



دانشگاه صنعتی امیرکبیر  
(پلی‌تکنیک تهران)  
دانشکده مهندسی برق

## گزارش پروژه درس مدارهای منطقی برنامه پذیر

دکتر شریعتمدار

نگارش

دانیال فتح پور 9923059  
عرفان قصری 9923061

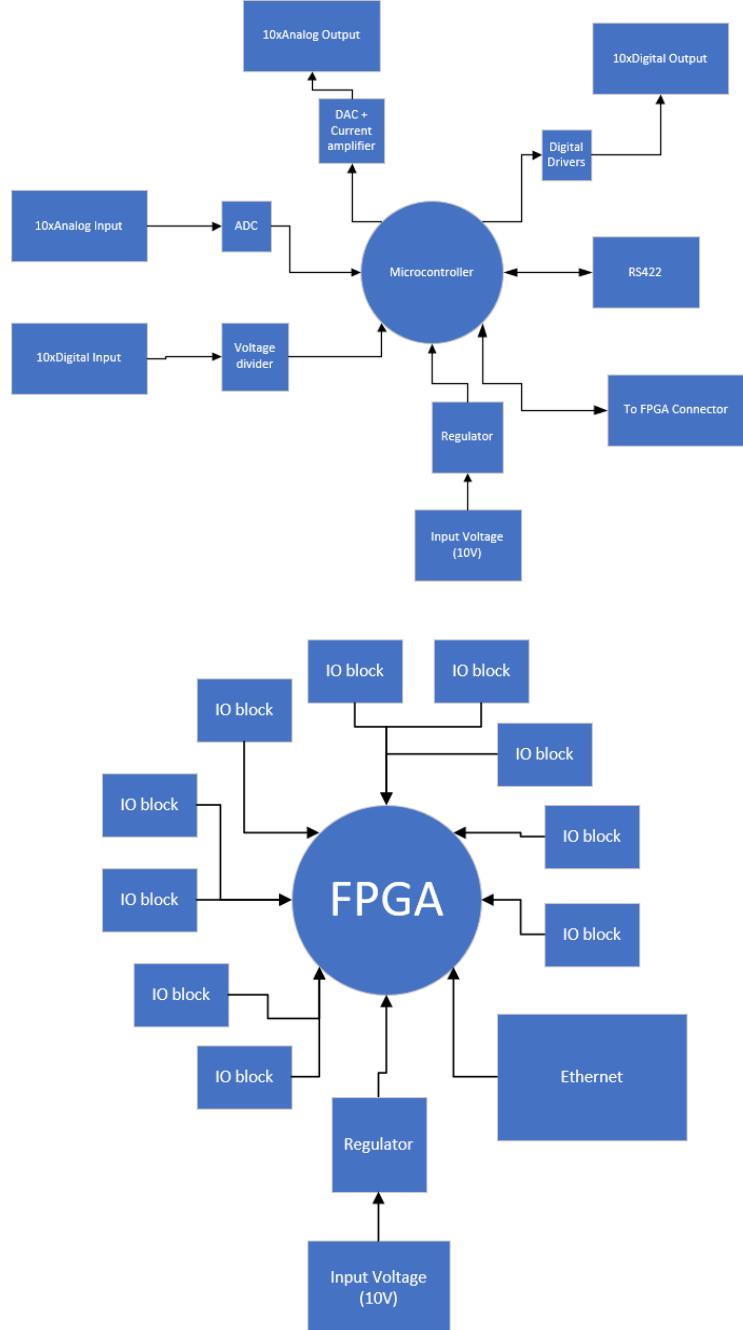
زمستان 1402

# فهرست محتوا

2 .....	بخش اول:
3 .....	انتخاب قطعات:
3 .....	انتخاب میکرو:
4 .....	انتخاب تراشه ADC و DAC :
4 .....	انتخاب درایور دیجیتال:
4 .....	انتخاب درایور RS422:
5 .....	انتخاب درایور Ethernet:
5 .....	انتخاب هسته اصلی:
6 .....	انتخاب تغذیه:
9 .....	ترسیم شماتیک:
9 .....	بخش نوان:
11 .....	بخش درایور ethernet:
12 .....	بخش درایور RS422:
12 .....	بخش راه انداز FPGA:
13 .....	بخش راه انداز FPGA:
13 .....	بخش های IO:
17 .....	بخش دوم:
18 .....	بخش سوم:
18 .....	تعریف پین ها:
22 .....	کدنویسی RS422:

## بخش اول: طراحی کلی مدار:

سیستم را به طور کلی به دو بخش برد اصلی (برد دارای FPGA، Ethernet، و قابلیت اتصال به برد های IO) و 10 عدد برد IO (قابلیت اتصال به 10 ورودی دیجیتال، 10 خروجی دیجیتال، 10 ورودی آنالوگ، 10 خروجی آنالوگ و RS422) تقسیم می کنیم که بلوک دیاگرام آن ها به صورت زیر است:

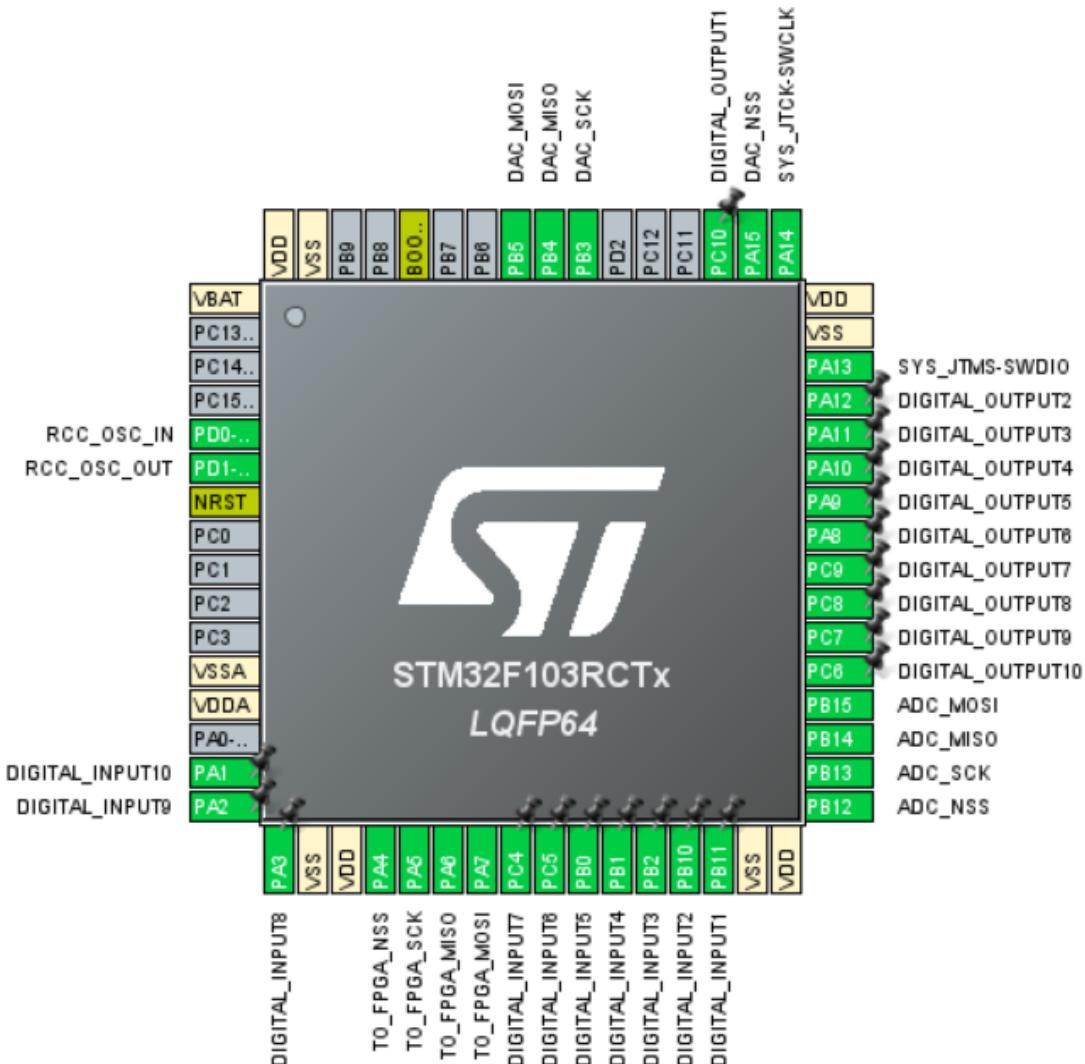


## انتخاب قطعات:

## انتخاب میکرو:

برای انتخاب میکرو مد نظر در هر IO Board بر اساس طرح کلی اولیه به برنامه STM32cubeMX مراجعه کرده و میکرو مد نظر را انتخاب می کنیم. در اینجا میکرو STM32F103RCT6 انتخاب شد. در اینجا خواسته ها عبارتند از:

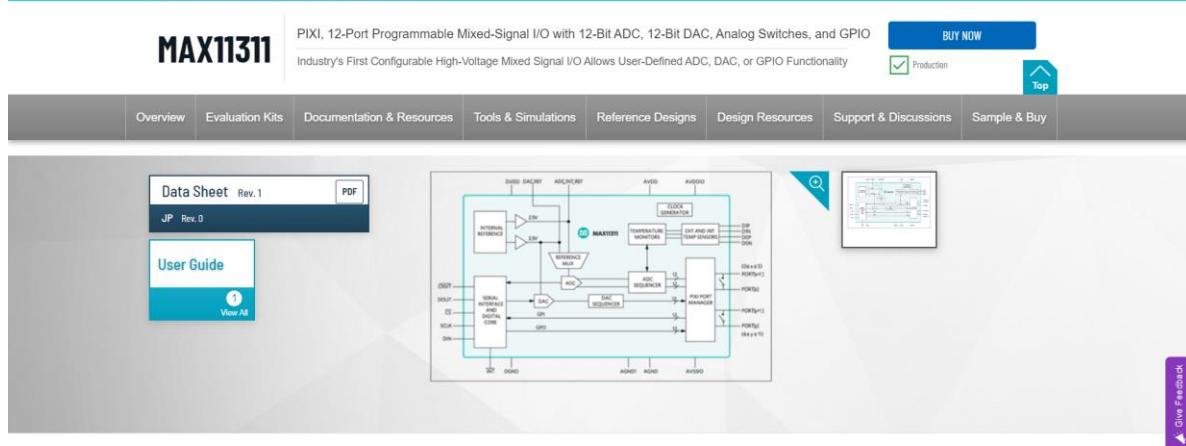
- 3 عدد SPI (یکی برای ارتباط با برد اصلی، یکی برای ADC و یکی برای DAC)
  - تعداد مناسبی پین GPIO برای اتصال به بقیه قسمت ها



همانطور که مشخص است بر اساس طراحی اولیه بالا تمامی خواسته های برد برطرف می شود.

## انتخاب تراشه ADC و DAC :

ابندا برای انتخاب تراشه ADC از طریق سایت شرکت Analog Devices اقدام می کنیم و بر اساس نیاز های این برد تراشه MAX11311 را انتخاب می کنیم (حداقل 10 کانال داشته باشد و سطح ولتاژ  $\pm 5$  را تحمل کند.) که متوجه می شویم هم می تواند ADC و هم DAC باشد و جریان خروجی آن 25 میلی آمپر است که بنابراین نیاز است از تقویت کننده جریان (یک مدار پوش پول) استفاده کنیم.



### Overview

Features and Benefits | Product Details

- 12 Configurable Mixed-Signal Ports Maximize Design
- 0 to +2.5V GPI Programmable Threshold

### Product Categories

D/A Converters (DAC)  
Single-Supply Voltage Output D/A Converters

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

...

## THVD1400 ACTIVE

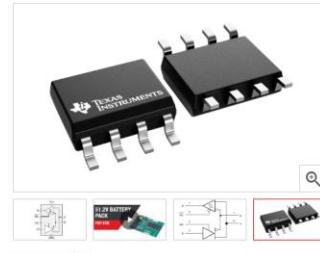
3.3-V to 5-V, 500-Kbps, RS-485 transceiver in small package option with ±12-kV IEC ESD pro

[Notifications](#)
[Order now](#)
[DATA SHEET](#)
[THVD1400, THVD1420 3.3-V to 5-V RS-485 Transceivers in Small Package with ±12-kV IEC ESD Protection datasheet \(Rev. B\)](#)
[PDF](#) | [HTML](#)
[Product details](#) | [Technical documentation](#) | [Design & development](#) | [Ordering & quality](#) | [Support & training](#)

### Product details

[Parameters](#) [Package | Pins | Size](#) [Features](#) [Description](#)

Number of receivers	1
Number of transmitters	1
Duplex	Half
Supply voltage (nom) (V)	3.3, 5
Signaling rate (max) (MBits)	0.5
IEC 61000-4-2 contact (±V)	12000
Fault protection (V)	-16 to 16



### Ethernet : انتخاب درایور

برای انتخاب این تراشه از PCB اپن سورس OpenRex استفاده می کنیم و همان تراشه ای که در آنجا استفاده شده را قرار می دهیم که تراشه KSZ9031RNXIA است.

[Products](#) | [Manufacturers](#) | [Services & Tools](#) | [Technical Resources](#) | [Help](#)
[Account & Orders](#)
[0](#)
[All Products](#) > [Semiconductors](#) > [Communication & Networking ICs](#) > [Ethernet ICs](#) > Microchip Technology KSZ9031RNXIA

[See an Error?](#)

#### KSZ9031RNXIA



Images are for reference only  
See Product Specifications

[Share](#)
**Mouser No:** 998-KSZ9031RNXIA

**Mfr. No:** KSZ9031RNXIA

**Mfr.:** Microchip Technology

**Customer No:** Customer No

**Description:** Ethernet ICs 1-Port GigabitEthernet PHY

[View in Development Tools Selector](#)
**Datasheet:** [KSZ9031RNXIA Datasheet \(PDF\)](#)
**ECAD Model:** [PCB Symbol, Footprint](#)

Download the free Library Loader to convert this file for your ECAD Tool. [Learn more about the ECAD Model](#).

**More Information:** [Learn more about Microchip Technology KSZ9031RNXIA](#)
 Compare Product

[Add To Project](#) | [Add Notes](#)

#### In Stock: 51.417

**Stock:** 51.417 Can Dispatch Immediately

**Factory Lead Time:** 14 Weeks

**Enter Quantity:**  Minimum: 1 Multiples: 1

[Buy](#)

#### Pricing (EUR)

Qty.	Unit Price	Ext. Price
1	6,45 €	6,45 €
10	5,93 €	59,30 €
25	5,31 €	132,75 €
100	5,01 €	501,00 €
250	4,67 €	1.214,20 €
520	4,36 €	2.262,00 €
1.040	4,34 €	4.513,60 €
5.200	Counts	

### انتخاب هسته اصلی:

برای انتخاب هسته اصلی از نرم افزار Vivado استفاده می کنیم و بر اساس خواسته های مسئله (10 عدد برد IO و یک عدد اترنت و 10 عدد ارتباط RS422 که در مجموع به حدود 100 عدد پایه نیازمندیم.) یک FPGA را انتخاب می کنیم. دلیل انتخاب تراشه FPGA به عنوان هسته اصلی سرعت آن و قابلیت موازی سازی آن است. نکته دیگر در مورد انتخاب FPGA ها این است که تا حد ممکن می خواهیم از تراشه ای استفاده کنیم که برد آماده آن وجود داشته باشد تا بتوانیم مدار های جانبی تراشه مرکزی را به درستی طراحی کنیم. بنابر تمام نکات گفته شده به سری برد های PicoZed از شرکت

می رسم و برد 7Z010 را به دلیل اینکه 113 پایه IO دارد (خود تراشه دارای 225 پین است) انتخاب می کنیم که دارای تراشه XC7Z010-1CLG225 است.

[Boards / Avnet PicoZed](#)

### Avnet PicoZed



Click to Enlarge

by: Avnet

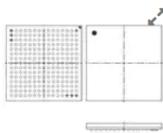
**AVNET**  
Reach Further™

PicoZed™ is a highly flexible, rugged, System-On-Module, or SOM that is based on the AMD Zynq 7000 SoC. It offers designers the flexibility to migrate between the 7010, 7015, 7020, and 7030 Zynq 7000 SoC devices in a pin-compatible footprint. The PicoZed module contains the common functions required to support the core of most SoC designs, including memory, configuration, Ethernet, USB, and clocks. It provides easy access to over 100 user I/O pins through three I/O connectors on the backside of the module. These connectors also support access to dedicated interfaces for Ethernet, USB, JTAG, power and other control signals, as well as the GTP/GTX transceivers on the 7015/7030 models. The transceiver based 7015 and 7030 versions of PicoZed are a superset of the 7010/7020 version, adding four high-speed serial transceiver ports to the I/O connectors. Designers can simply design their own carrier card, plug-in PicoZed, and start their application development with a proven Zynq 7000 SoC sub-system.

The PicoZed kit is supported by the [www.picozed.org](http://www.picozed.org) community website where users can download kit documentation and reference designs as well as collaborate with other engineers also working on Zynq designs.

Price: \$178.00  
 Part Number: AES-Z7PZ-7Z010-SOM-G/REV-E  
 Device Support: Zynq-7000  
 Partner Tier: Premier  
[View Partner Profile](#)

**XC7Z010-1CLG225C**

<b>AMD XILINX</b>  	<b>Mouser No:</b> 217-XC7Z010-1CLG225C <b>Mfr. No:</b> XC7Z010-1CLG225C <b>Mfr.:</b> Xilinx <b>Customer No:</b> Customer No <b>Description:</b> SoC FPGA XC7Z010-1CLG225C <b>Datasheet:</b> <a href="#">XC7Z010-1CLG225C Datasheet (PDF)</a> <b>More Information</b> <a href="#">Learn more about Xilinx XC7Z010-1CLG225C</a>
--	---

Images are for reference only  
See Product Specifications

[Share](#)

Compare Product      [Add To Project](#) | [Add Notes](#)

**In Stock: 9**

<b>Stock:</b>	9 Can Dispatch Immediately
<b>Factory Lead Time:</b>	108 Weeks <a href="#">?</a>
<b>Enter Quantity:</b>	Minimum: 1 Multiples: 1 <input type="button" value="Buy"/>

**Pricing (EUR)**

Qty.	Unit Price	Ext. Price
1	69.58 €	69.58 €

### انتخاب تغذیه:

بر اساس هسته اصلی، درایور اترنت، میکرو و تراشه آنالوگ به چند سطح ولتاژ DC نیازمندیم که برای هر کدام باید رگولاتور مربوطه را قرار دهیم. این تراشه رگولاتور ها را از سایت شرکت Texas instrument انتخاب می کنیم و در صورت توان از مدار های برد های اپن سورسی که در اختیار داریم کمک می گیریم.

**1V:**  
TLV62130RGT

Product Index > Integrated Circuits (ICs) > Power Management (PMIC) > Voltage Regulators - DC DC Switching Regulators > Texas Instruments TLV62130RGTT

### TLV62130RGTT

DigiKey Part Number	296-32407-2-ND - Tape & Reel (TR) 296-32407-1-ND - Cut Tape (CT) 296-32407-6-ND - Digi-Reel®
Manufacturer	Texas Instruments
Manufacturer Product Number	TLV62130RGTT
Description	IC REG BUCK ADJ 3A 16VQFN
Manufacturer Standard Lead Time	6 Weeks
Detailed Description	Buck Switching Regulator IC Positive Adjustable 0.9V / 1 Output 3A 16-VFQFN Exposed Pad
Customer Reference	<input type="text"/>
Datasheet	<a href="#">Datasheet</a>
EDA/CAD Models	<a href="#">TLV62130RGTT Models</a>

**In-Stock: 1,515**  
Can ship immediately  
**QUANTITY**  
  
**Add to List** **Add to Cart**

All prices are in USD  
**Cut Tape (CT) & Digi-Reel®**

QUANTITY	UNIT PRICE	EXT PRICE
1	\$1.97000	\$1.97
10	\$1.76600	\$17.66
25	\$1.66640	\$41.66
100	\$1.41990	\$141.99

\* All Digi-Reel orders will add a \$7.00 reeling fee.

**Tape & Reel (TR)**

QUANTITY	UNIT PRICE	EXT PRICE
----------	------------	-----------

## 1.2V: MIC33050-4YHL TR

DigiKey

All Products  Enter keyword or part #  Login or REGISTER | 0 item(s)

Product Index > Integrated Circuits (ICs) > Power Management (PMIC) > Voltage Regulators - DC DC Switching Regulators > Microchip Technology MIC33050-4YHL-TR

### MIC33050-4YHL-TR

DigiKey Part Number	576-3044-2-ND - Tape & Reel (TR) 576-3044-1-ND - Cut Tape (CT) 576-3044-6-ND - Digi-Reel®
Manufacturer	Microchip Technology
Manufacturer Product Number	MIC33050-4YHL-TR
Description	IC REG BUCK 1.2V 600MA 12MLF
Manufacturer Standard Lead Time	12 Weeks
Detailed Description	Buck Switching Regulator IC Positive Fixed 1.2V Output 600mA 12-VFDFN Exposed Pad, 12-MLF®
Customer Reference	<input type="text"/>
Datasheet	<a href="#">Datasheet</a>
EDA/CAD Models	<a href="#">MIC33050-4YHL-TR Models</a>

**0 In Stock**  
Check Lead Time  
Request Stock Notification  
**QUANTITY**  
  
**Add to List** **Add to Cart**

All prices are in USD  
**Cut Tape (CT) & Digi-Reel®**

QUANTITY	UNIT PRICE	EXT PRICE
1	\$2.06000	\$2.06
25	\$1.72000	\$43.00
100	\$1.56000	\$156.00

\* All Digi-Reel orders will add a \$7.00 reeling fee.

**Tape & Reel (TR)**

QUANTITY	UNIT PRICE	EXT PRICE
----------	------------	-----------

## 1.5V: TLV62130RGT

## 1.8V: TLV62130RGT

## 2.5V: TPS7A26

**TPS7A26** ACTIVE

500-mA, 18-V, ultra-low-IQ, high-accuracy, adjustable low-dropout voltage regulator with power good

[DATA SHEET](#)  [TPS7A26 500-mA, 18-V, Ultra-Low IQ, Low-Dropout Linear Voltage Regulator With Power-Good datasheet \(Rev. B\)](#) [PDF](#) | [HTML](#)

[Order now](#)

Product details | Technical documentation | Design & development | Ordering & quality | Support & training

### Product details

[Parameters](#) [Package | Pins | Size](#) [Features](#) [Description](#)

Output options	Adjustable Output, Fixed Output
Iout (max) (A)	0.5
Vin (max) (V)	18
Vin (min) (V)	2.4
Vout (max) (V)	17.41
Vout (min) (V)	1.24
Fixed output options (V)	1.25, 1.8, 2.5, 3.3, 5



## 3.3V: ISL8024AIRTAJZ-T7A

Products • Manufacturers Services & Tools Technical Resources Help Account & Orders •  [0](#)

All Products > Semiconductors > Power Management ICs > Voltage Regulators & Voltage Controllers > Switching Voltage Regulators > Renesas / Intersil ISL8024AIRTAJZ-T7A [See an Error?](#)

**ISL8024AIRTAJZ-T7A**

**intersil**

Images are for reference only  
See Product Specifications

[Share](#)

Mouser No:	968-ISL8024AIRTAJZT7
Mfr. No:	ISL8024AIRTAJZ-T7A
Mfr.:	Renesas / Intersil
Customer No:	Customer No
Description:	Switching Voltage Regulators 4A 2MHz ADJ VOUT LW IQ SYNCH BUCKG 250P
Lifecycle:	End of Life: Scheduled for obsolescence and will be discontinued by the manufacturer
Datasheet:	<a href="#">ISL8024AIRTAJZ-T7A Datasheet (PDF)</a>
ECAD Model:	PCB Symbol, Footprint & 3D Model
Download the free Library Loader to convert this file for your ECAD Tool. <a href="#">Learn more about the ECAD Model</a>	

**In Stock: 82**

Stock: 82 Can Dispatch Immediately

Quantities greater than 82 will be subject to minimum order requirements.

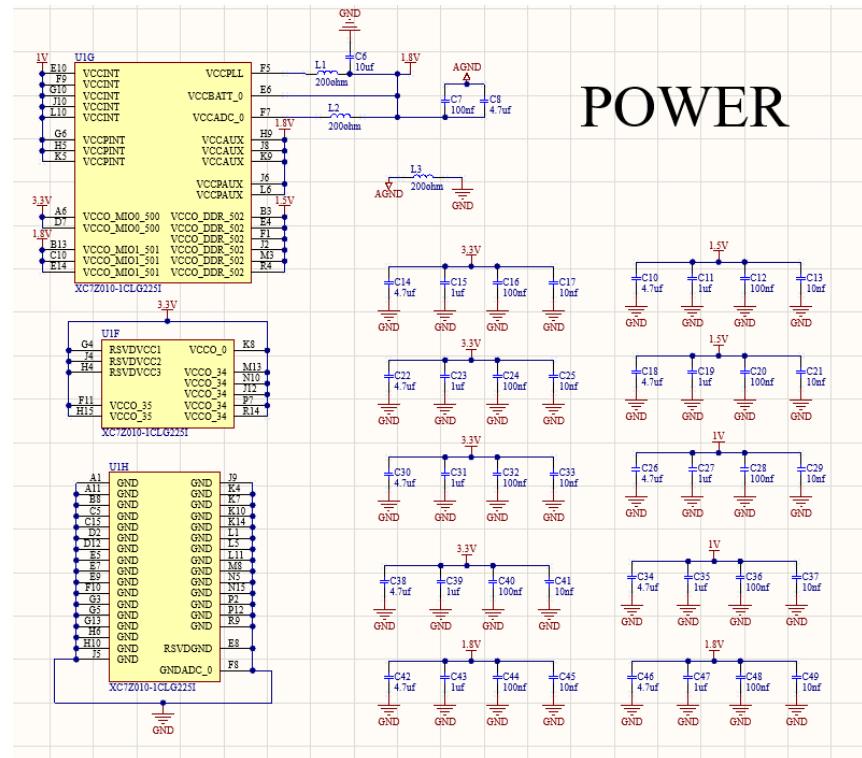
Mfr.: Minimum: 1 Multiples: 1

Enter Quantity:  [Buy](#)

**Pricing (EUR)**

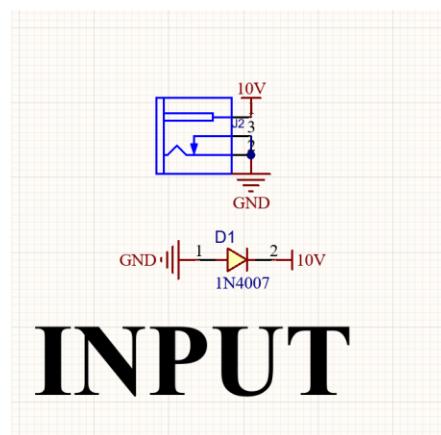
Qty.	Unit Price	Ext. Price
<b>Cut Tape / MouseReel™</b>		
1	2,19 €	2,19 €
10	1,86 €	18,60 €
100	1,59 €	159,00 €
<b>Full Reel (Order in multiples of 250)</b>		
250	1,50 €	375,00 €
500	1,43 €	715,00 €

## ترسیم شماتیک: بخش توان:



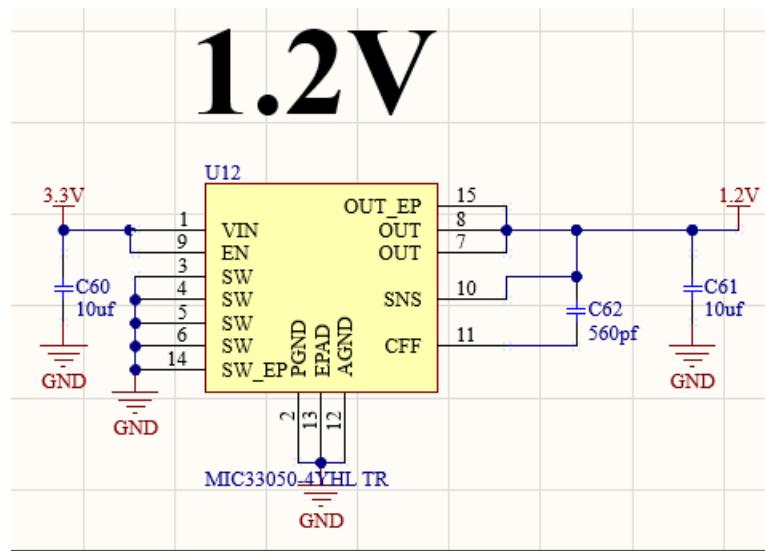
تصویر شماره ۱ : تامین توان تراشه اصلی

در این بخش از نمونه های ارایه شده در فایل های تدریس یاری استفاده کردیم، خازن های با ظرفیت ها متفاوت برای حذف نویز در فرکانس های متفاوت به دلیل ضریب کیفیت متفاوت خازن ها(ضرایب کمتر برای خازن های بزرگ تر) به کار رفته اند.



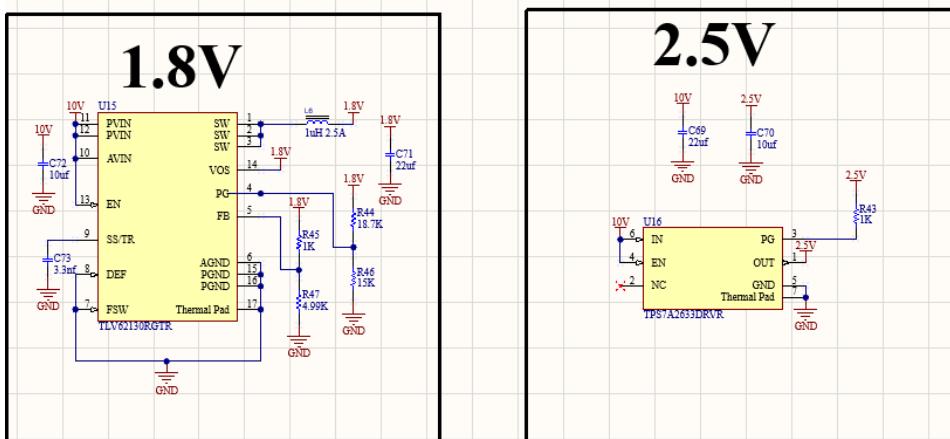
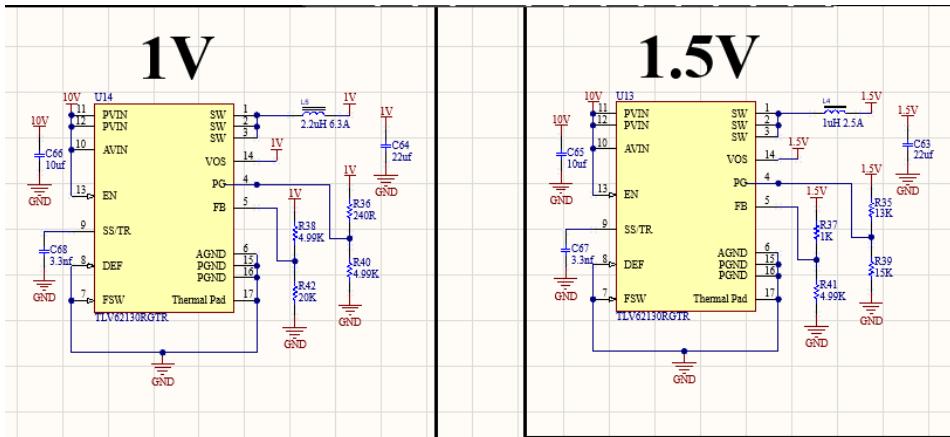
تصویر شماره ۲ : ورودی برق و دیود برای جلوگیری از آسیب بین مدار در صورت اتصال اشتباه قطب های مثبت و منفی

# 1.2V



تصویر شماره ۳ : تولید ولتاژ یک ولت

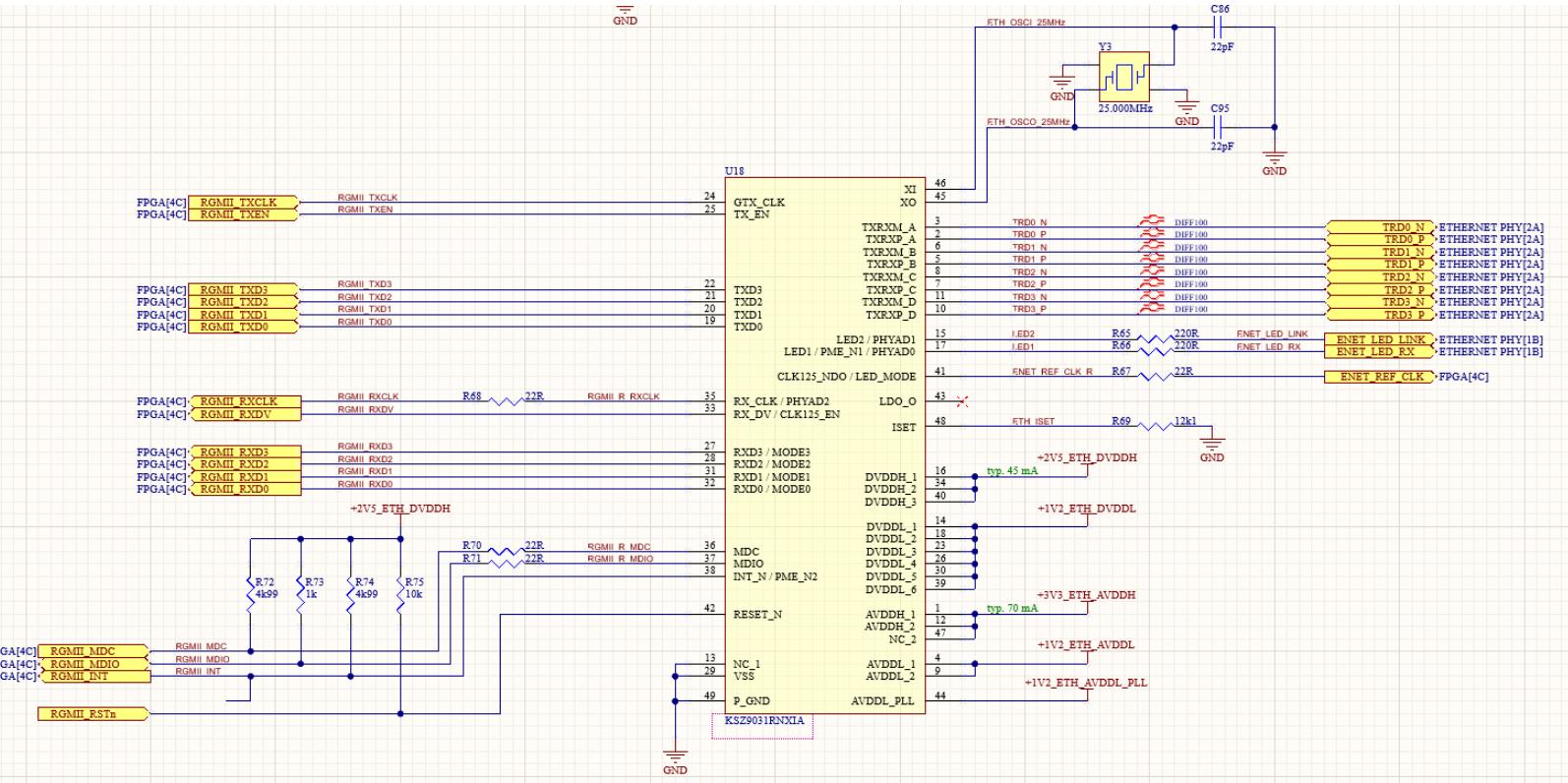
در این قسمت و قسمت های مشابه همانند مثال های ارایه شده در data-sheet تراشه بهره بردیم، خازن ها و سلف ها(بعض افریت ها) نیز به جهت جلوگیری از نویز قرار گرفته اند.



تصویر شماره ۴ : سایر سطوح ولتاژ

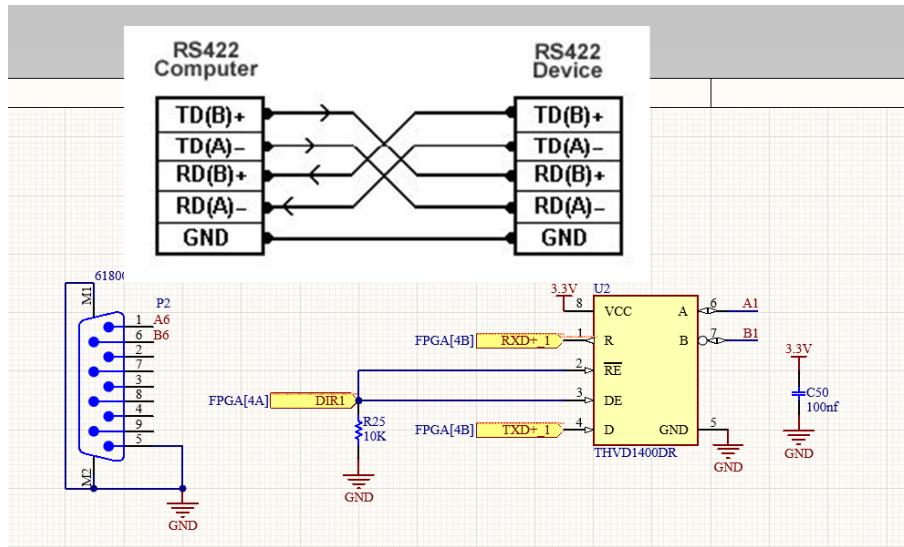
## بخش درایور : ethernet

برای پیکربندی درایور اترنت نیز از مثال خود دیتا شیت استفاده کردیم و تنها پورت های دینا را از یک طرف به تراشه و از طرف دیگر به پورت وصل کردیم:



تصویر شماره 5 : درایور اترنت به همراه نوسانگر مناسب

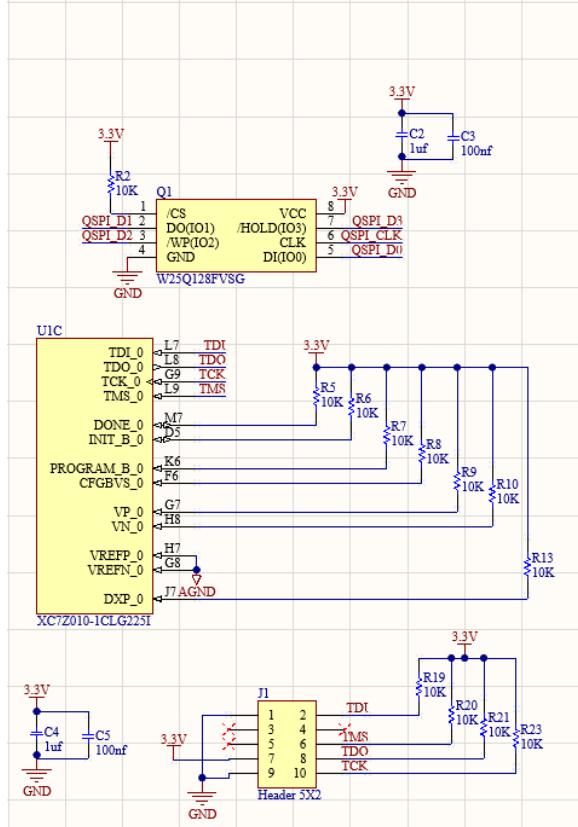
## بخش درایور RS422



تصویر شماره ۶ : مجموعه درایور و پورت مادگی مناسب

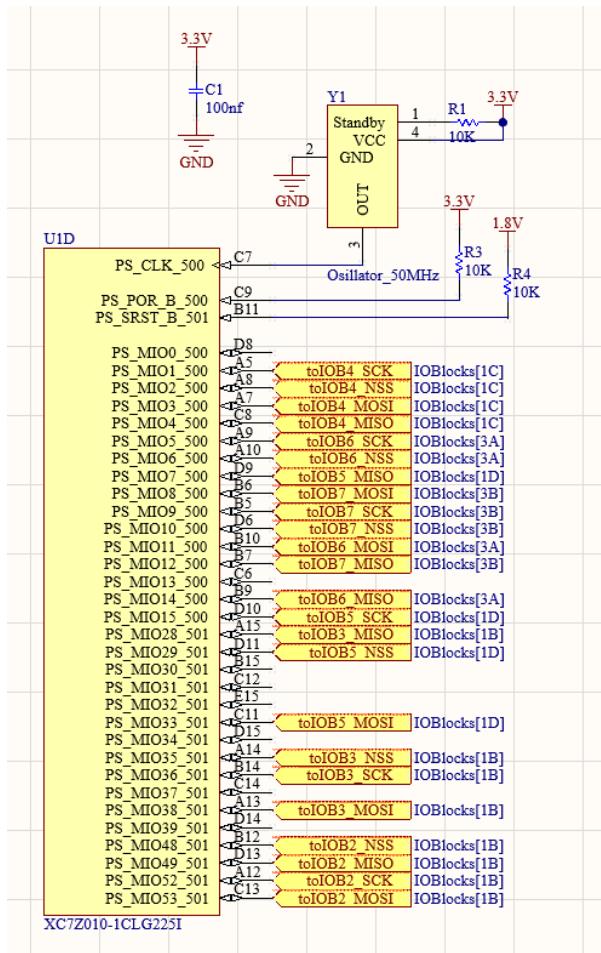
بخش راه انداز FPGA  
برای این بخش از یک حافظه کوچک استفاده کردیم:

## FPGA SETUP



## بخش راه انداز :FPGA

به دلیل شباht بخش های مختلف تراشه اصلی تنها یک بخش از آنرا در تصویر زیر آوردم، برخی پورت ها از پیش تابین شده بودند که آنها را به مقادیر مناسبشان متصل کردیم، سایر پورت هارا با توجه به اطلاعاتی که توسط Vivado درباره نوع پورت(برای مثال برخی پورت های **read only** بودند) تامین کرده بود به قسمت های مناسب( IO-block ) برای سیگنال های آنalog و دیجیتال که بخش های بعدی به آن میپردازم و درایور های RS422 و Ethernet به طور مستقیم متصل کردیم.



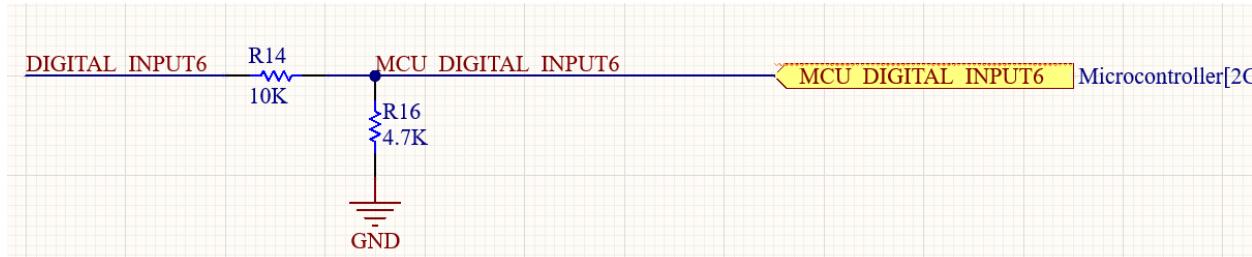
تصویر شماره 7 : قسمتی از خود تراشه

## بخش های IO :

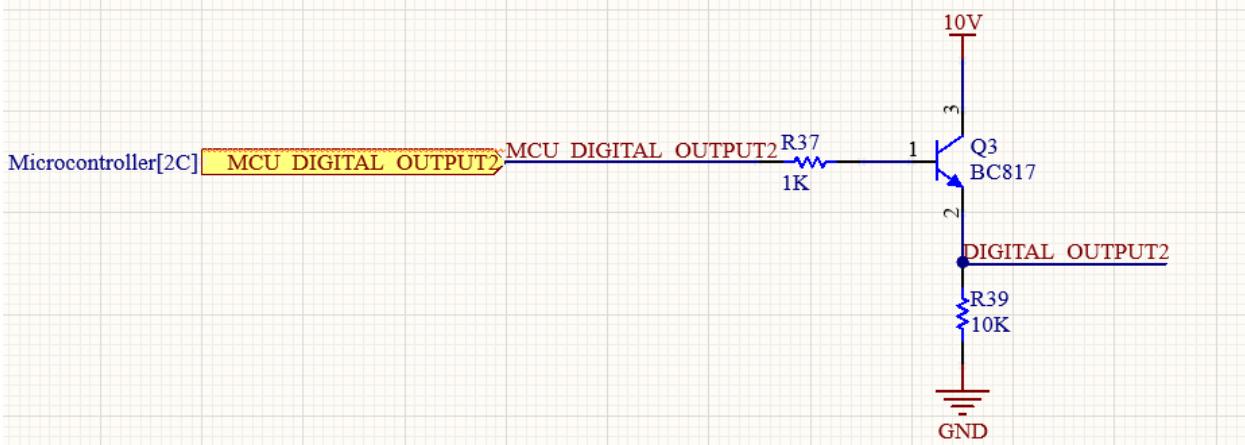
10 نمونه از بخش های IO را به برد اصلی وصل خواهیم کرد و هر برد IO بخش های زیر را دارد:

Power .1

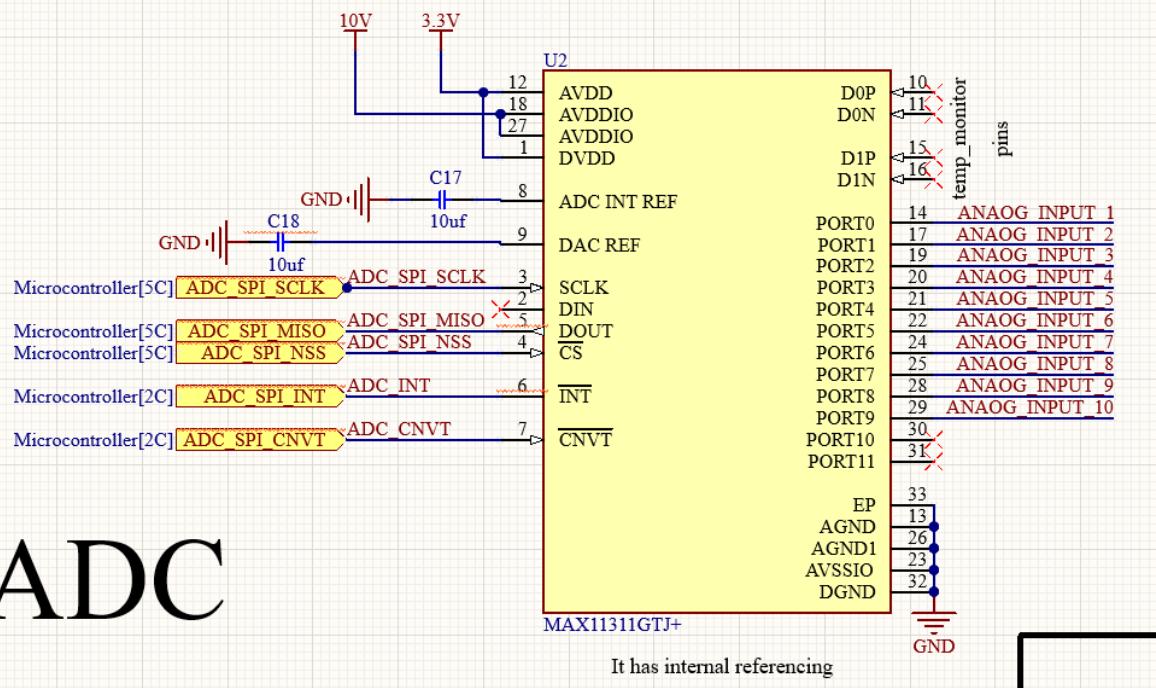
.2. مبدل ولتاژ برای تامین ولتاژ مناسب میکرو:



.3. مدار pull-down شده برای تامین جریان خروجی دیجیتال:

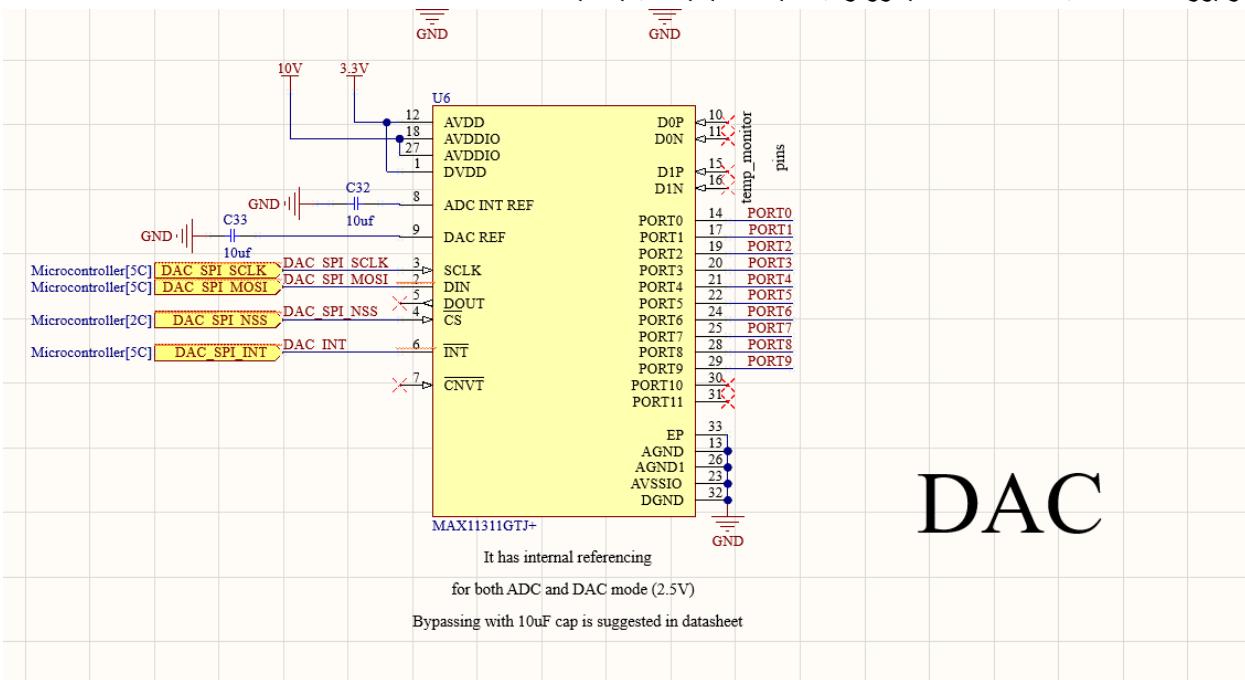


.4. درایور مناسب برای سیگنال های ورودی که اطلاعات را با SPI به تراشه میکرو منتقل میکند(با رزولوشن 16 بیت):



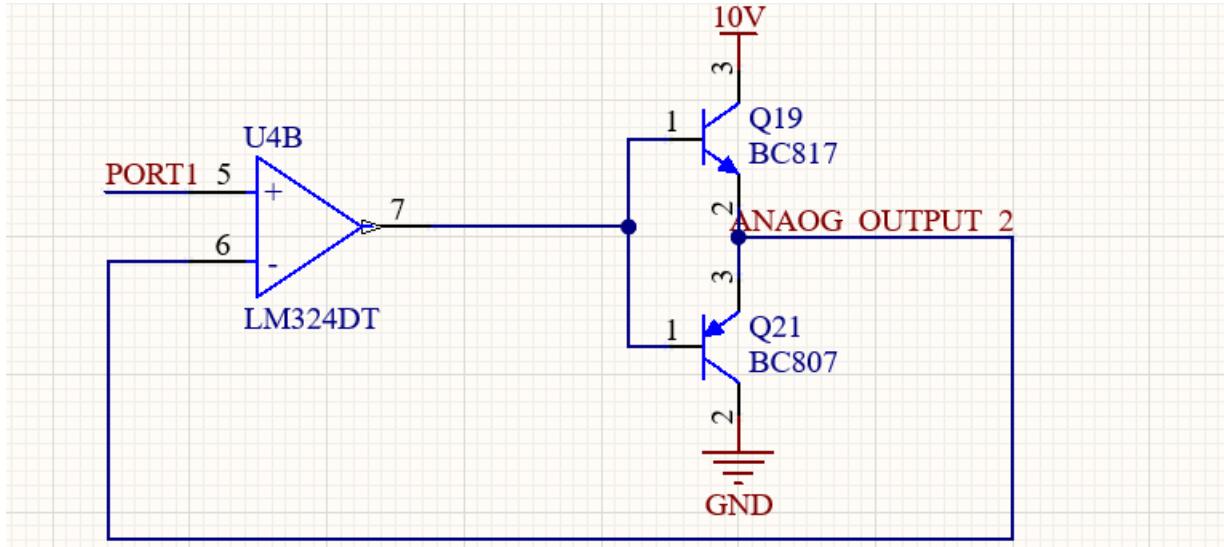
# ADC

.5 درایور ADC مناسب که اطلاعات میکرو را به سیگنال دیجیتال تبدیل میکند:

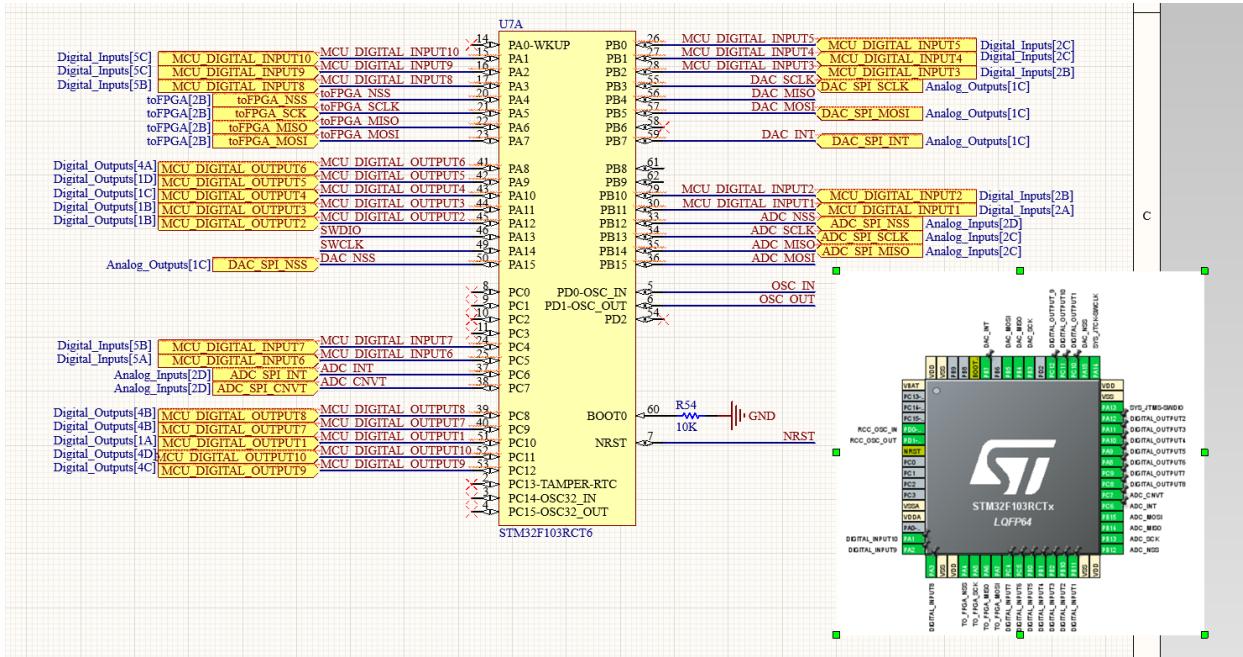


# DAC

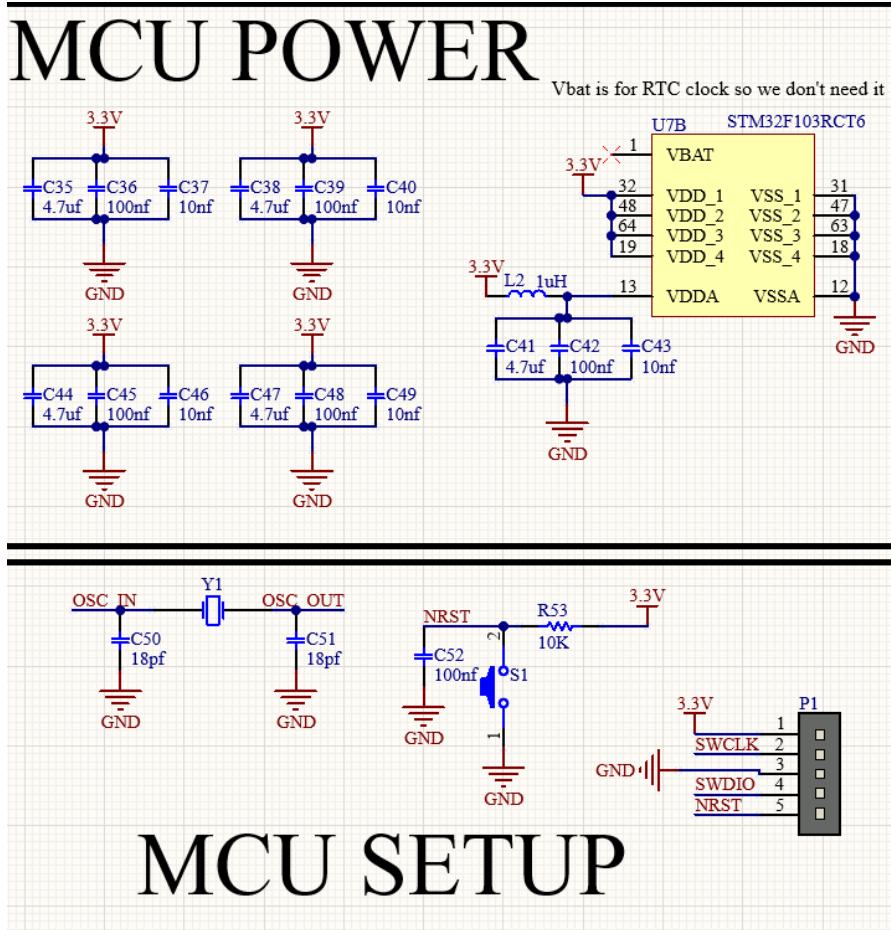
.6 مدار Push-pull برای تامین جریان DAC :



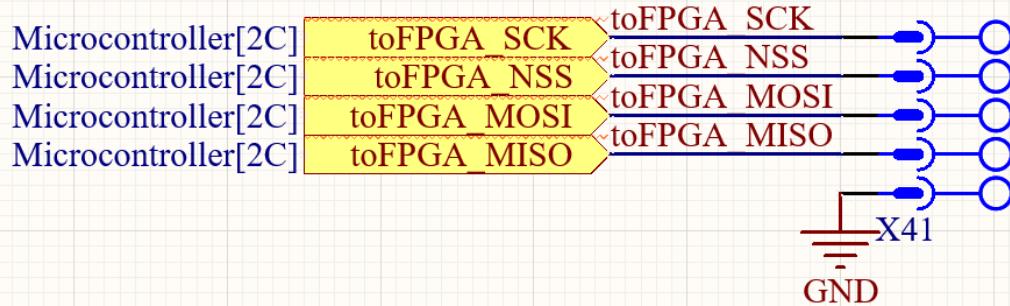
میکرو مناسب که مقادیر سیگنال ها را کنترل کند و اطلاعات را با تراشه اصلی رد و بدل کند:



8. راه انداز میکرو و تامین کلک آن با کریستال:



.9. یک کانکتور که اطلاعات را به تراشه اصلی طبق SPI منتقل کند:



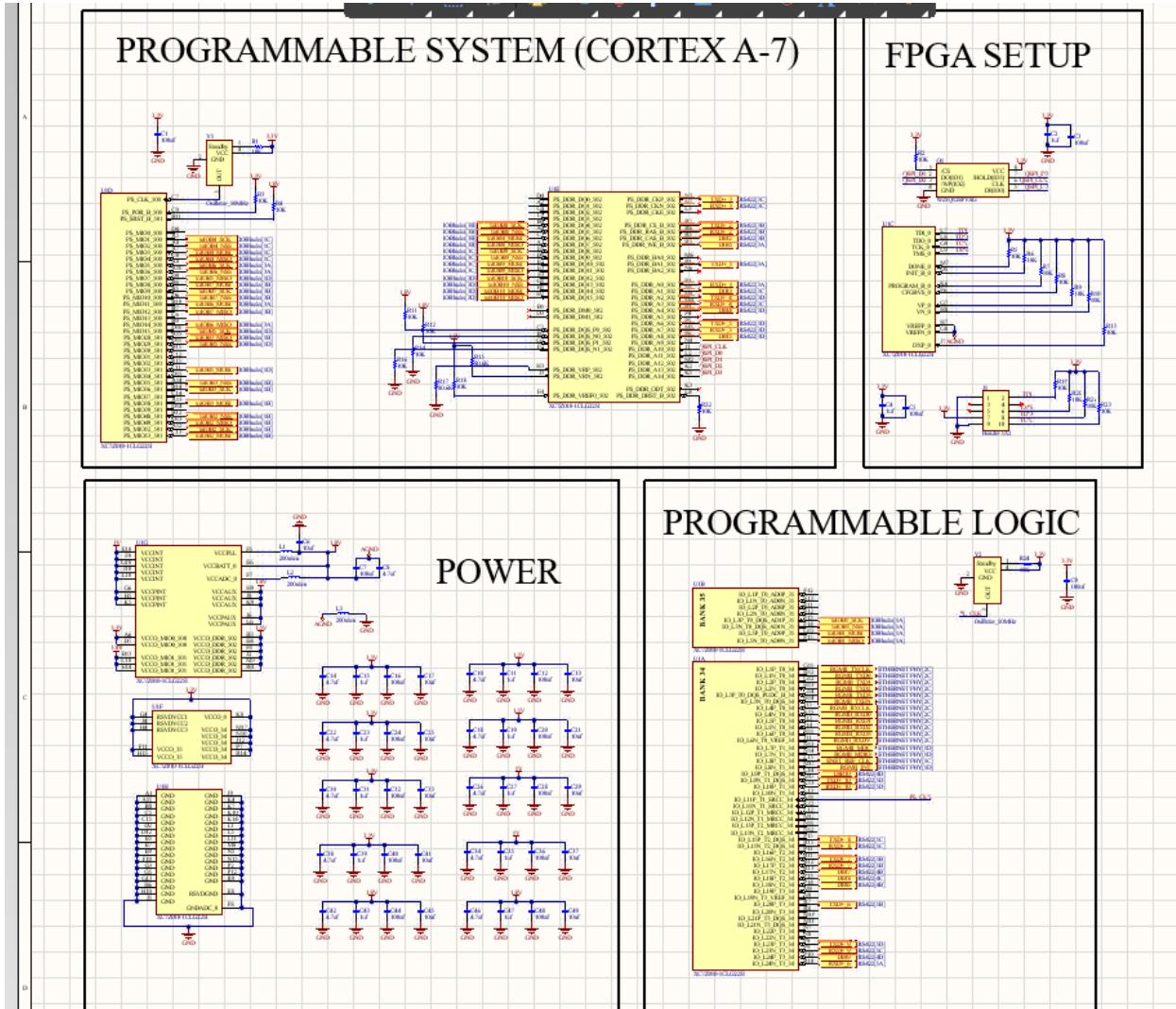
Microcontroller is connected to FPGA via SPI

## بخش دوم:

این بخش علارقم سنگین ترین بخش پروژه بودن توضیح خاصی ندارد و صرفا درد و رنج بود.

## بخش سوم تعريف پین ها:

در ابتدا TOP مازول را طبق شکل زیر(صرف اپورت های از پیش رزور نشده) به ترتیب در قسمت اپورت مازول وارد میکنیم و سپس از طریق elaborate design به تخصیص پایه های مناسب میپردازیم:



```
entity top is
  port (
    clk50MHz : in std_logic;
    toIOB4_SCK : out std_logic;
    toIOB4_NSS : out std_logic;
    toIOB4_MOSI : out std_logic;
    toIOB4_MISO : in std_logic;
    toIOB6_SCK : out std_logic;
    toIOB6_NSS : out std_logic;
```

```
toIOB5_MISO : in std_logic;
toIOB7_MOSI : out std_logic;
toIOB7_SCK : out std_logic;
toIOB7_NSS : out std_logic;
toIOB6_MOSI : out std_logic;
toIOB7_MISO : in std_logic;
toIOB6_MISO : in std_logic;
toIOB5_SCK : out std_logic;
toIOB3_MISO : in std_logic;
toIOB5_NSS : out std_logic;
toIOB5_MOSI : out std_logic;
toIOB3_NSS : out std_logic;
toIOB3_SCK : out std_logic;
toIOB3_MOSI : out std_logic;
toIOB2_NSS : out std_logic;
toIOB2_MISO : in std_logic;
toIOB2_SCK : out std_logic;
toIOB2_MOSI : out std_logic;
toIOB8_SCK : out std_logic;
toIOB8_NSS : out std_logic;
toIOB8_MOSI : out std_logic;
toIOB8_MISO : in std_logic;
toIOB9_SCK : out std_logic;
toIOB9_NSS : out std_logic;
toIOB9_MOSI : out std_logic;
toIOB9_MISO : in std_logic;
toIOB10_SCK : out std_logic;
toIOB10_NSS : out std_logic;
toIOB10_MOSI : out std_logic;
toIOB10_MISO : in std_logic;
TXD3 : out std_logic;
RXD3 : in std_logic;
TXD2 : out std_logic;
RXD2 : in std_logic;
DIR2 : out std_logic;
DIR1 : out std_logic;
TXD1 : out std_logic;
RXD1 : in std_logic;
DIR3 : out std_logic;
TXD4 : out std_logic;
RXD4 : in std_logic;
DIR4 : out std_logic;
TXD5 : out std_logic;
RXD5 : in std_logic;
DIR5 : out std_logic;
```

```
QSPI_CLK : out std_logic;
QSPI_D0 : inout std_logic;
QSPI_D1 : inout std_logic;
QSPI_D2 : inout std_logic;
QSPI_D3 : inout std_logic;
TDI : in std_logic;
TDO : out std_logic;
TCK : in std_logic;
TMS : in std_logic;
toIOB1_SCK : out std_logic;
toIOB1_NSS : out std_logic;
toIOB1_MOSI : out std_logic;
toIOB1_MISO : in std_logic;
RGMII_TXCLK : out std_logic;
RGMII_RXD0 : in std_logic;
RGMII_RXD1 : in std_logic;
RGMII_RXD2 : in std_logic;
RGMII_RXD3 : in std_logic;
RGMII_TXEN : out std_logic;
RGMII_RXCLK : in std_logic;
RGMII_RXD0 : in std_logic;
RGMII_RXD1 : in std_logic;
RGMII_RXD2 : in std_logic;
RGMII_RXD3 : in std_logic;
RGMII_RXDV : in std_logic;
RGMII_MDC : out std_logic;
RGMII_MDIO : inout std_logic;
ENET_REF_CLK : in std_logic;
RGMII_INT : in std_logic;
DIR10 : out std_logic;
TXD10 : out std_logic;
RXD10 : in std_logic;
PL_CLK : out std_logic;
TXD8 : out std_logic;
RXD8 : in std_logic;
TXD7 : out std_logic;
RXD7 : in std_logic;
DIR7 : out std_logic;
DIR8 : out std_logic;
DIR6 : out std_logic;
TXD6 : out std_logic;
TXD9 : out std_logic;
RXD9 : in std_logic;
DIR9 : out std_logic;
RXD6 : in std_logic
```

```
    );
end entity top;
```

پایه ها طبق تصویر و از بالا به پایین و از چپ به راست انتخاب شده اند

کانترین تولید شده:

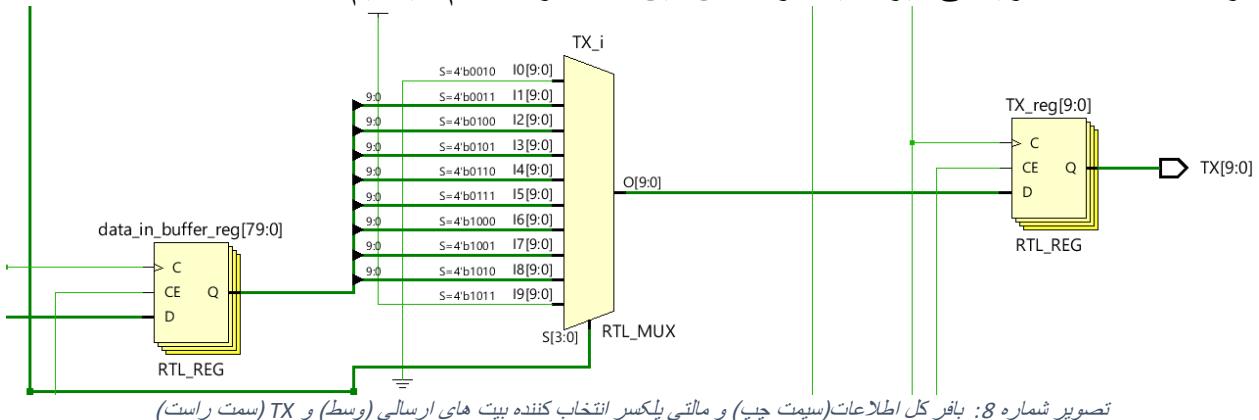
```
constrained.xdc
1  # clk50MHz
2  set_property PACKAGE_PIN C7 [get_ports clk50MHz]
3  set_property IOSTANDARD LVCMS33 [get_ports clk50MHz]
4  #-----
5
6  #toIOB4_SCK
7  set_property PACKAGE_PIN A5 [get_ports toIOB4_SCK]
8  set_property IOSTANDARD LVCMS33 [get_ports toIOB4_SCK]
9  #-----
10
11 #toIOB4_NSS
12 set_property PACKAGE_PIN A8 [get_ports toIOB4_NSS]
13 set_property IOSTANDARD LVCMS33 [get_ports toIOB4_NSS]
14 #-----
15
16 #toIOB4_MOSI
17 set_property PACKAGE_PIN A7 [get_ports toIOB4_MOSI]
18 set_property IOSTANDARD LVCMS33 [get_ports toIOB4_MOSI]
19 #-----
20
21 #toIOB4_MISO
22 set_property PACKAGE_PIN C8 [get_ports toIOB4_MISO]
23 set_property IOSTANDARD LVCMS33 [get_ports toIOB4_MISO]
24 #-----
25
26 #toIOB6_SCK
27 set_property PACKAGE_PIN A9 [get_ports toIOB6_SCK]
28 set_property IOSTANDARD LVCMS33 [get_ports toIOB6_SCK]
29 #-----
30
31 #toIOB6_NSS a10
32 set_property PACKAGE_PIN A10 [get_ports toIOB6_NSS]
33 set_property IOSTANDARD LVCMS33 [get_ports toIOB6_NSS]
34 #-----
```

## کدنویسی RS422

برای این طراحی با توجه به اینکه ساخت سیگنال تفاضلی در قسمت درایور انجام میشود، ما تنها با سیگنال های ارسالی (RX و TX) به همراه سیگنال های کنترل داده (CTS و RTS) سر و کار داریم، همچنین اطلاعات را در قالب پکت های 8 بیتی (به همراه یک بیت شروع و یک بیت پایان، یعنی در مجموع 10 بیت ارسال میکنیم)

کار نسبتا ساده است، سیگنال داده ورودی را (که توسط خود تراشه تامین میشود) را به 8 قسمت میشکنیم و در هر کلاک یکی از این قسمت ها را بر روی 10 پورت TX سوار میکنیم تا ارسال شوند.

در شروع داده های پایلوت 0 و در پایان ارسال نیز پایلوت 1 ارسال میشود.  
برای اطلاعات دریافتی نیز دقیقاً بر عکس این عمل را انجام میدهیم.



```
entity RS422 is
port(
    clk : in std_logic;
    data_in : in std_logic_vector(79 downto 0); -- 10 bytes of data
    data_out : out std_logic_vector(79 downto 0); -- 10 bytes of data
    RTS : out std_logic := '0';
    CTS : in std_logic := '0';
    TX : out std_logic_vector(9 downto 0);
    RX : in std_logic_vector(9 downto 0);
    send : in std_logic;
    ready : out std_logic
);
end RS422;
```

تجزیه ورودی و ترکیب RX به همراه ساخت مجموعه حالت ها

```

type state_type is (idle, request_to_send, sending_start_bit, send_0, send_1,
send_2, send_3, send_4, send_5, send_6, send_7, send_stop_bit);
signal state : state_type := idle;
begin
    data_in_buffer0 <= data_in_buffer(9 downto 0);
    data_in_buffer1 <= data_in_buffer(19 downto 10);
    data_in_buffer2 <= data_in_buffer(29 downto 20);
    data_in_buffer3 <= data_in_buffer(39 downto 30);
    data_in_buffer4 <= data_in_buffer(49 downto 40);
    data_in_buffer5 <= data_in_buffer(59 downto 50);
    data_in_buffer6 <= data_in_buffer(69 downto 60);
    data_in_buffer7 <= data_in_buffer(79 downto 70);

    data_out_buffer(9 downto 0) <= data_out_buffer0;
    data_out_buffer(19 downto 10) <= data_out_buffer1;
    data_out_buffer(29 downto 20) <= data_out_buffer2;
    data_out_buffer(39 downto 30) <= data_out_buffer3;
    data_out_buffer(49 downto 40) <= data_out_buffer4;
    data_out_buffer(59 downto 50) <= data_out_buffer5;
    data_out_buffer(69 downto 60) <= data_out_buffer6;
    data_out_buffer(79 downto 70) <= data_out_buffer7;

```

سويچ بين حالت ها و ارسال اطلاعات مناسب:

```

if rising_edge(clk) then
    case state is
        when idle =>
            if send = '1' then
                data_in_buffer <= data_in;
                ready <= '0';
                state <= request_to_send;
            end if;
        when request_to_send =>
            RTS <= '1';
            if CTS = '1' then
                RTS <= '0';
                state <= sending_start_bit;
            end if;
        when sending_start_bit =>
            TX <= "0000000000";
            state <= send_0;
        when send_0 =>
            TX <= data_in_buffer0;
            state <= send_1;
        when send_1 =>
            TX <= data_in_buffer1;
            state <= send_2;

```

-----حالات های دیگر صرف نظر شده ولی در سورس موجودند-----

```
when send_7 =>
    TX <= data_in_buffer7;
    state <= send_stop_bit;
when send_stop_bit =>
    TX <= "1111111111";
    ready <= '1';
    state <= idle;
end case;
end if;
```

دریافت اطلاعات نیز به همین صورت میباشد.

در نهایت برای تست کردن قطعات 1 نمونه از RS422 را میسازیم و ورودی یکی را به خروجی دیگری متصل میکنیم و انتظار داریم تا دیتای ورودی هر یک در خروجی دیگری ظاهر شود.

```
uut1: RS422 port map (
    clk => clk,
    data_in => data_in1,
    data_out => data_out1,
    RTS => RTS,
    CTS => CTS,
    TX => TX1,
    RX => RX1,
    send => send,
    ready => ready
);

uut2: RS422 port map (
    clk => clk,
    data_in => data_in2,
    data_out => data_out2,
    RTS => RTS,
    CTS => CTS,
    TX => TX2,
    RX => RX2,
    send => send,
    ready => ready
);

RX1 <= TX2;
RX2 <= TX1;
```

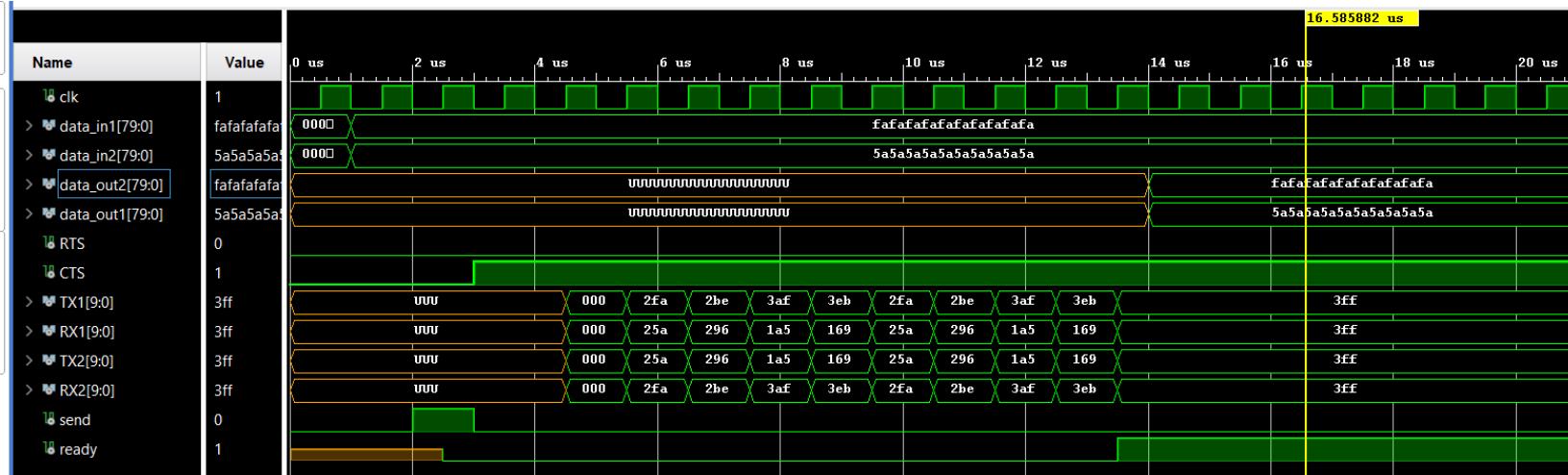
کلاک را برای مثال 1 مگ در نظر گرفتم ولی در واقعیت این مقدار بسیار پایین تر و در حدود چند صد کیلوبایت است.

```
clk_gen(clk, 1.0e6, 50.0); -- 100 MHz clock with 50% duty cycle is T = 1 us
```

```
stim_proc: process
begin
    -- hold reset state for 100 ns.
    wait for 1 us;

    -- insert stimulus here
    -- data_in1 <= x"FAFAFAFAFAFAFAFAFAFAFAFAFA";
    -- data_in2 <= x"5A5A5A5A5A5A5A5A5A5A5A5A";
    data_in1 <= x"000000000000000000000000";
    data_in2 <= x"FFFFFFFFFFFFFF";
    wait for 1 us;

    send <= '1';
    wait for 1 us;
    با یک شدن این سیگنال اجازه ارسال صادر میشود;
    CTS <= '1';
    send <= '0';
    wait;
end process;
```



میبینیم که طبق انتظار دیتای ورودی 1 در خروجی 2 و همچنین ورودی 2 در خروجی 1 ظاهر شد پس طرح به درستی کار میکند.

باند ریت در این مثال  $1\text{Mbit/s}$  است و لی برای بورد با توجه به محدود بودن نرخ ارسال و دریافت درایور انتخابی به نیم مگابیت بر ثانیه، باند ریت  $500\text{kbit/s}$  محدود میشود. البته این در حالت ایدهآل بوده چرا که عواملی مانند طول کابل نیز میتوانند بر این مقدار تاثیر گذاشته و آنرا باز هم بکاهند.

(در نظر داشتیم که ماژولیشنی بر روی اطلاعات توسط درایور صورت نمیپذیرد یعنی

$$(\text{bit}/\text{symbol} = 1)$$