

hardware.io

Hardware Security Conference and Training

Hacking NAND Memory Pinout using Logic Analyzer

Alexander "Sasha" Sheremetov, Rusolut, Poland

WHO NEEDS THAT?

EVERYONE WHO HAS EVER WORKED WITH:

- PORTABLE STORAGE (CARDS, FLASHDRIVES, ETC)
 - VEHICLE INFOTAINMENT SYSTEMS, ECUs, etc
 - SMARTPHONES
 - TABLETS
 - DRONES
 - ROUTERS
 - WEARABLES/SMARTWATCHES
 - LAPTOPS
 - SOLID STATE DRIVES
 - VOICE RECORDERS
 - MULTIMEDIA PLAYERS
 - SMART TV
 - INTERNET OF THINGS
- ...AND MUCH MORE...

DIGITAL FORENSICS

SECURITY
RESEARCHERS

DATA RECOVERY

VENDORS & INDUSTRY

TRANSPORTATION
SAFETY BOARDS

WHY? BECAUSE WE WANT TO READ NAND MEMORY IN CASE IF:

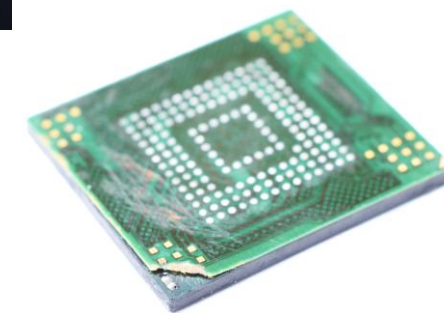
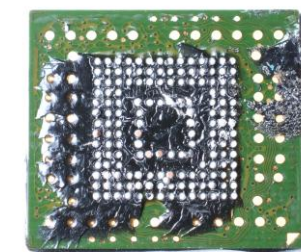
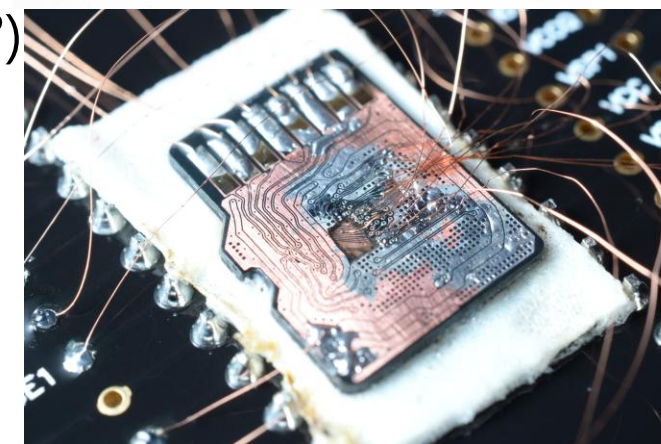
- Flash memory devices unrecognized by PC
- Secured/Locked flash media such as SD, eMMC, etc
- Damaged flash devices (fire, water, cracks)
- Recovery of data artefacts from non-erased blocks

<https://ieeexplore.ieee.org/document/9777707>

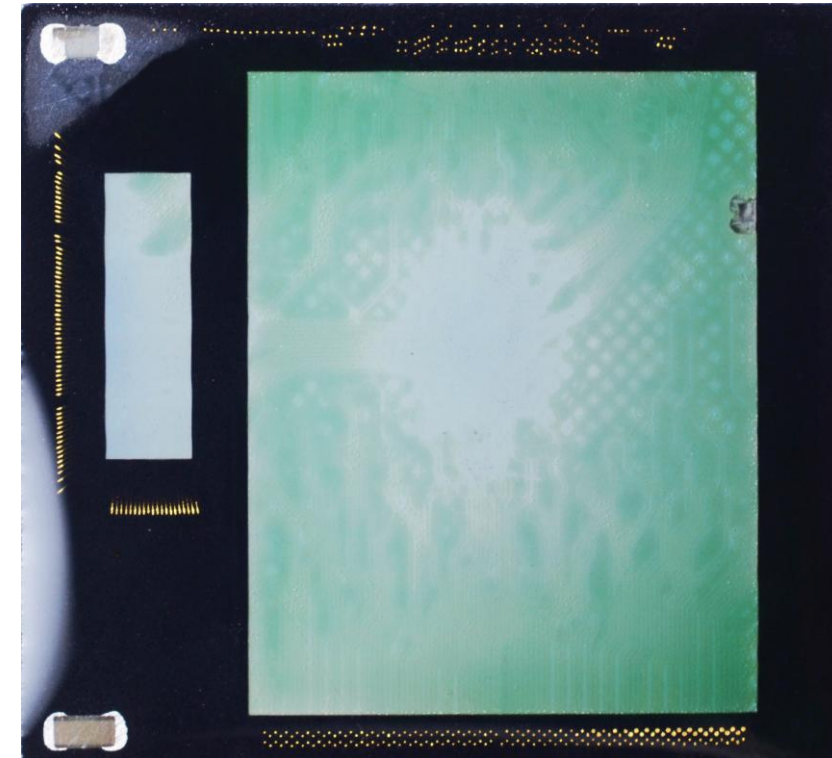
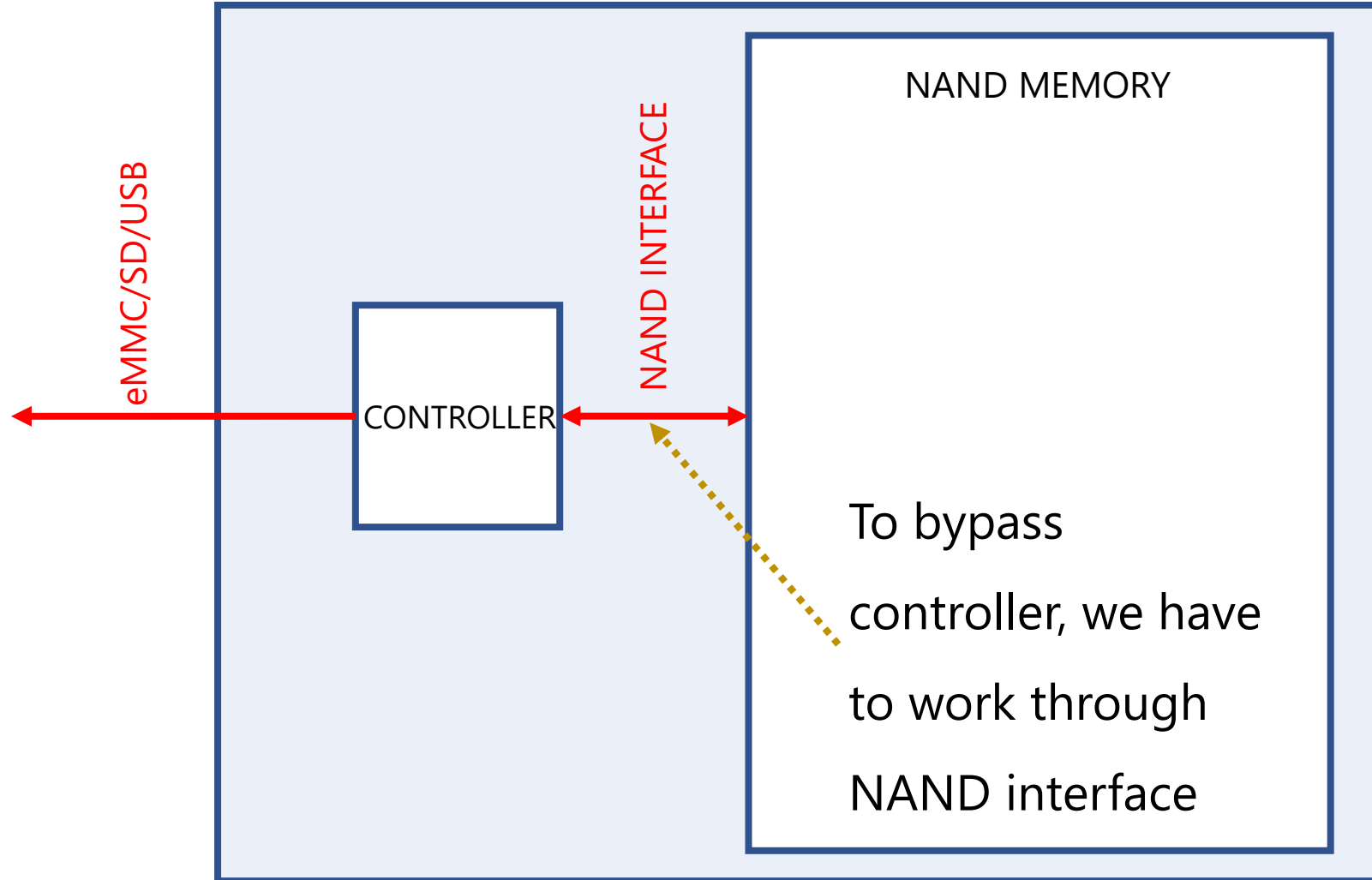
- Access every byte of the memory (hacking RPMB...?)

<https://www.sciencedirect.com/science/article/pii/S2666281723002019>

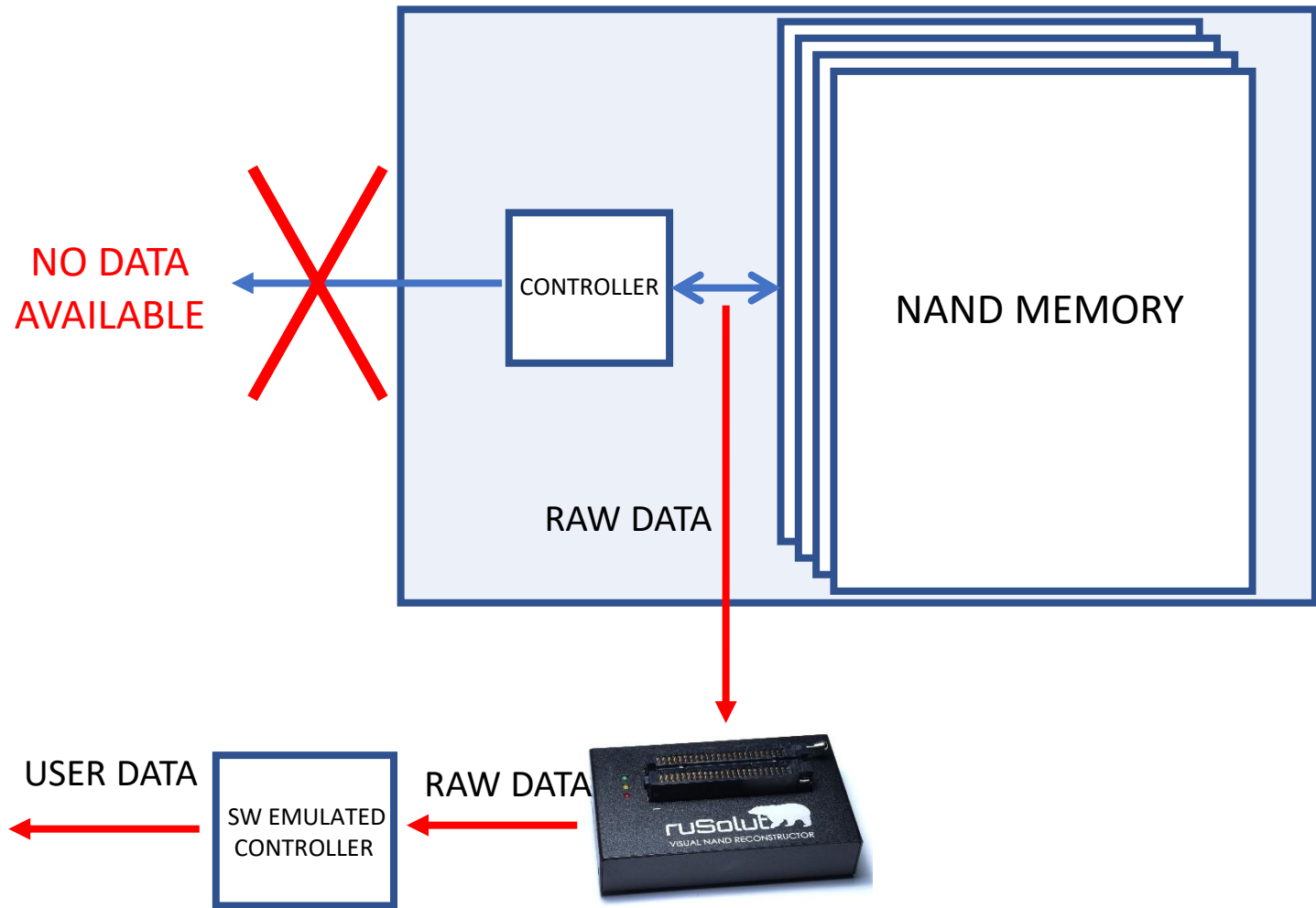
- Reverse engineering of FW/security/protocol layer



EMBEDDED NAND STRUCTURE



CHIP-OFF DATA ACQUISITION METHOD



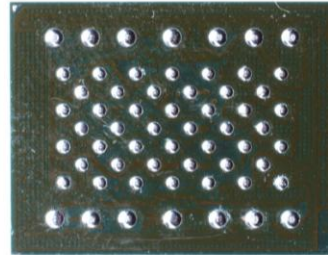
1. NAND memory connection
2. Physical image extraction
3. Controller emulation

KNOWN PACKAGES OF NAND

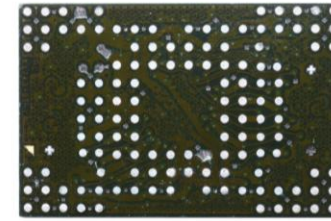
TSOP48



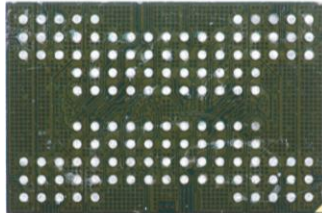
LGA52



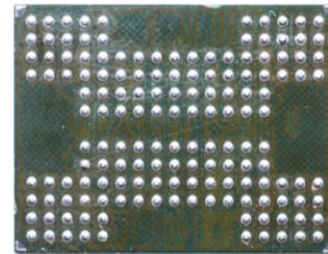
BGA154



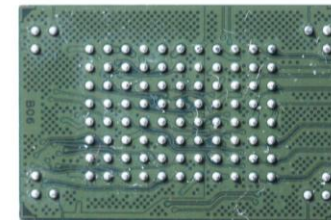
BGA132



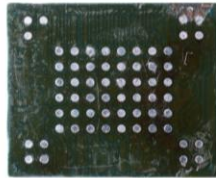
BGA152



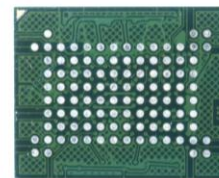
BGA100



BGA63



BGA107



BGA137



EASY TO READ



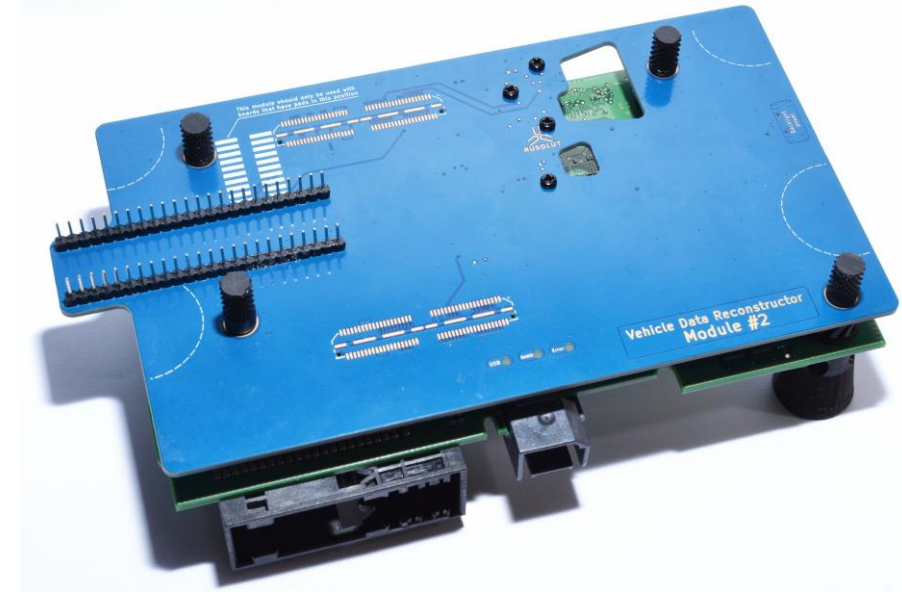
BGA152



BGA100



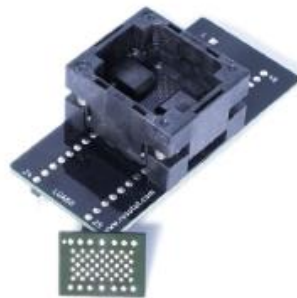
BGA132



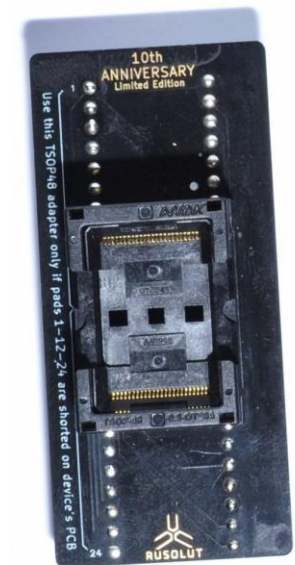
BGA107



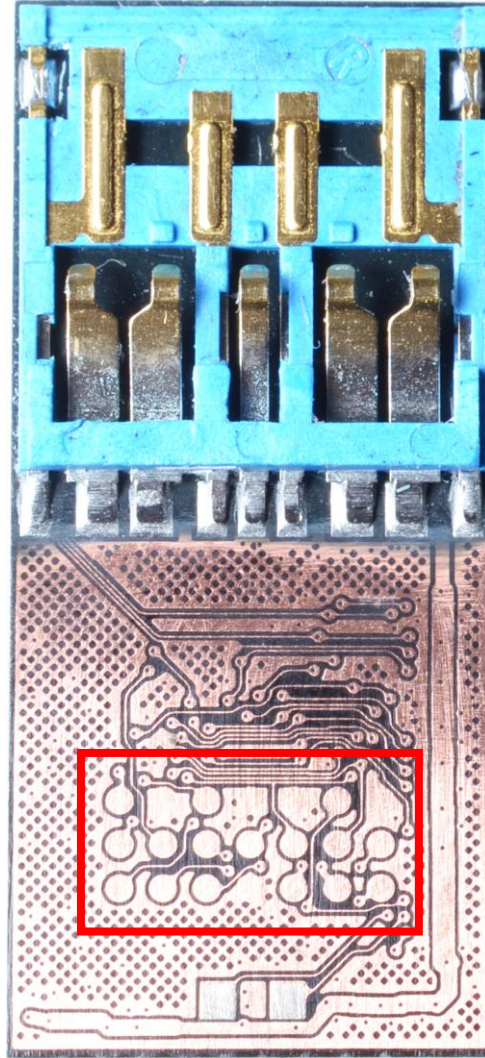
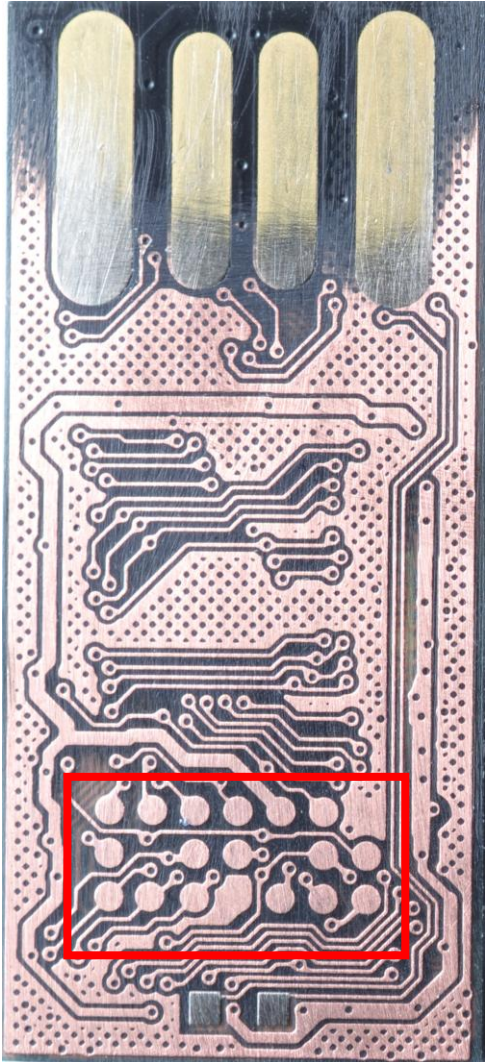
BGA137



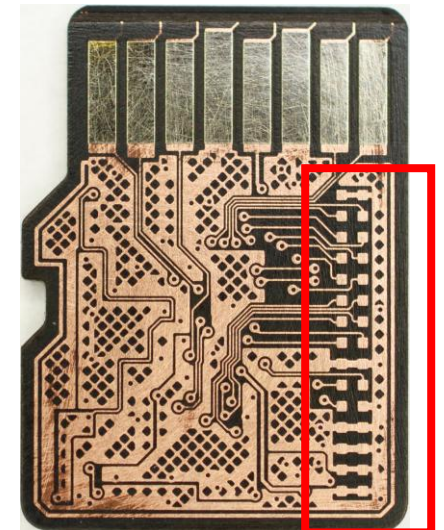
LGA60 (FOR TH58TFT1DFKLAVH AND ALIKE)



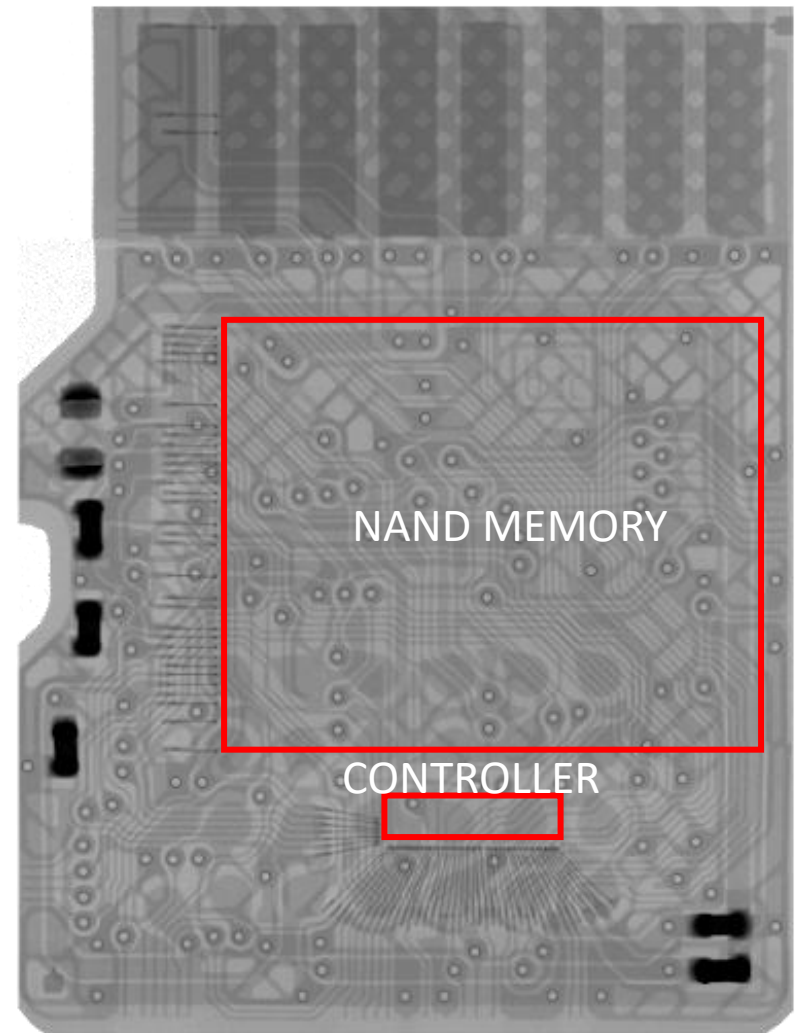
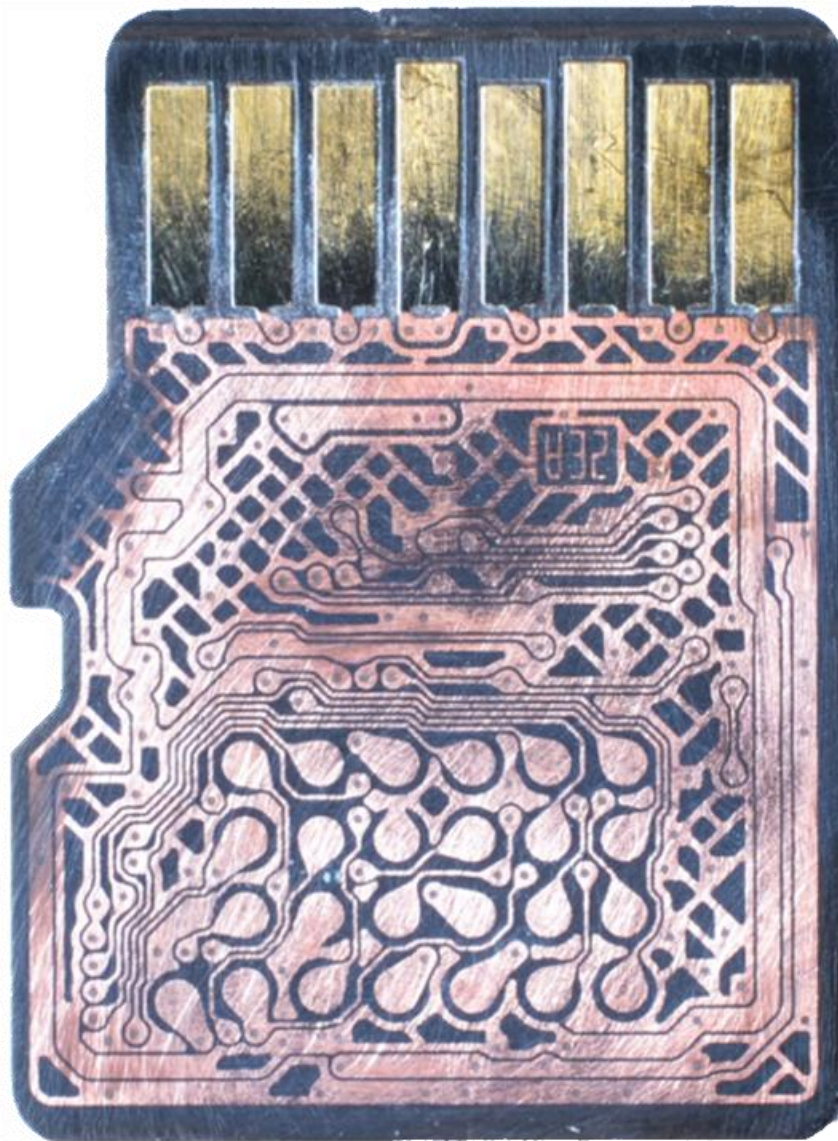
Technological pads of monolithic chips



Pinout - ???



MicroSD inside



MicroSD inside

Monolith pinouts X Workspace

Monolithic device type

USB Flash disk
MicroSD card
SD card
Others

Technological pads

No pads Technological pads

Device size

Short Normal

* CTRL + Click to disable a filter
Reset all filters

0125 0126 0127 0128

0133

Print

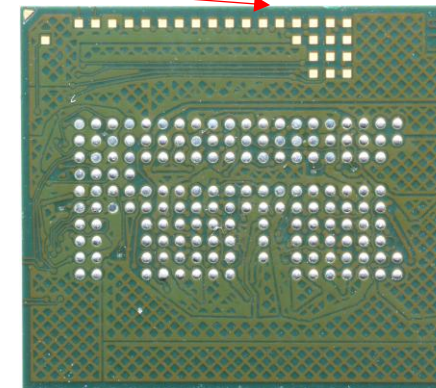
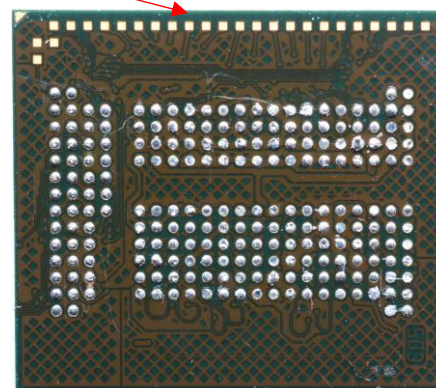
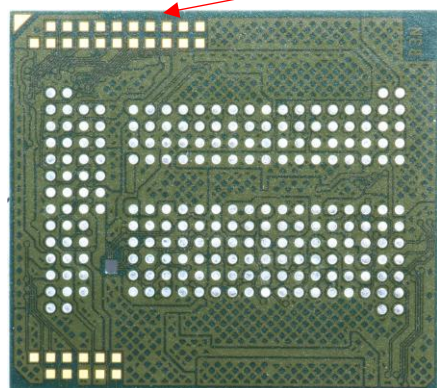
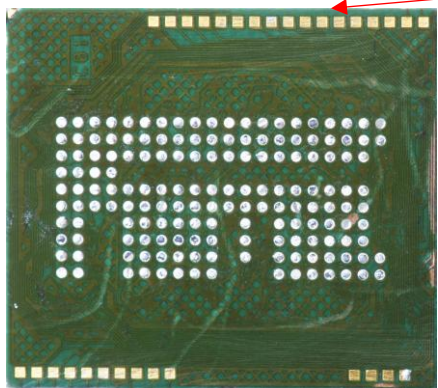
0132

0136

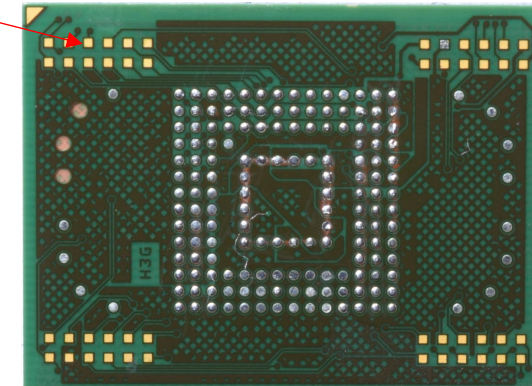
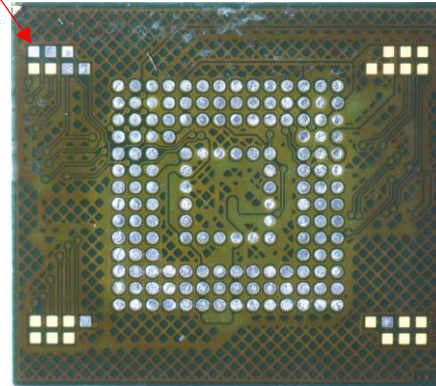
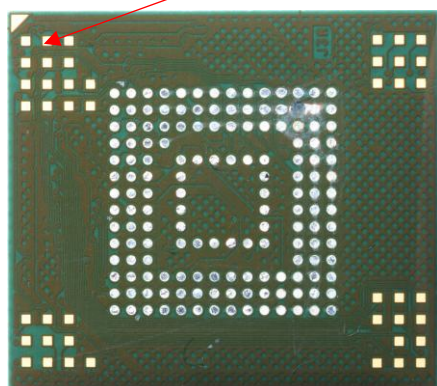
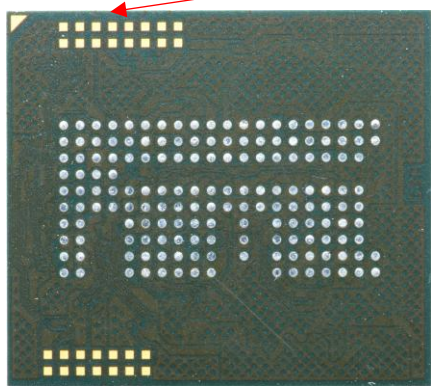
RE Vcc R/B0-1 CE0 IO7 IO6 IO5 CE1 IO3 Vss CLE ALE IO4 IO1 WE IO2 IO0

FGC3
FRT-FGC3-1581
1638-AR

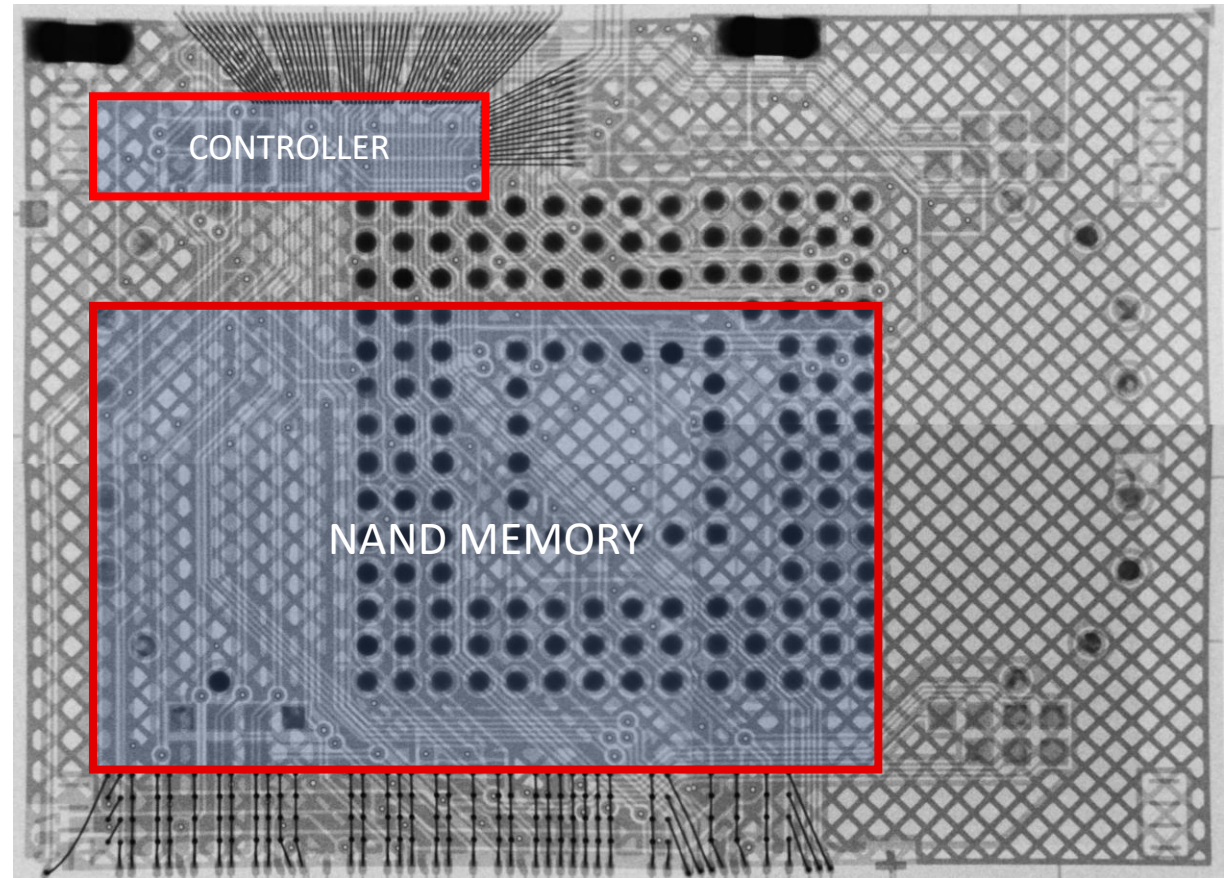
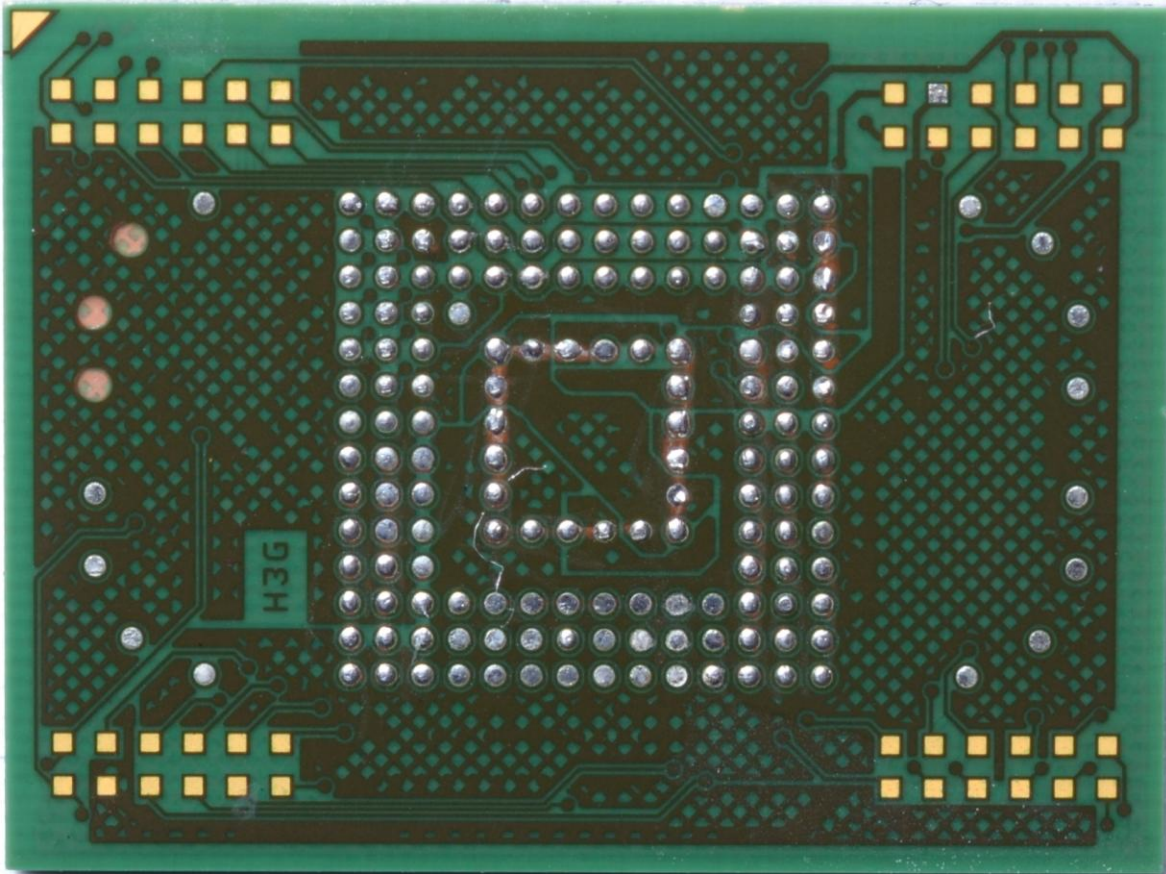
Technological pads of eMMC



Pinout - ???



eMMC inside



NAND INTERFACE – TSOP48

IO0...IO7 – Data bus

Vcc/VccQ – power 3,3 or 1,8 V

Vss/VssQ - Ground

CLE – Command latch enable

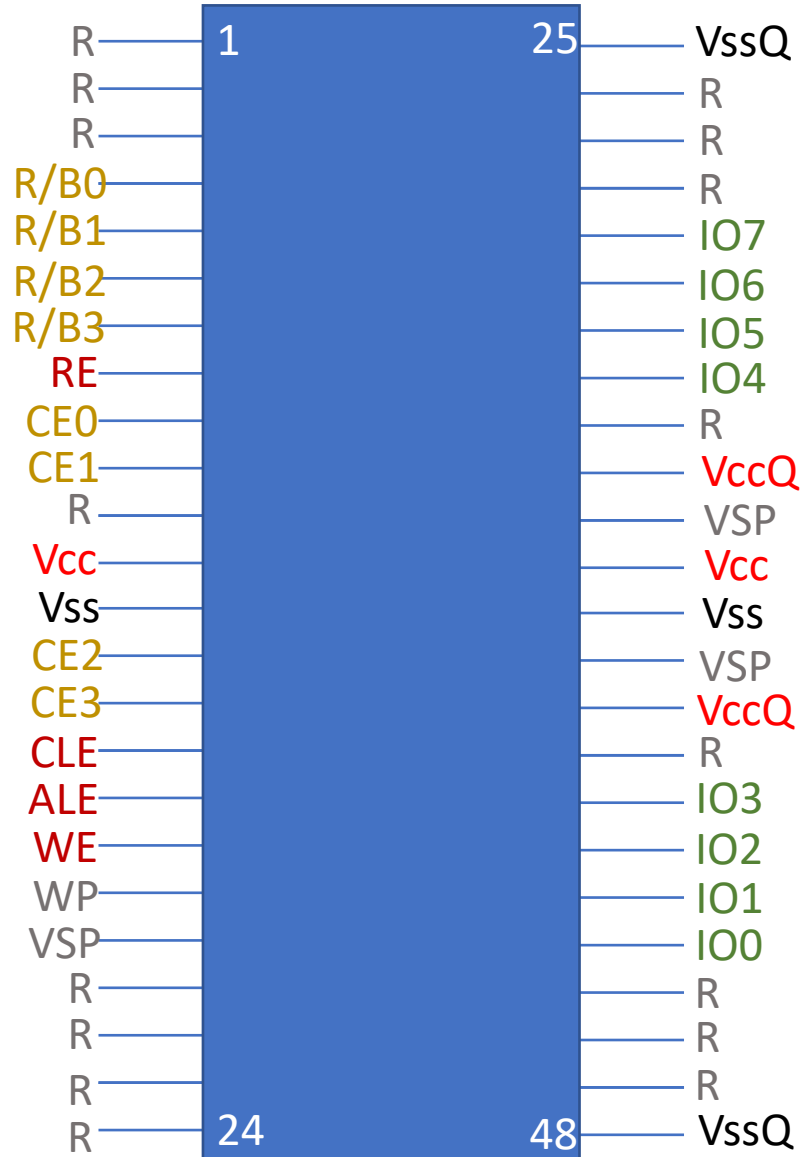
ALE – Address latch enable

RE – Read enable

WE – Write enable

R/B0...R/B3 – Ready/Busy status

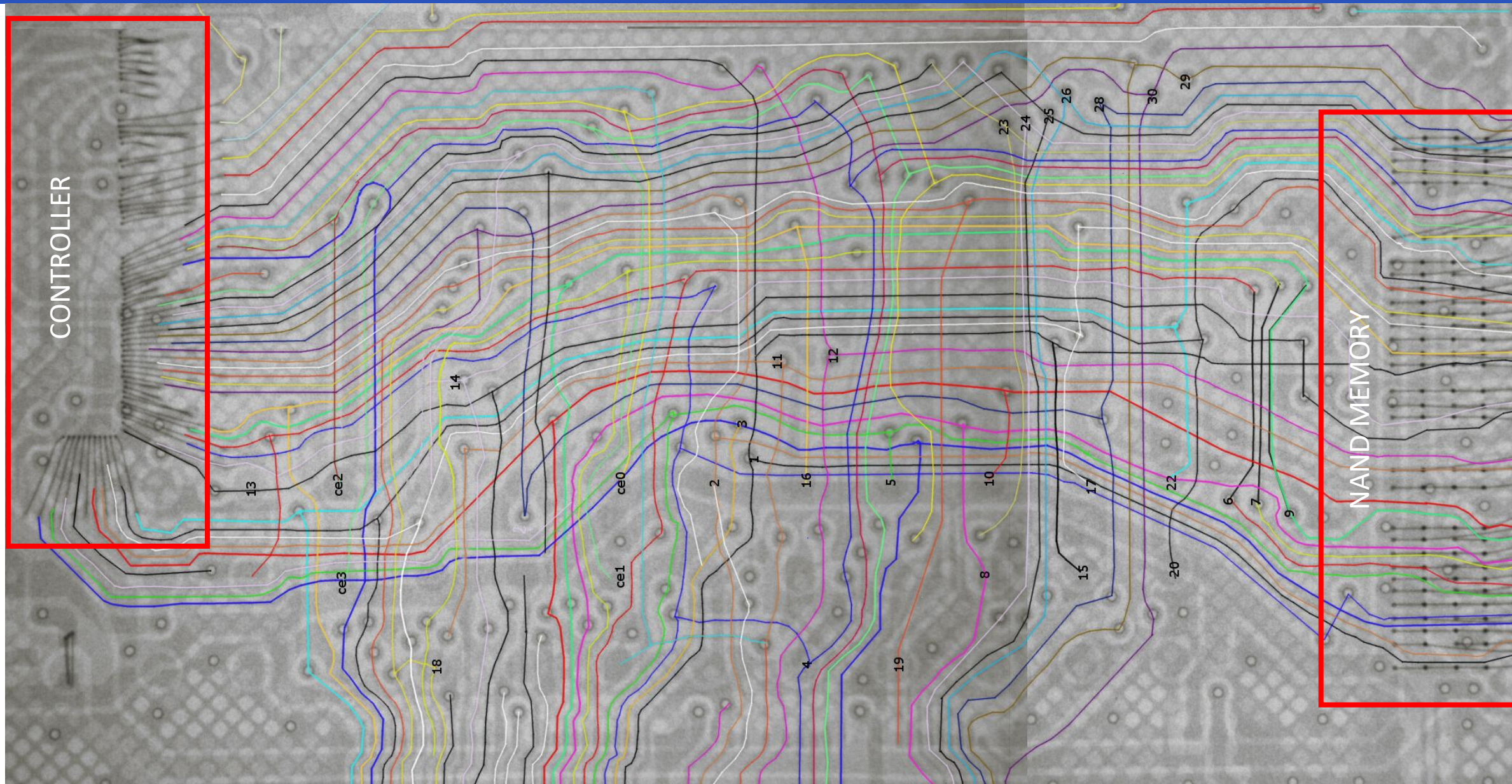
CE0...CE3 – Crystal enable



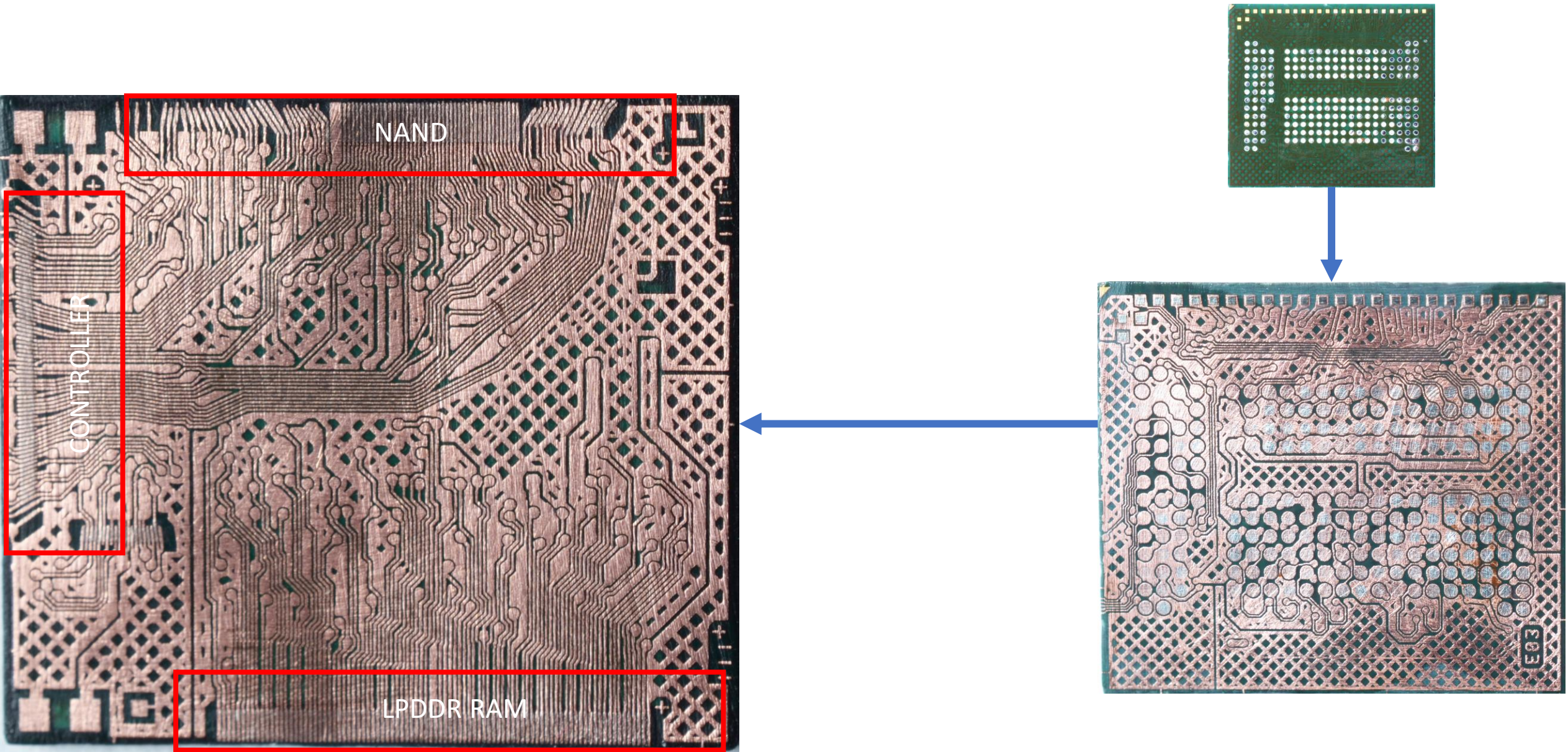
THREE MOST POPULAR METHODS OF UNKNOWN MEMORY PINOUT ANALYSIS

- XRAY inspection - schematics analysis ☹️
- Layer dissection – schematics analysis 😐
- Logic analyzer – signal analysis 😊

XRAY PINOUT ANALYSIS



LAYER DISSECTION – EASIER TO ANALYZE THAN XRAY



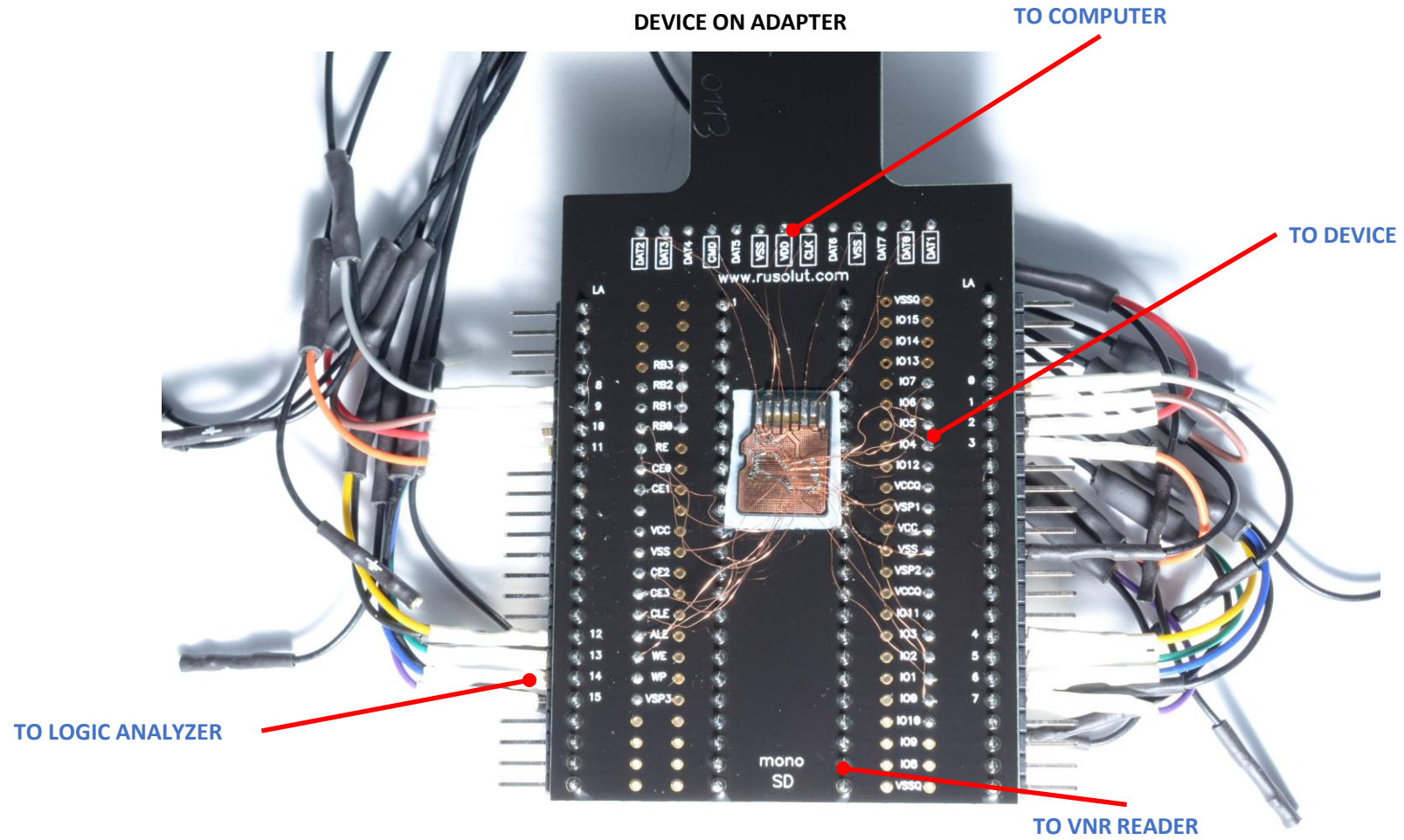
MICROSOLDERING MADE EASY 😊



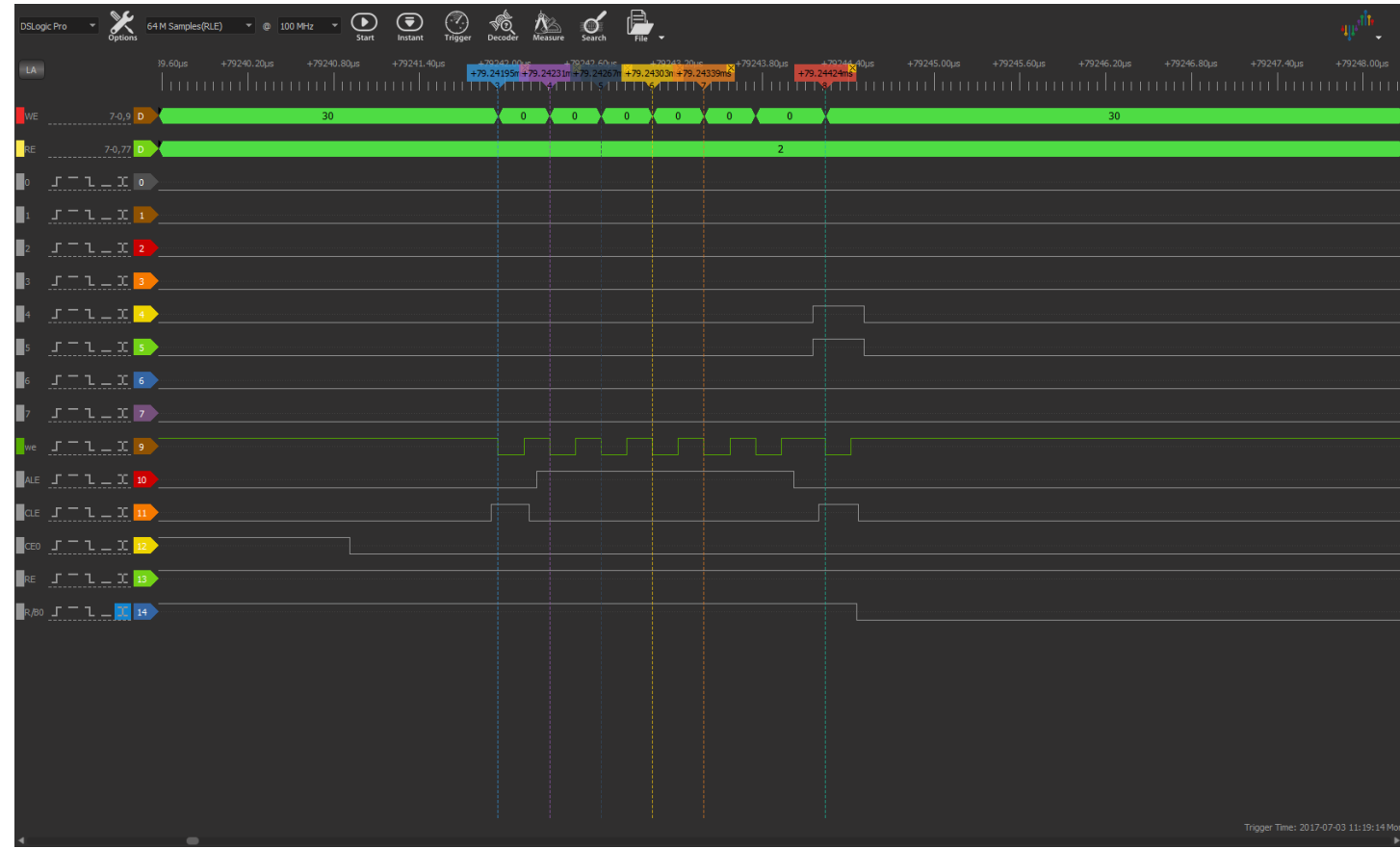
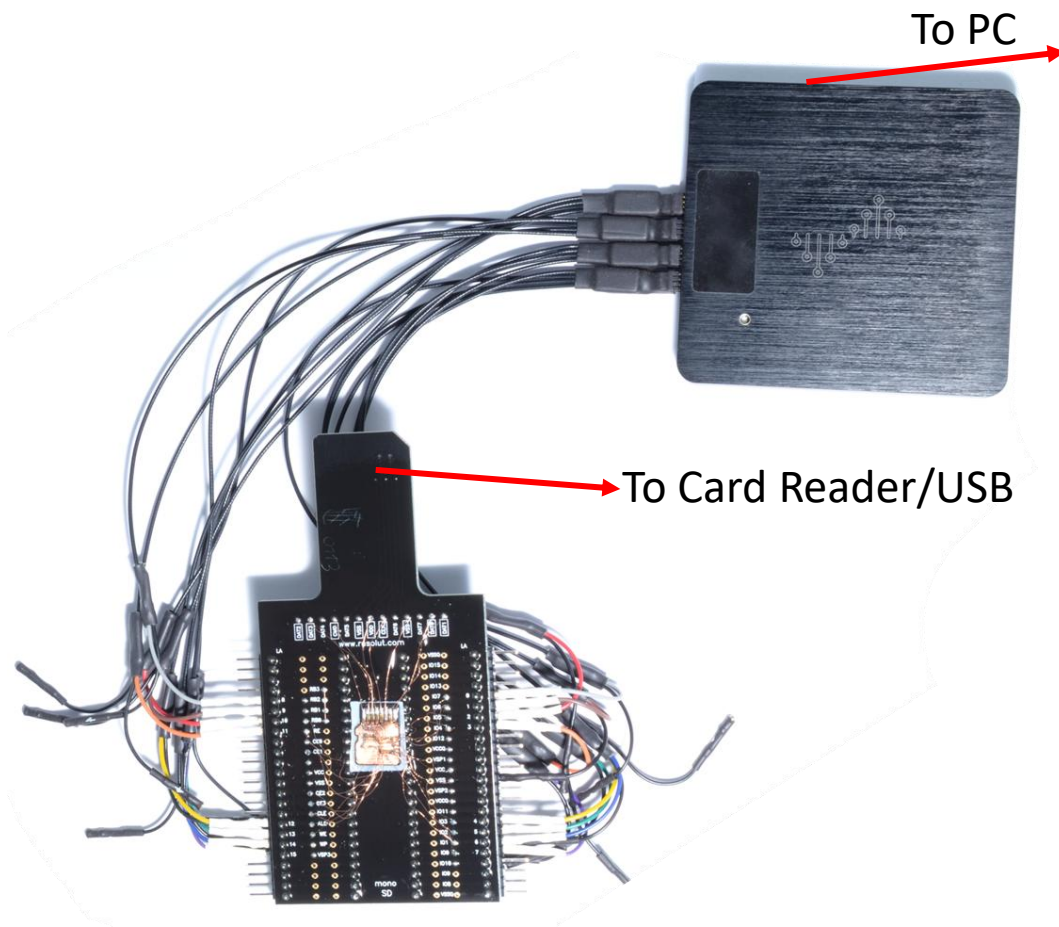
You  Tube



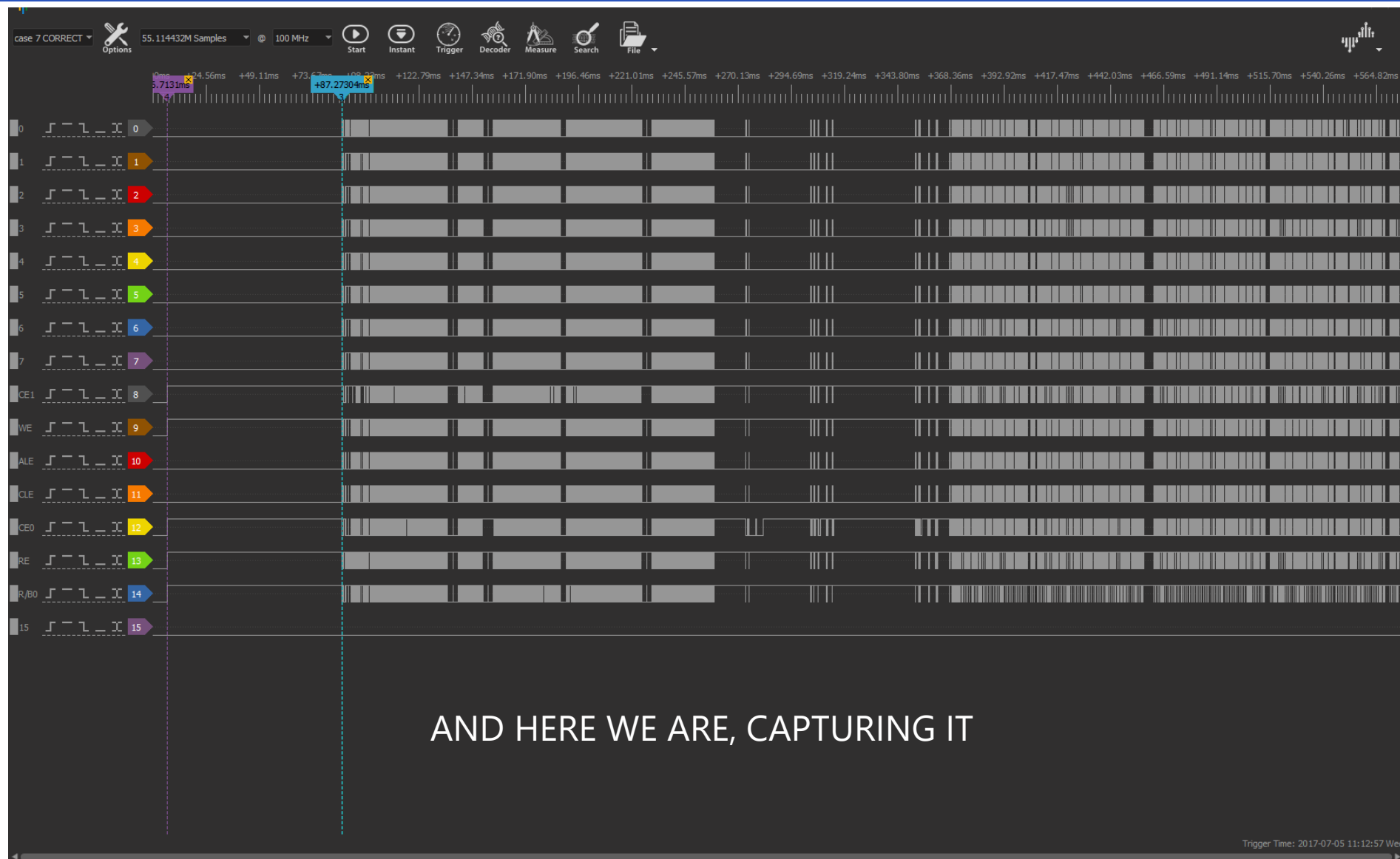
SOLDER DEVICE TO ADAPTER



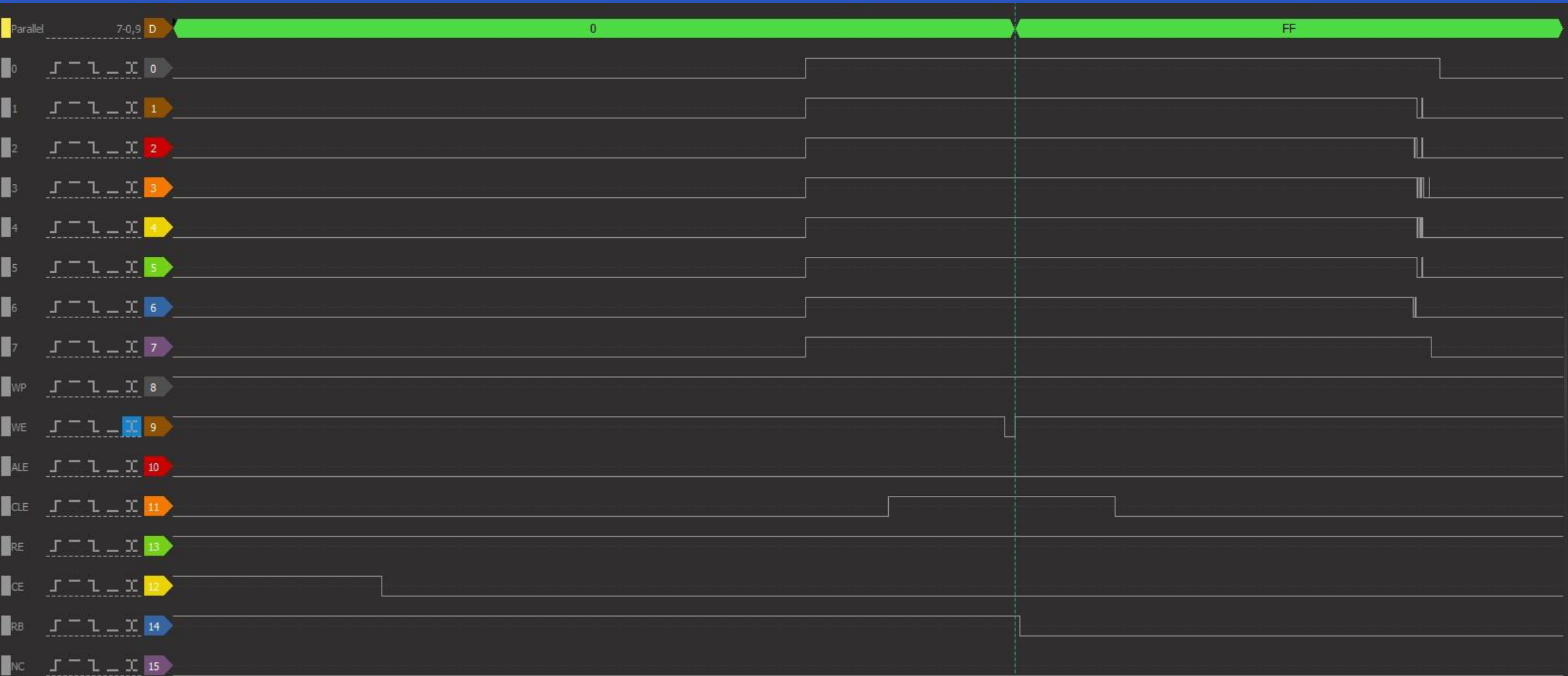
SIGNAL/PROTOCOL ANALYSIS WITH LOGIC ANALYZER



CONTROLLER READS SYSTEM DATA FROM NAND AFTER POWER UP

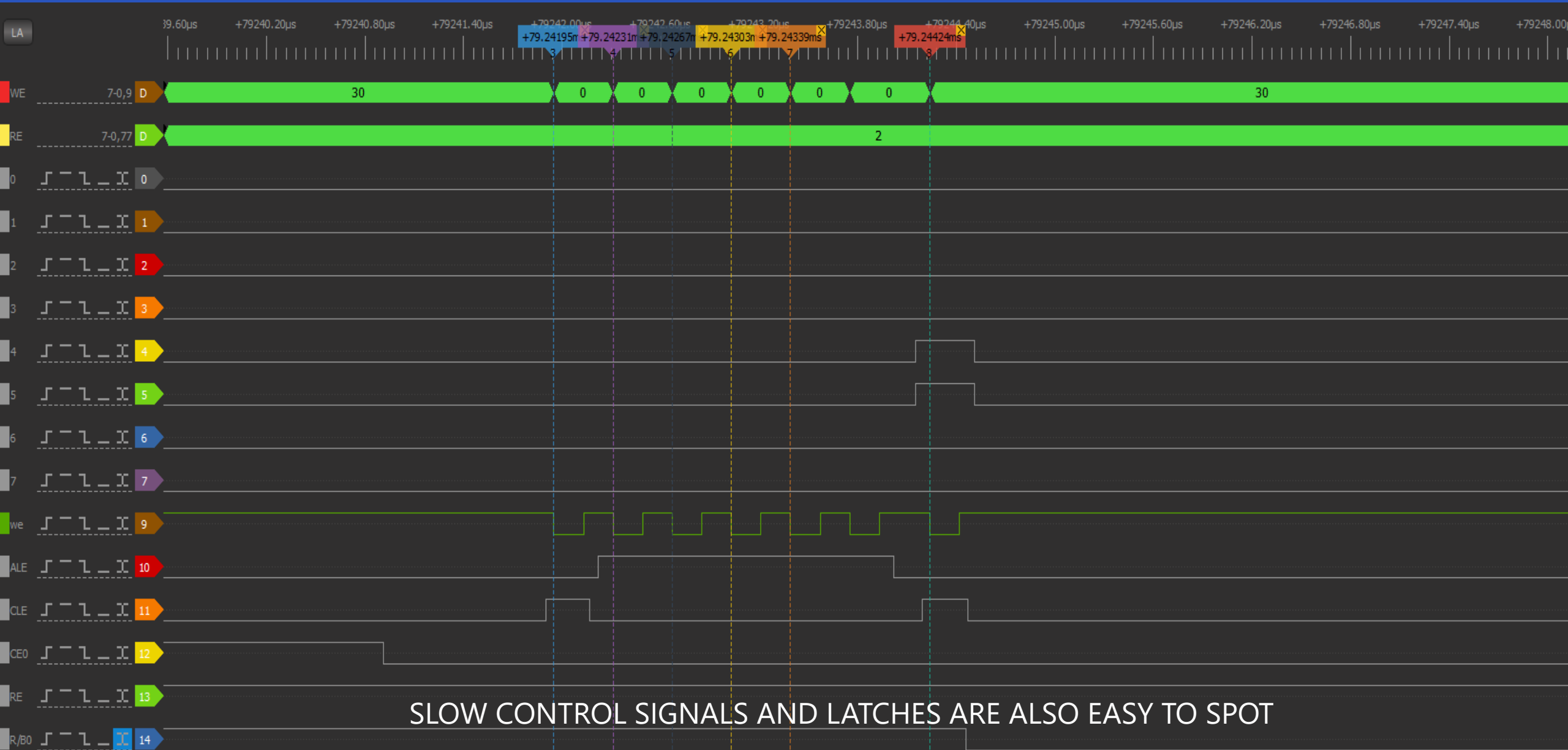


PATTERN ANALYSIS



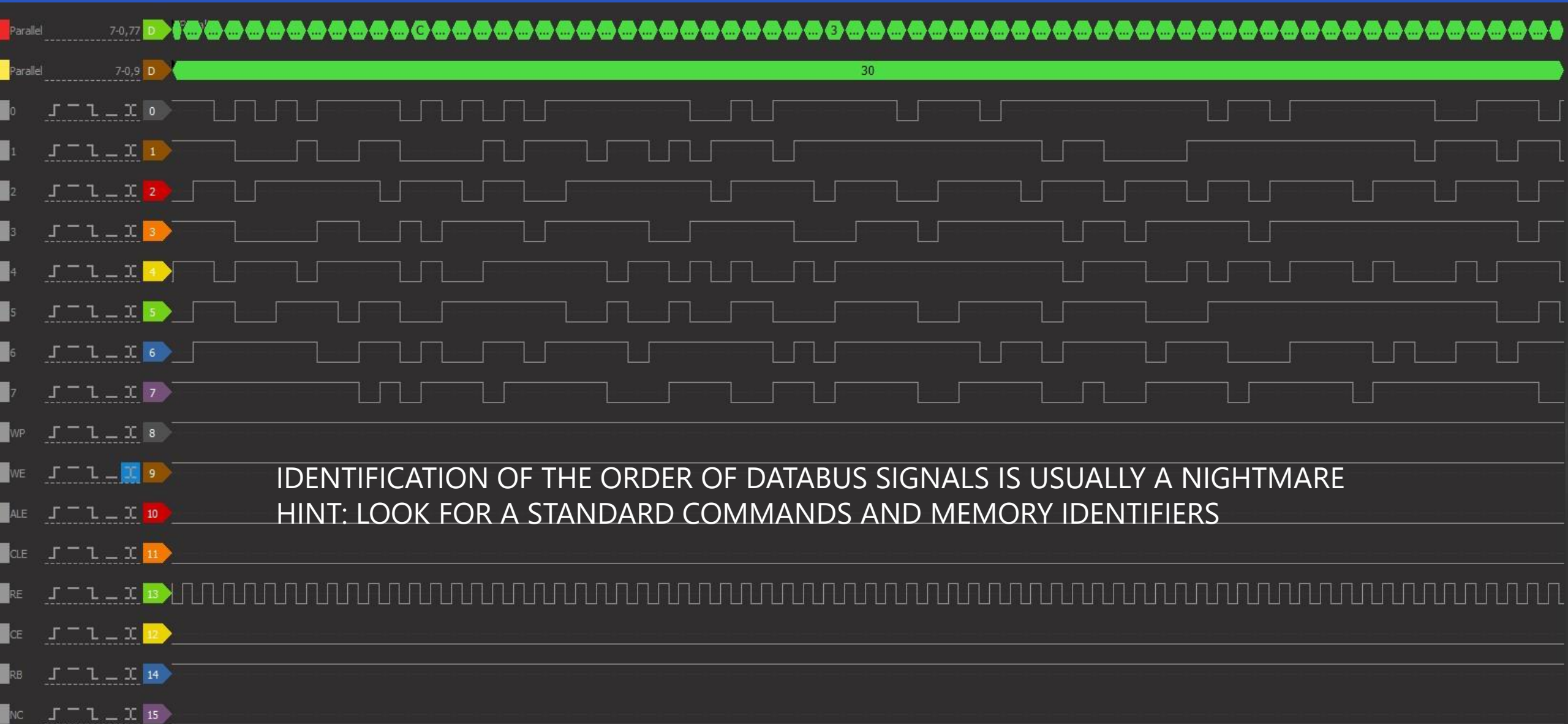
SLOW CONTROL SIGNALS ARE EASILY IDENTIFIABLE

PATTERN ANALYSIS



SLOW CONTROL SIGNALS AND LATCHES ARE ALSO EASY TO SPOT

PATTERN ANALYSIS



ONCE THE RIGHT PLACE WITH COMMANDS IS FOUND PINOUT CAN BE BRUTE FORCED



Pinout analyzer X Workspace

Find databus pinout

Hex

Known lines	Logical analyzer	Pinout
Data bus	Channel	Channel
<input type="text"/>	CH0	<input type="text" value="CH0"/>
<input type="text"/>	CH1	<input type="text" value="CH1"/>
<input type="text"/>	CH2	<input type="text" value="CH2"/>
<input type="text"/>	CH3	<input type="text" value="CH3"/>
<input type="text"/>	CH4	<input type="text" value="CH4"/>
<input type="text"/>	CH5	<input type="text" value="CH5"/>
<input type="text"/>	CH6	<input type="text" value="CH6"/>
<input type="text"/>	CH7	<input type="text" value="CH7"/>

Read ID bits

Detected ID

Unknown ID

Find pinout Clear Hex to Bits Bits to Hex

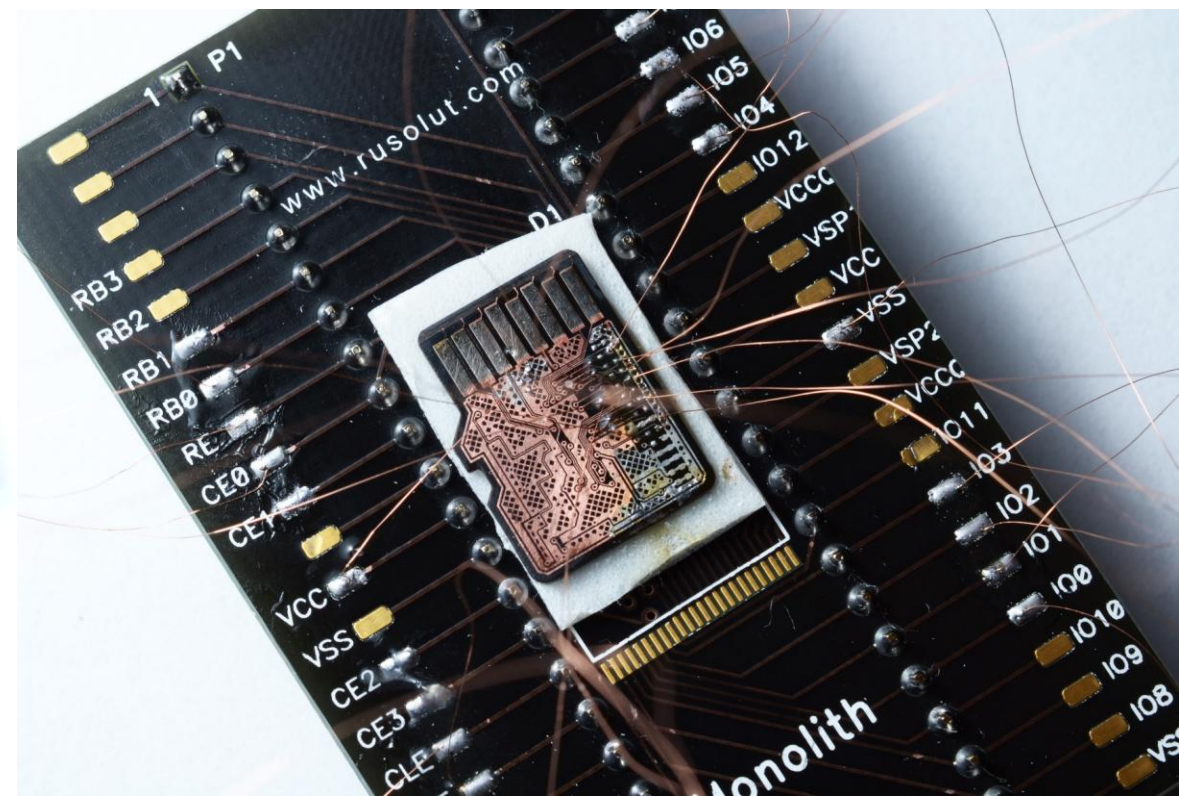
Hex/Bit calculator 1

Hex

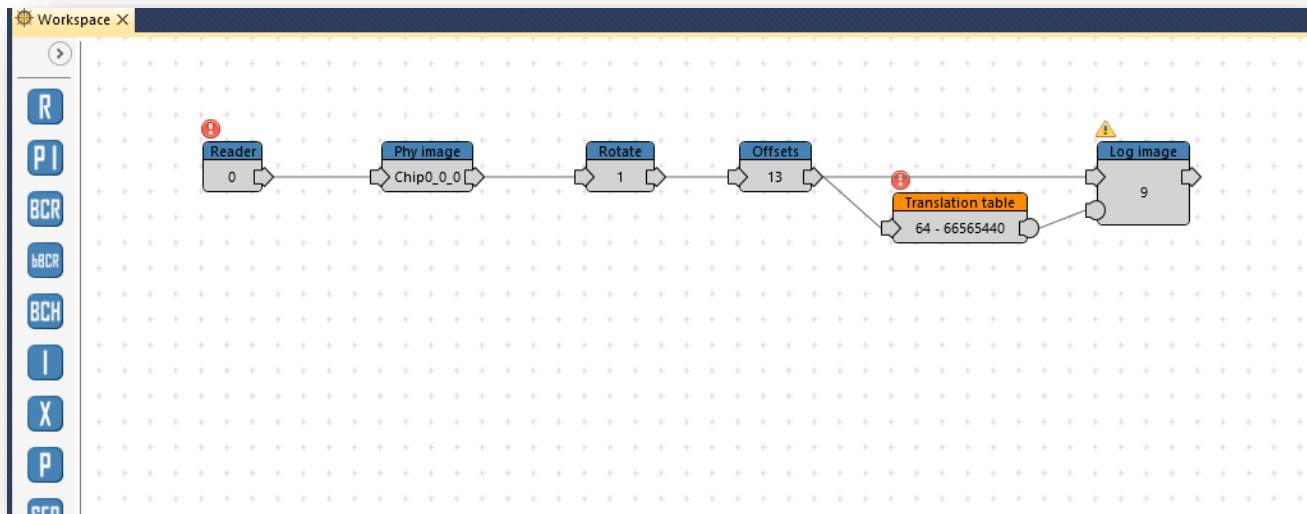
IO	Bit
<input type="text" value="IO0"/>	
<input type="text" value="IO1"/>	
<input type="text" value="IO2"/>	
<input type="text" value="IO3"/>	
<input type="text" value="IO4"/>	
<input type="text" value="IO5"/>	
<input type="text" value="IO6"/>	
<input type="text" value="IO7"/>	

Hex to Bits Bits to Hex Clear

READ MEMORY AND GET DATA!



FILE SYSTEM RECONSTRUCTION



MBR ▶ Volume2 (QNX6) 2.10 GB ▶

- ▲ Dump
 - ▲ MBR
 - ▲ Volume0 (QNX6) 1.46 GB
 - ▶ Root
 - ▲ Volume1 (QNX6) 6.00 MB
 - ▶ Root
 - ▲ Volume2 (QNX6) 2.10 GB
 - ▶ Root
 - ▶ .boot
 - ▶ audio
 - ▶ core
 - ▶ displaymanager
 - ▶ itr
 - ▶ logging
 - ▶ media
 - ▶ messaging
 - ▶ mirrorlink
 - ▶ networking

THANK YOU!

VISIT OUR BOOTH!