

# Исследование топологии интегральных микросхем с помощью CV

Пятницкая О. Т.  
Разработчик центра DS&CV  
АО «ДиАйПи»

# ИНТЕГРАЛЬНЫЕ МИКРОСХЕМЫ



Электронные изделия, все элементы в котором соединены и выполняют общую функцию.

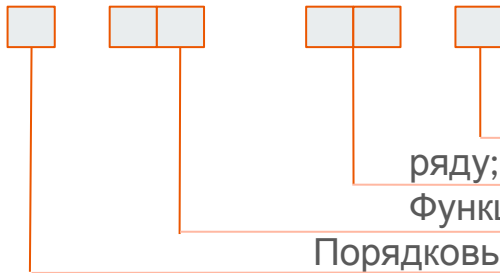
Микросхемы являются **основной элементной базой** вычислительной техники и играют **решающую роль** в развитии полупроводниковой и электронной промышленности.

Продажа и покупка IP-блоков современных микросхем составляет огромный рынок, на котором встречаются такие явления как пиратство, промышленный шпионаж и контроль за авторскими правами.

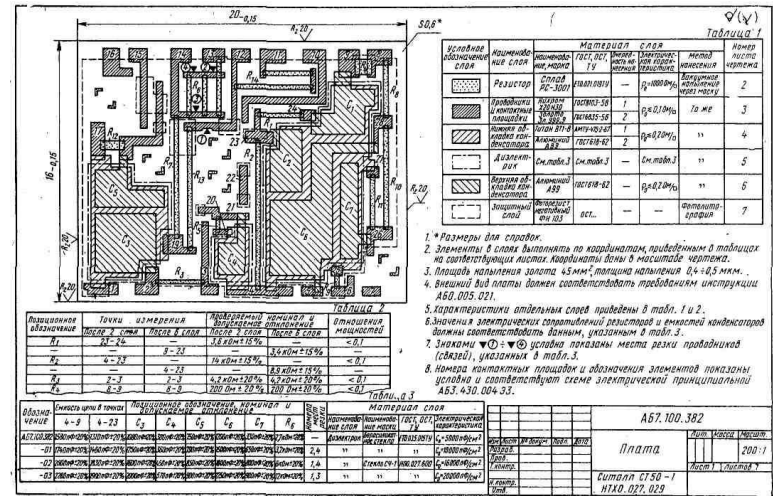
**Топология** - это расположение в этой микросхеме различных элементов и их связи между собой.

# Маркировки

СЕРИЯ ИНТЕГРАЛЬНОЙ МИКРОСХЕМЫ



Группа по конструктивно-технологическому исполнению.  
 Порядковый номер разработки серии;  
 Функциональное назначение;  
 Номер разработки в функциональном ряду;



1. Размеры для справок.
2. Элементы в слое выполняются по координатам, приведенным в таблицих на соответствующих листах. Координаты даны в масштабе чертежа.
3. Площадь нанесения золота 43 мм², платины 44 мм².
4. Внешний вид платы должен соответствовать требованиям инструкции А60.005.021.
5. Характеристики отдельных слоев приведены в табл. 1 и 2.
6. Значения электрических сопротивлений резисторов и емкости конденсаторов должны соответствовать значениям, указанным в табл. 2.
7. Знаками ▼ (▼) условия плавки места резки проводников (свай), указанных в табл. 3.
8. Номера контактных площадок и обозначения элементов показаны условно и соответствуют схеме электрической принципиальной АБ.430.004.03.

# Виды микросхем

## По конструктивно-технологическому исполнению:

- 1  
5 □ Полупроводниковые - все на едином кристалле;
- 1 □ Бескорпусные;
- 2  
4  
8 □ Гибридные(микросборки) - несколько компонентов в едином корпусе;
- 3 □ Прочие:
  - Пленочные;
  - Вакуумные;
  - Керамические.

## По степени интеграции:

**N** - число элементов и компонентов, входящих в ИС

**N < 100**

**100 < N < 1000**

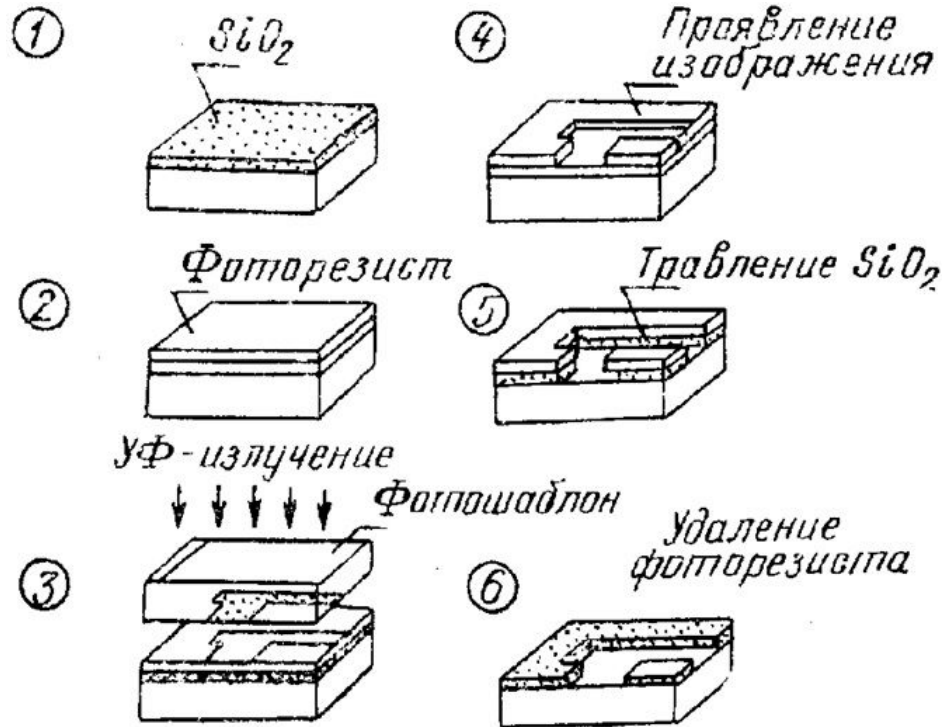
**N > 1000**

**N > 10000**

Схема.

- Малая Интегральная Схема;
- Средняя Интегральная Схема;
- Большая Интегральная Схема;
- Сверхбольшая Интегральная

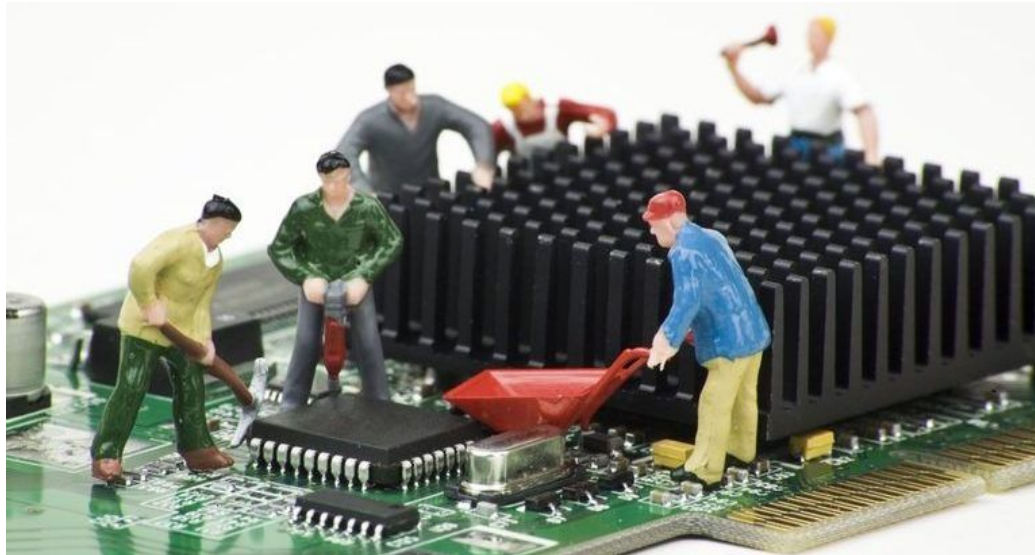
# Фотолитография



# Fabless

---

Модель организации бизнеса в электронной промышленности, при котором у компании-производителя нет собственных производственных мощностей.



# ОБРАТНАЯ РАЗРАБОТКА

---

Обратное проектирование  
Обратный инжиниринг Реверс-  
инжиниринг

## Reverse engineering

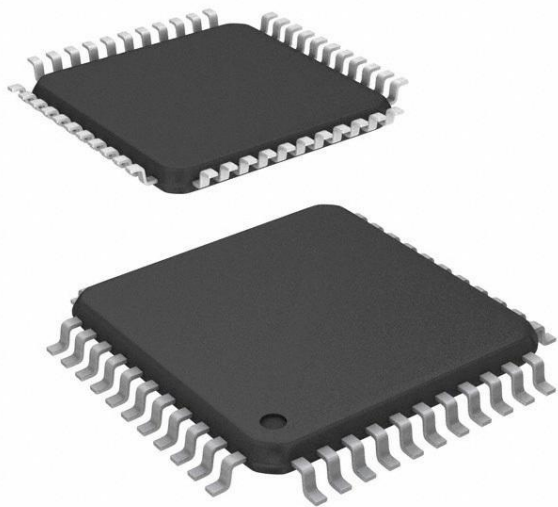
Исследование устройства или программы.

### Цели:

- Понимание работы устройства;
- Воспроизведение и изменение устройства;
- Поиск программных закладок;
- Обнаружение недокументированных возможностей;
- Промышленный шпионаж;
- Патентные и лицензионные суды.

# Недекларированные возможности

Аппаратные трояны - едва ли не самые большие угрозы для любого электронного



Могут быть вставлены в неиспользуемые места чипа, изменением размеров ячеек или их прореживанием.

Полномасштабный **реверс-инжиниринг** для реконструкции списка всех цепей и взаимосвязей - наиболее эффективный метод, однако он:

- Разрушителен для образцов;
- Требуется дорогостоящего оборудования;
- Затратен по времени.

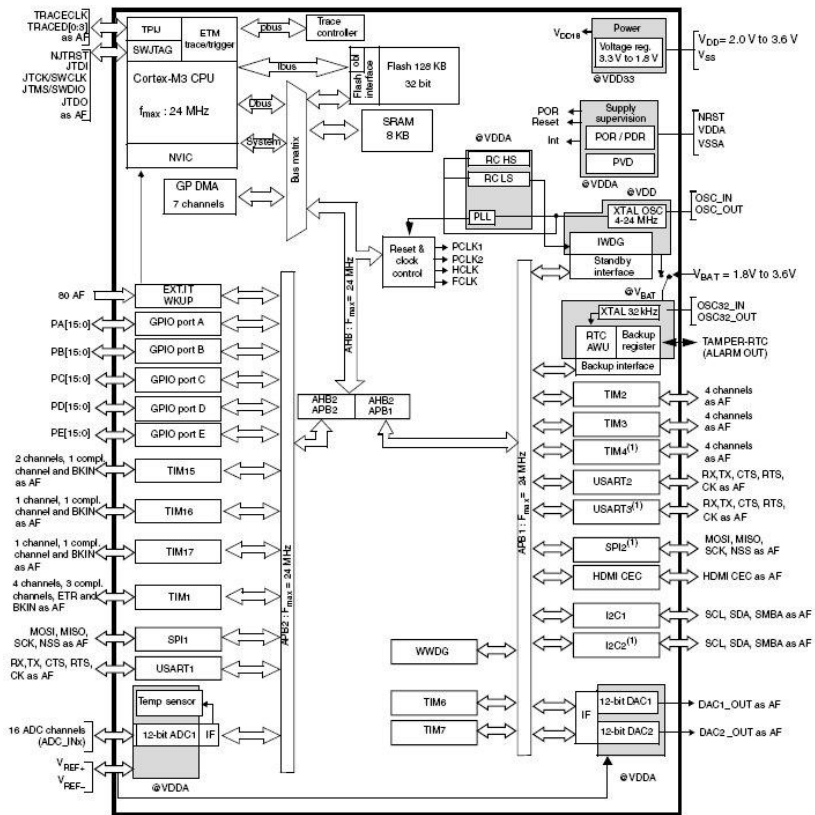


# STM32F100RB

Микроконтроллер широкого спектра назначения.

## Ключевые особенности:

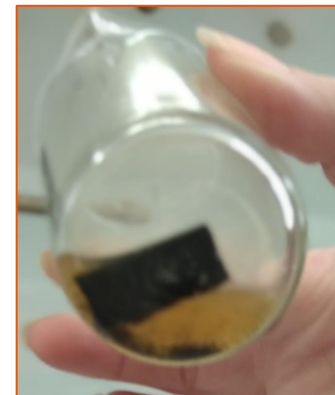
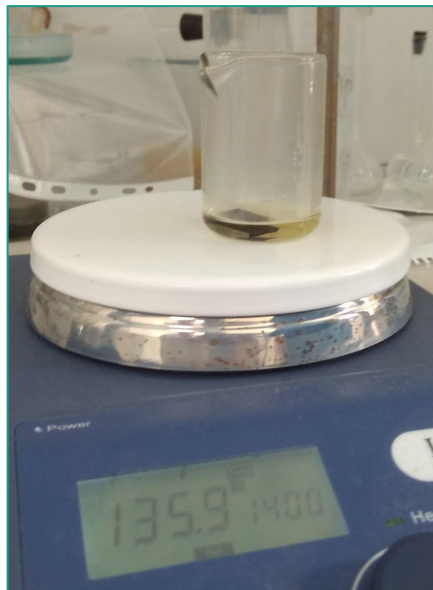
- Ядро: ARM® 32-битный процессор Cortex® -M3
- Максимальная частота 24 МГц, производительность 1,25 DMIPS / МГц (Dhrystone 2.1)
- Умножение и аппаратное деление за один цикл
- Flash память в 128 Кбайт
- RAM память в 8 Кбайт
- Работа при температуре от -40 до 85 градусов по Цельсию



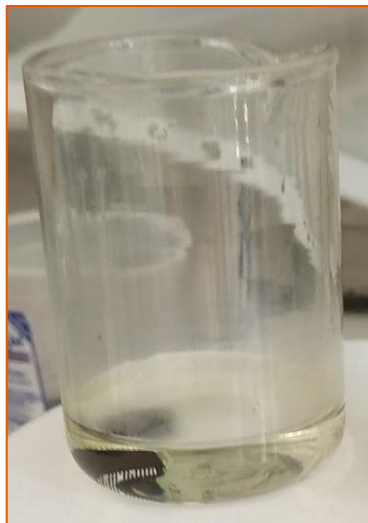
Блок -диаграмма микроконтроллера - Value Line block diagram

# Кислотный реверс

Первым делом необходимо “обнажить” кристалл - отделить от защитного корпуса.



Растворяем его в азотной кислоте при нагревании.



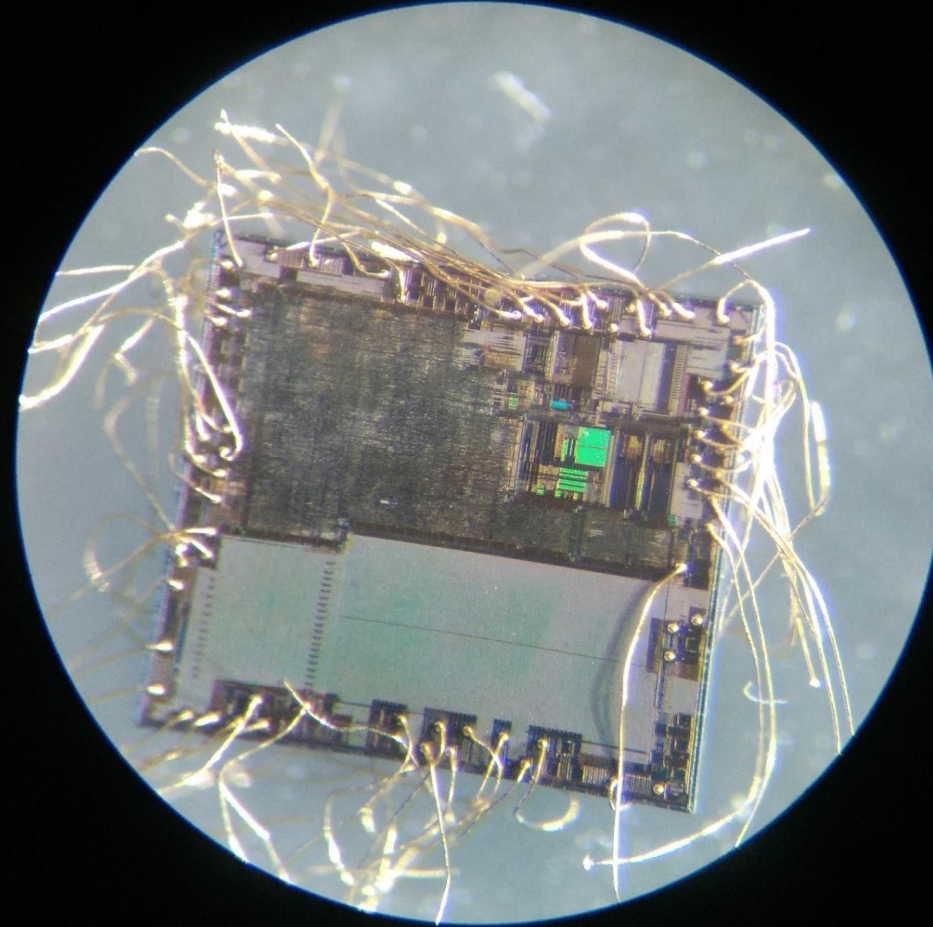
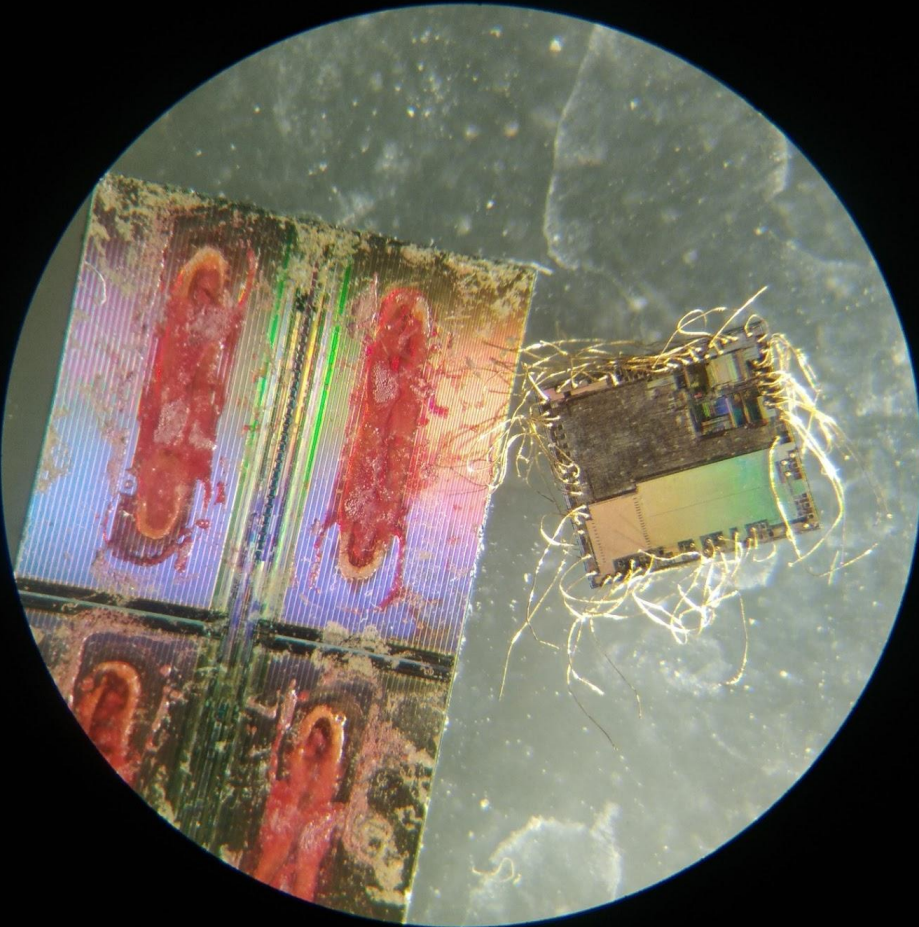
В конце промываем водой.



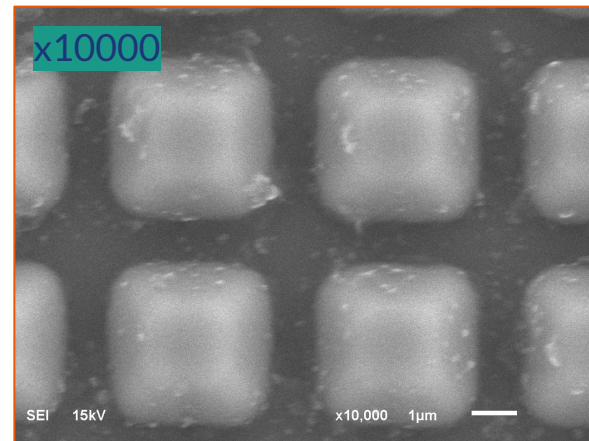
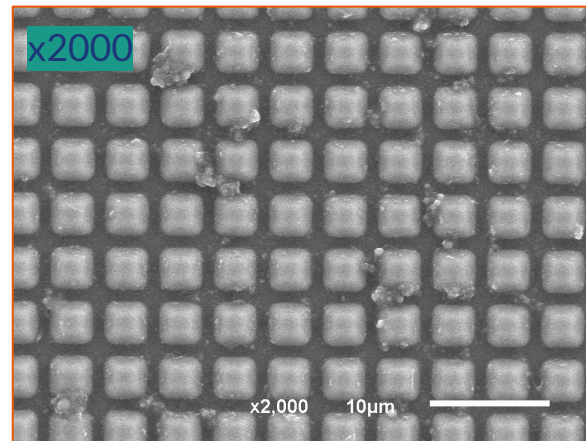
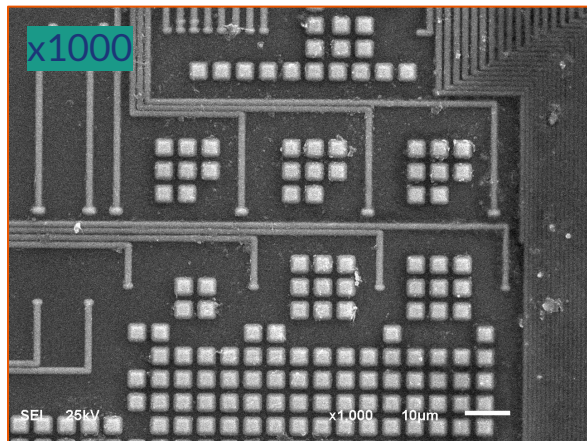
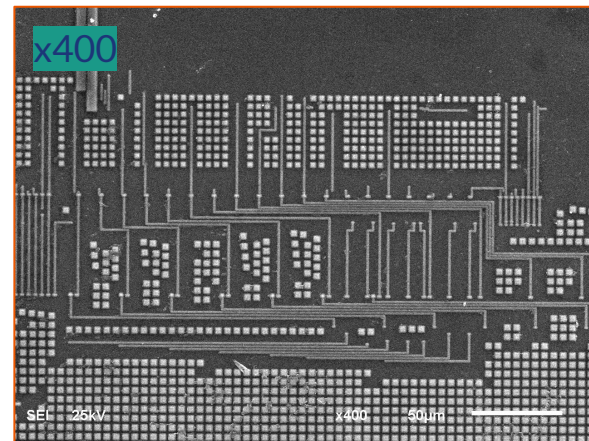
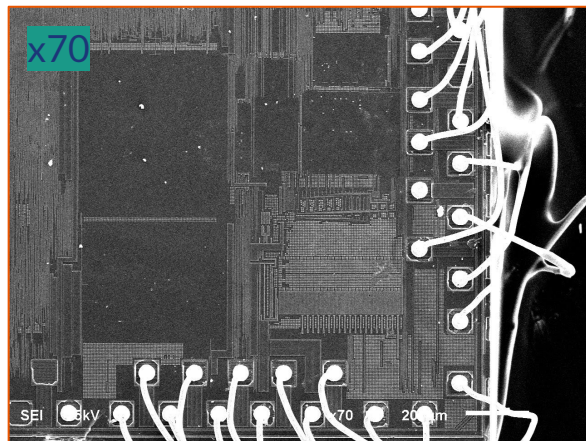
После того, как микроконтроллер  
“оголен”, можно приступить к  
фотографированию



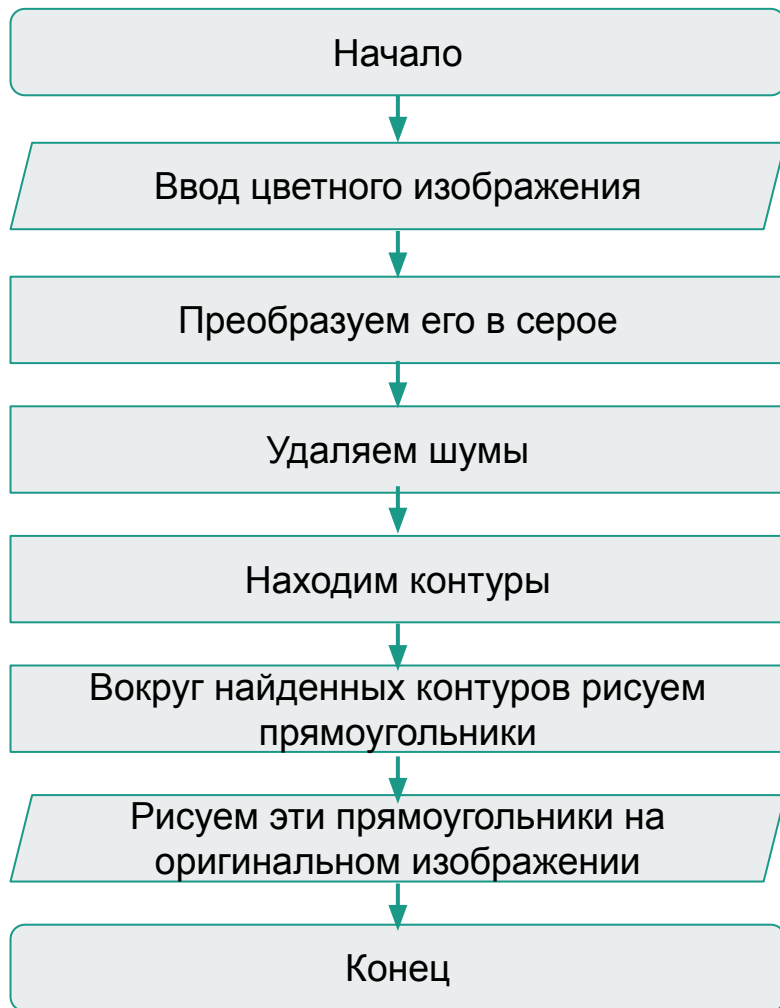
# Оптическая микроскопия (микроскоп МБС-10)



# Растровая электронная микроскопия (JEOL JSM-6510)



# Блок-схема предстоящей обработки полученных изображений.



```
In [309]: import cv2
import matplotlib.pyplot as plt
import numpy as np
import imutils
from random import randrange

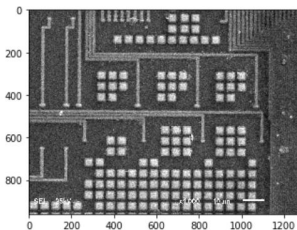
%matplotlib inline
```

```
In [310]: img = cv2.imread('data/point-1_0018.bmp')
img.shape
```

```
Out[310]: (960, 1280, 3)
```

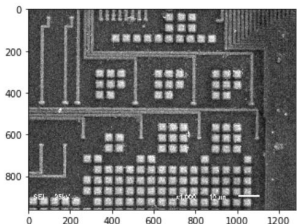
```
In [311]: plt.imshow(img)
```

```
Out[311]: <matplotlib.image.AxesImage at 0x31b27e30>
```



```
In [312]: #преобразует в серое
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
#отрисовываем
plt.imshow(gray, cmap='gray')
```

```
Out[312]: <matplotlib.image.AxesImage at 0x31b65b10>
```



```
In [313]: #преобразует в серое
ret, thresh = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)
```

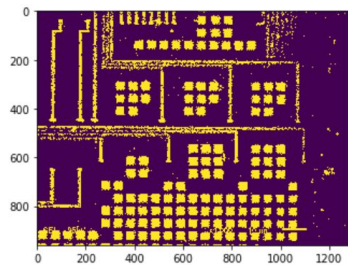
```
In [314]: plt.imshow(thresh)
```

```
Out[314]: <matplotlib.image.AxesImage at 0x32414c10>
```

Исходное изображение

Преобразуем в серое



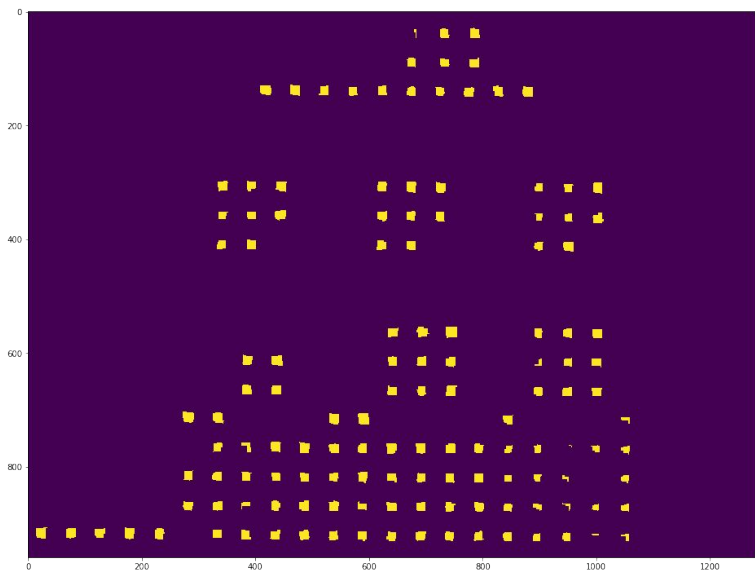


“серое” изображение

```
In [315]: #следующие две строчки удаляют шум и разные полосы  
kernel = np.ones((15,15),np.uint8)  
erosion = cv2.erode(thresh,kernel,iterations = 1)
```

```
In [316]: plt.figure(figsize=(16,16))  
plt.imshow(erosion)
```

```
Out[316]: <matplotlib.image.AxesImage at 0x32453870>
```



Удаляем “шум”

```
In [317]: # gr = cv2.cvtColor(erosion, cv2.COLOR_BGR2GRAY)
# ret,thr = cv2.threshold(gr,127,255,cv2.THRESH_BINARY)

In [318]: plt.figure(figsize=(16,16))

Out[318]: <Figure size 1152x1152 with 0 Axes>
<Figure size 1152x1152 with 0 Axes>

In [319]: #поиск контуров
cnts = cv2.findContours(erosion.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_TC89_KCOS)[0]

In [320]: #делаем из контуров прямоугольники
boxes = [cv2.boundingRect(cnt) for cnt in cnts]

In [321]: #рисуем прямоугольники
for box in boxes:
    x,y,w,h = box
    cv2.rectangle(img, (x-10, y-10), (x + w + 10, y + h + 10), (randrange(255), randrange(255), randrange(255)), 4)

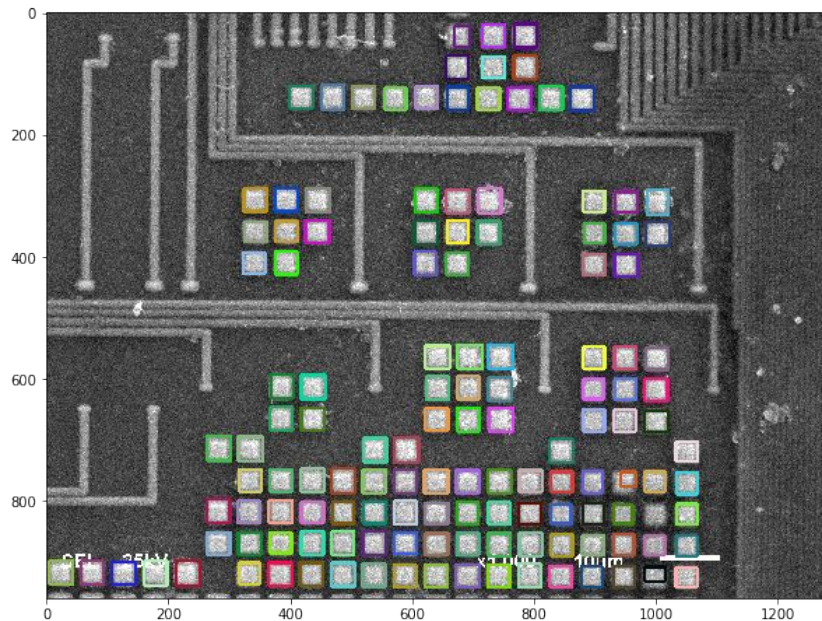
In [322]: plt.figure(figsize=(10,10))
plt.imshow(img)

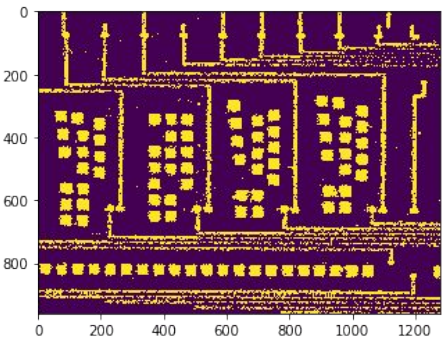
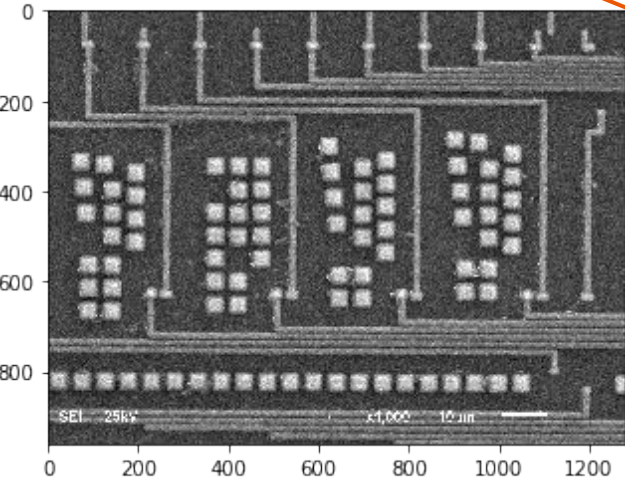
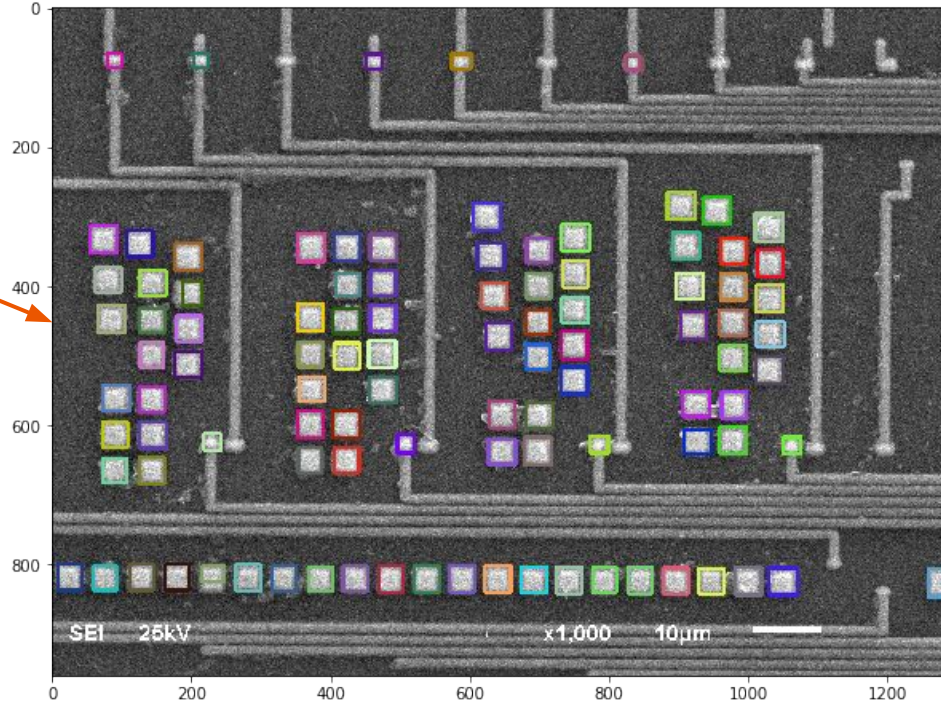
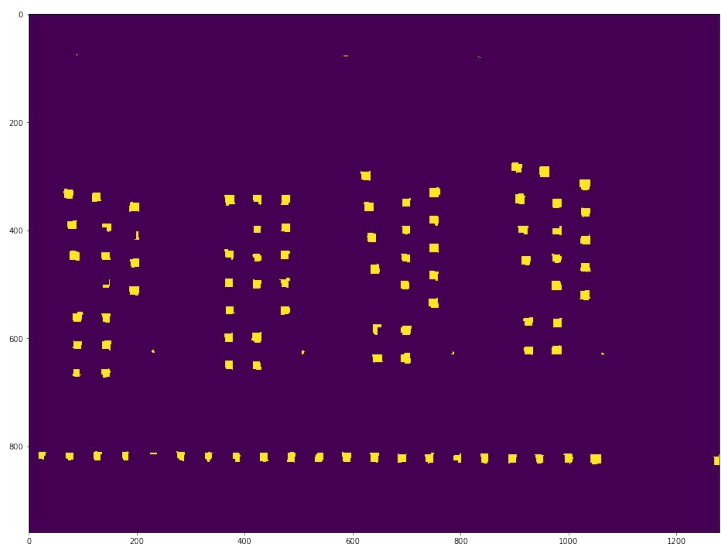
Out[322]: <matplotlib.image.AxesImage at 0x3248dfb0>
```

Ищем контуры

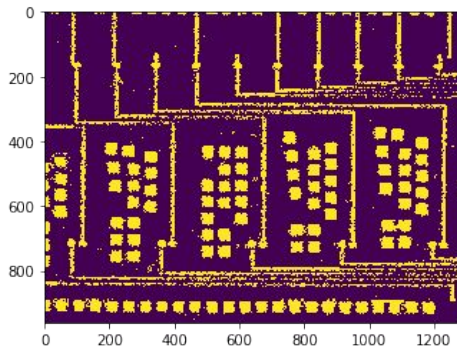
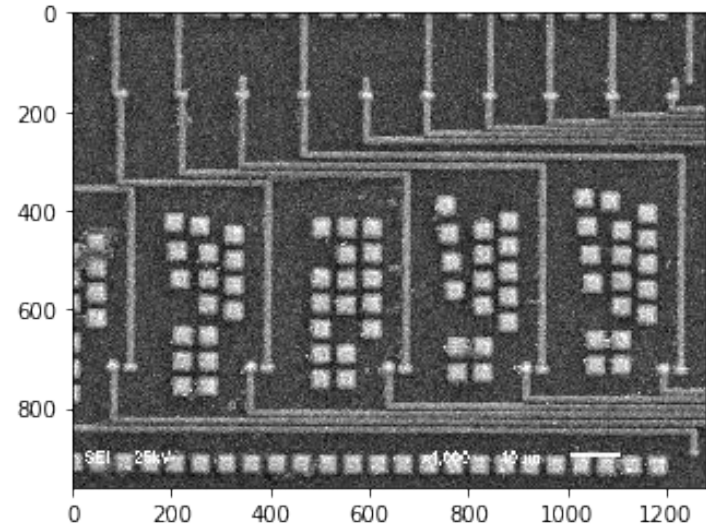
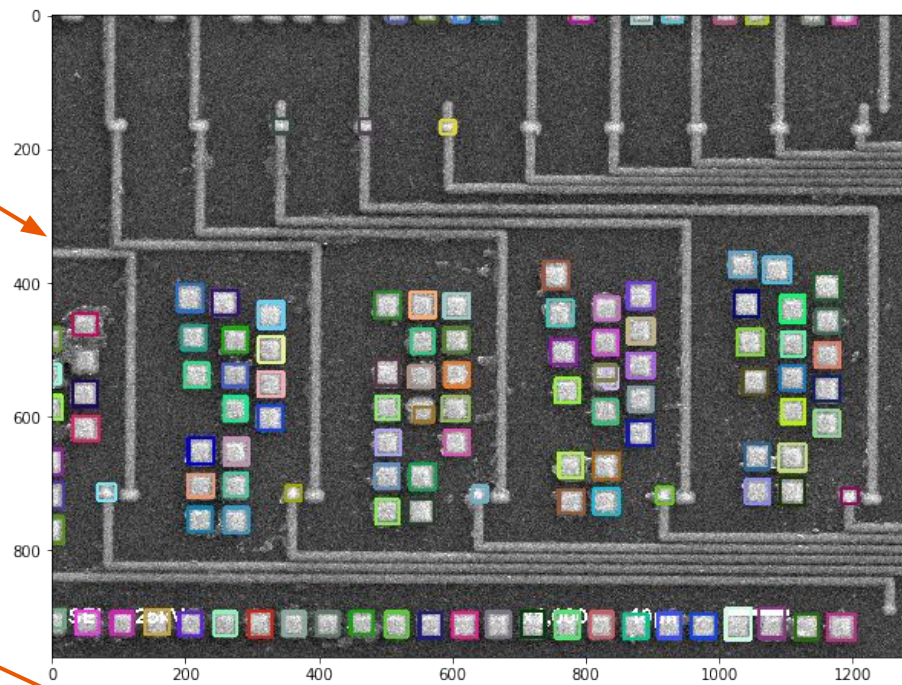
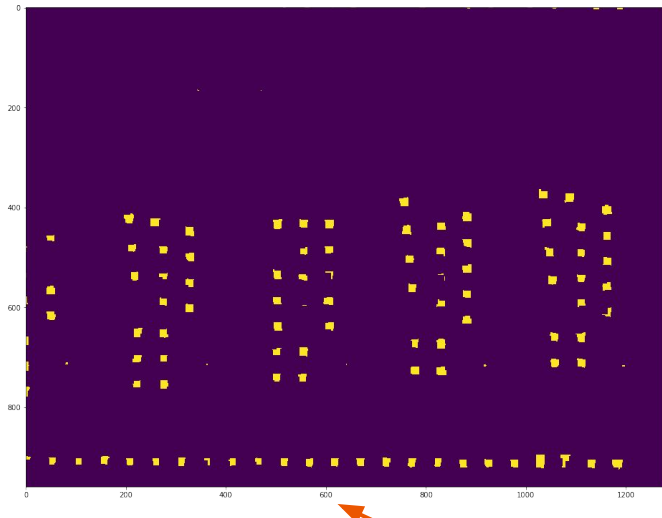
Обводим их  
прямоугольниками

Накладываем их на  
оригинальное изображение

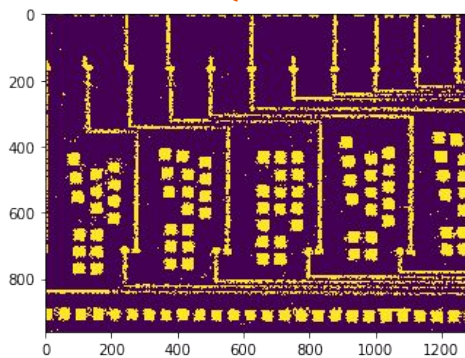
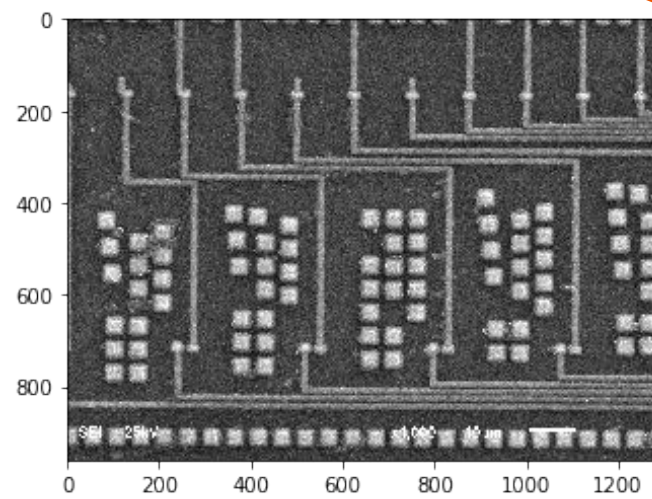
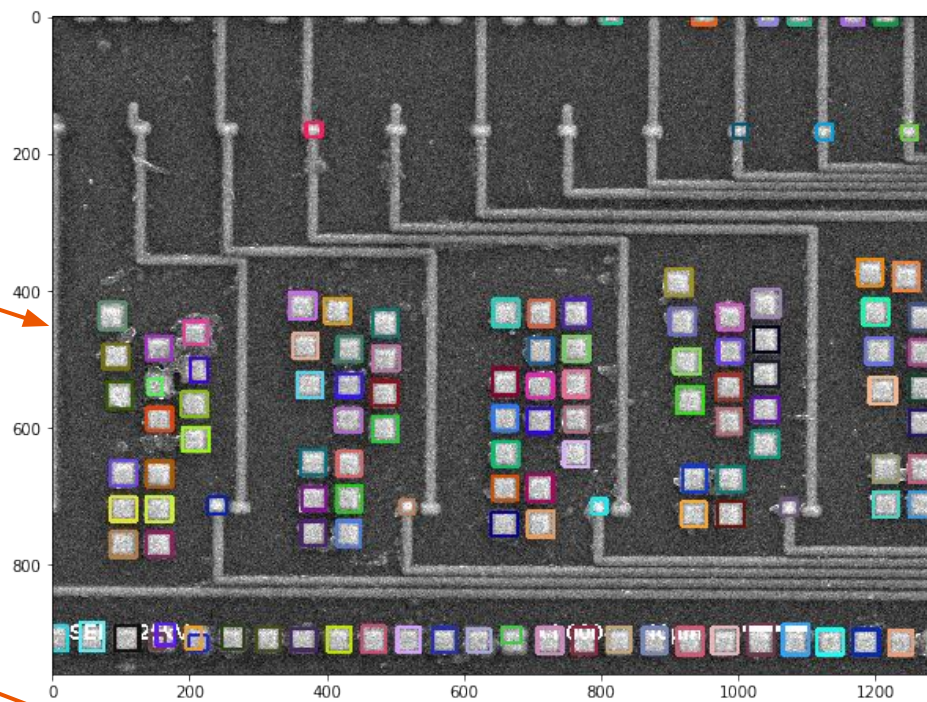
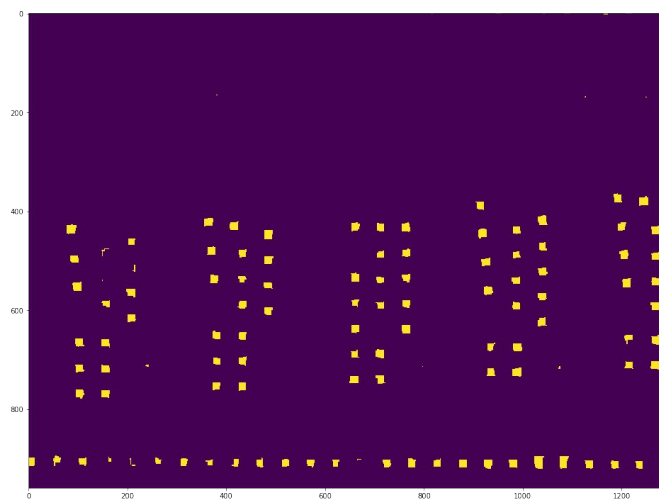




Тестируем на втором изображении того же масштаба



Тестируем на третьем  
изображении того же  
масштаба



Тестируем на четвертом  
изображении того же  
масштаба

# Выводы

---

Реализованное программное обеспечение позволяет провести автоматический поиск элементов по взаимному расположению кремниевых/металлических структур на микроизображениях. Используемые изображения были получены при помощи растровой электронной микроскопии.

Реализованный программный комплекс чувствителен к масштабу изображения, частичной расфокусировке, повороту в пределах  $0-123^\circ$ , может корректно использоваться для монохромных контрастных и/или цветных изображениях. Поэтому, при исследовании используемых изображений некоторые элементы были обнаружены с погрешностями.

Возможности созданного программного комплекса могут быть расширены разработкой методов автоматического сопоставления изображений одного участка поверхности, полученных различными методами или же с эталоном

---

**Спасибо за внимание!**