

## تمرین 7-Segement

فاطمه علی‌ملکی  
مهدی حق‌وردی

۲۸ آذر ۱۴۰۲

### فهرست مطالب

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

## ۱ اتصال چهار 7-Segment به برد FPGA

### توضیحات سایت<sup>۱</sup>:

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 پایه های 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 و 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.

”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

این فایل مانند clock\_divide.vhd آزمایش قبلی‌ست.

### ۲.۲ فایل decodeSevenSeg.vhd

این فایل طبق جدولی که برای 7-Segment های برد پازج، در وبسایت<sup>۲</sup> آورده شده است، ۱۶ مقدار ۰ تا ۹ و حروف a تا f را توسط یک دیکودر ۴ به ۷ در خروجی قرار می‌دهد.

### ۳.۲ فایل sevenSegment.vhd

این فایل، فایل اصلی ماست. فایل از دو component عه clock\_divide و decodeSevenSeg استفاده می‌کند.

فایل ۳ فرایند دارد.

• یک فلیپ فلاپ D

• بلوک next state

<sup>1</sup><https://posedge.ir/1399/07/21/connect-7segment-to-fpga/>

<sup>2</sup><https://posedge.ir/1399/07/21/connect-7segment-to-fpga/>

### • بلوک output logic

این بلوک، بلوک مهم در این معماری است. بر اساس خروجی فلیپ فلاپ، این بلوک تصمیم می‌گیرد که کدام یک از 7-Segment ها روشن شود. که از مقدار ۰۰ تا ۱۰ گرفته و قسمت else آن، چهارمین 7-Segment را روشن می‌کند.

در هر بلاک if, elsif و یا else ابتدا آدرس 7-Segment روی سیگنال خروجی an قرار می‌گیرد و سپس ۴ بیت از عددی که طبق decoderSevenSeg کد شده است را روی snum قرار می‌دهد و در نهایت خروجی decodeSevenSeg را روی سیگنال sseg قرار می‌دهد.

## 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 ندادیم عدد دقیق آن را نمی‌دانیم<sup>۳</sup> اما مکان کدهایی که باید تغییر کنند در تصویر ۱(ا) آمده‌اند.<sup>۴</sup>

### 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

برای خاموش کردن 7-Segment ها (چون طبق کدی که به ما دادید، Common Cathode هستند) باید تمامی ورودی‌های آنها را ۱ کنیم. پس همانطور که در تصویر ۱(ج) مشخص است، یک شرط دیگر با مقدار ۱۱۱۱ اضافه می‌کنیم و خروجی را ۱۱۱۱۱۱۱ قرار می‌دهیم. سپس مقداری که در ۲.۳ تغییر دادیم با دوباره تغییر می‌دهیم و مانند تصویر ۲ بجای مقادیری که می‌خواهیم، ۱۱۱۱ می‌گذاریم تا خروجی آنها ۱۱۱۱۱۱۱ شود و خاموش شوند.

## فهرست تصاویر

۱	تغییرات clock_divide, decodeSevenSeg و sevenSegment . . .	۴
۲	تغییرات در sevenSegment.vhd . . . . .	۵

<sup>۳</sup> در زمان آزمایش عدد آنها را روی فایل گزارش با خودکار می‌شود.  
<sup>۴</sup> پراگم‌ها



```

signal clk_100Hz: std_logic;
signal s_reg: unsigned(1 downto 0) ;
signal s_next: unsigned(1 downto 0);
signal myNum: std_logic_vector(15 downto 0):= "1111010111110001";
-- X 5 X 1
signal snum: std_logic_vector(3 downto 0);

```

شکل ۲: تغییرات در sevenSegment.vhd