenSegment.vhd

۳ چالش‌ها

در این قسمت به توضیح چالش‌های مطرح شده، می‌پردازیم.

۱.۳ چالش اول

طبق توضیحاتی که در (۱) آمد، ما باید در فایل clock_divide.vhd تغییراتی را اعمال کنیم که تعداد بار روشن شدن هر یک از 7-Segment ها آنقدری زیاد بشود، که چشم انسان قابلیت تشخیص روشن خاموش شدن آنها را نداشته باشد. این یعنی عددهایی که در این فایل هستند باید کوچک‌تر شوند تا سرعت خاموش روشن شدن افزایش یابد. چون در زمان نوشتن این گزارش هنوز آزمایش را انجام نداده‌ایم عدد دقیق آن را نمیدانیم^۲ اما مکان کدهایی که باید تغییر کنند در تصویر ۱(ا) آمده‌اند.^۳

۲.۳ چالش دوم

همانطور که در قسمت‌های ۲.۲ و ۳.۲، نحوه‌ی عملکرد توضیح داده شد، پس باید طبق تصویر ۱(ب) تغییراتی را اعمال کنیم.

۳.۳ چالش سوم

فهرست تصاویر

^۲ در زمان آزمایش عدد آنها را روی فایل گزارش با خودکار می‌شود.
^۳ پراگم‌ها

signal clk_100Hz: std_logic;	signal clk_100Hz: std_logic;	29	29
signal s_reg: unsigned(1 downto 0) ;	signal s_reg: unsigned(1 downto 0) ;	30	30
signal s_next: unsigned(1 downto 0);	signal s_next: unsigned(1 downto 0);	31	31
signal myNum: std_logic_vector(15 downto 0):= "1100010100110001";	signal myNum: std_logic_vector(15 downto 0):= "1100010100110001";	32	32
signal snum: std_logic_vector(3 downto 0);	signal snum: std_logic_vector(3 downto 0);	33	33
		34	34
@@ 20,14 begin	@@ 20,14 begin	20	20
end process;	end process;	21	21
process(s_reg)	process(s_reg)	22	22
begin	begin	23	23
if(s_reg = 1e6) then	if(s_reg = ()) then	24	24
s_next <= (others => '0');	s_next <= (others => '0');	25	25
else s_next <= s_reg + 1;	else s_next <= s_reg + 1;	26	26
end if;	end if;	27	27
end process;	end process;	28	28
process(s_next)	process(s_next)	29	29
begin	begin	30	30
if(s_next < 5e5) then	if(s_next < ()) then	31	31
oeb_next <= '1';	oeb_next <= '1';	32	32
else oeb_next <= '0';	else oeb_next <= '0';	33	33
end if;	end if;		

ب. عدد 135C

ا. تغییرات کلای