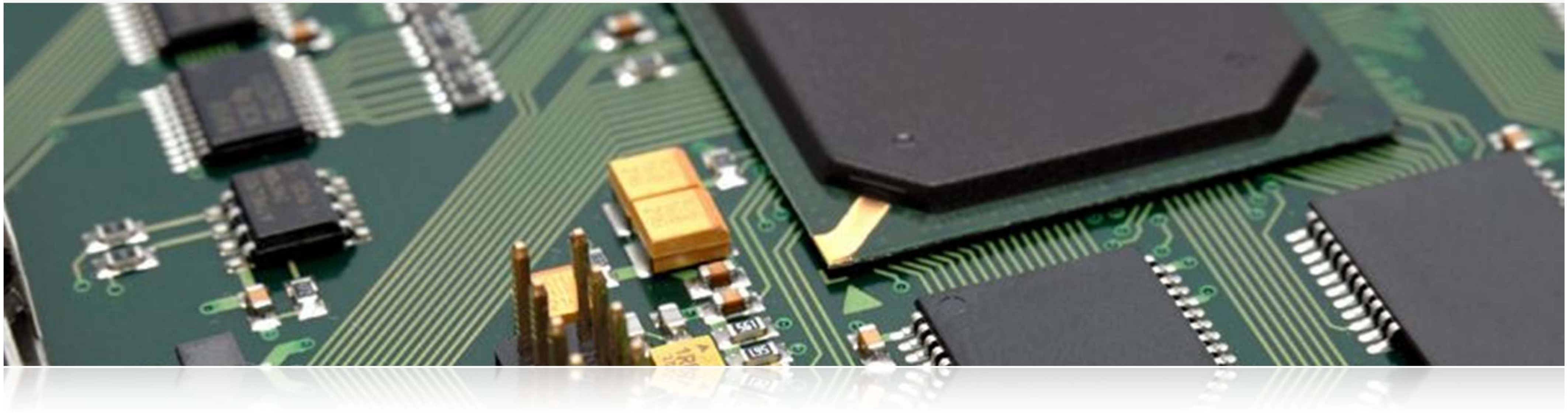


Testing and Programming:

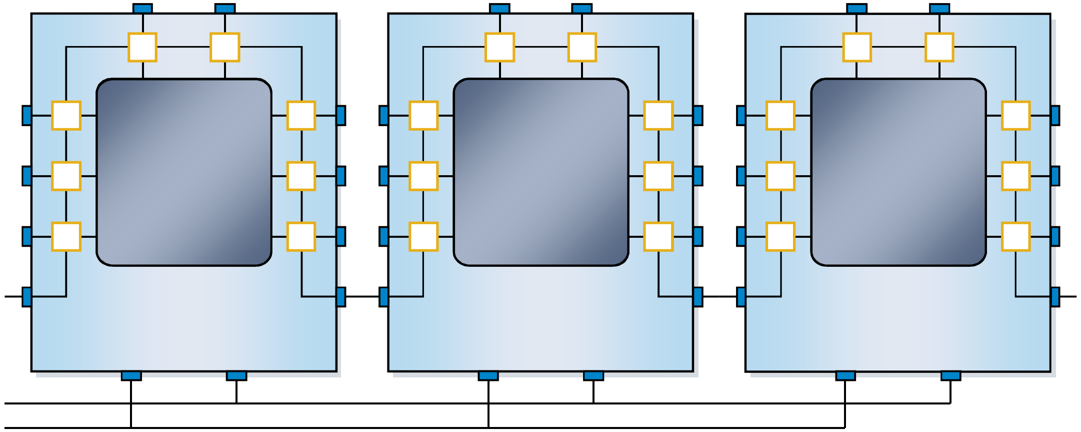
JTAG handles latest memories and flash devices

XJTAG Inc.

What is JTAG?

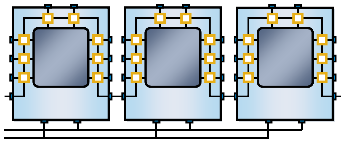
JTAG is a serial communication protocol:

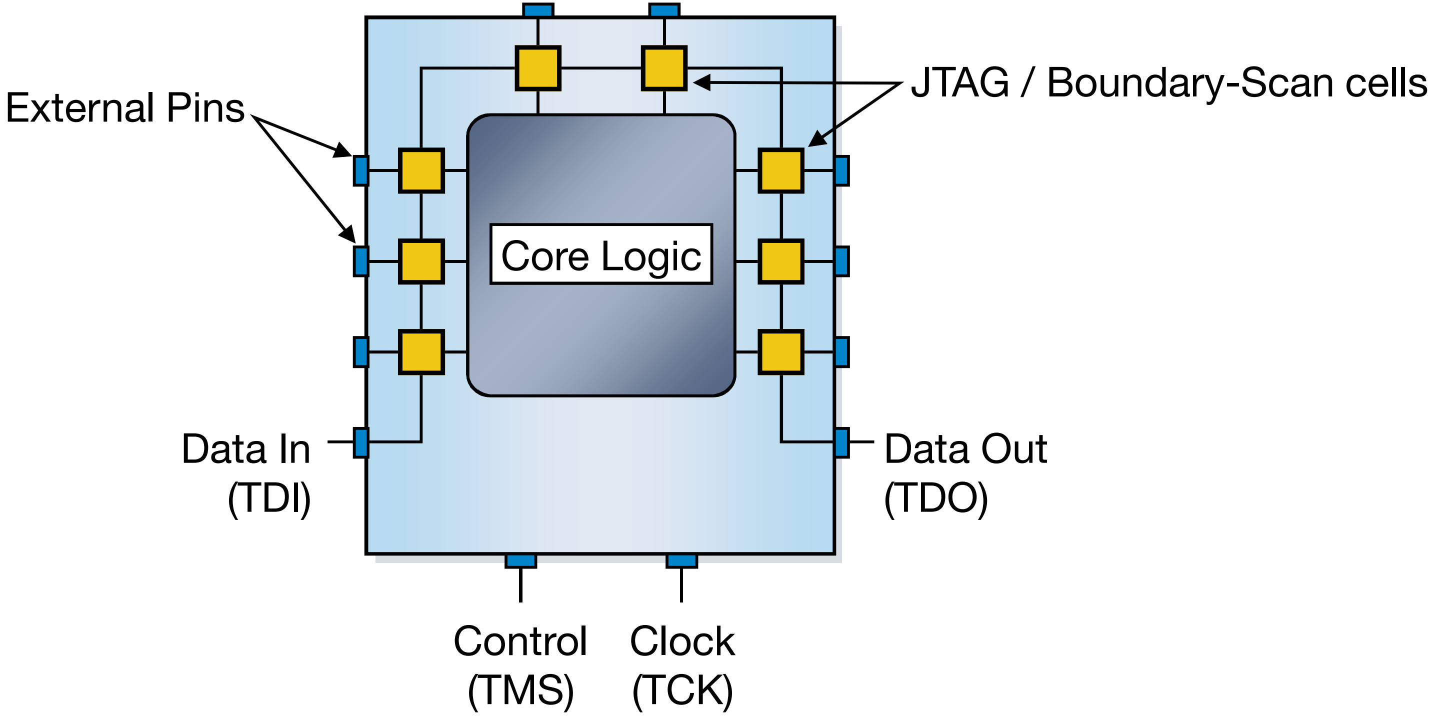
- Designed to allow connectivity testing of PCBs
- Gives set + read access to pins/balls of JTAG devices
- Disconnects device core logic so that I/O is controlled by the JTAG system
- Allows JTAG devices to be used to test non-JTAG devices by driving/reading shared nets on the PCB
  - Many types of device can therefore be tested



What is JTAG?

- 4 JTAG data signals (TDI, TDO, TMS, TCK)
- JTAG Cell(s) inserted between core logic and pin
- Found on all CPLDs, FPGAs, most CPUs, some other devices
- Devices are daisy-chained together on a board





The need to test memory

- ✓ **Testing prototype at board bring-up**
- ✓ **Production line testing**
- ✓ **In-system programming**

- Requirement is to test PCB assembly / connectivity, not the correct operation of memory device
- Diagnosis of errors is also important
  - Functional tests often detect a fault but have difficulty locating it

JTAG-Testable memory

Includes:

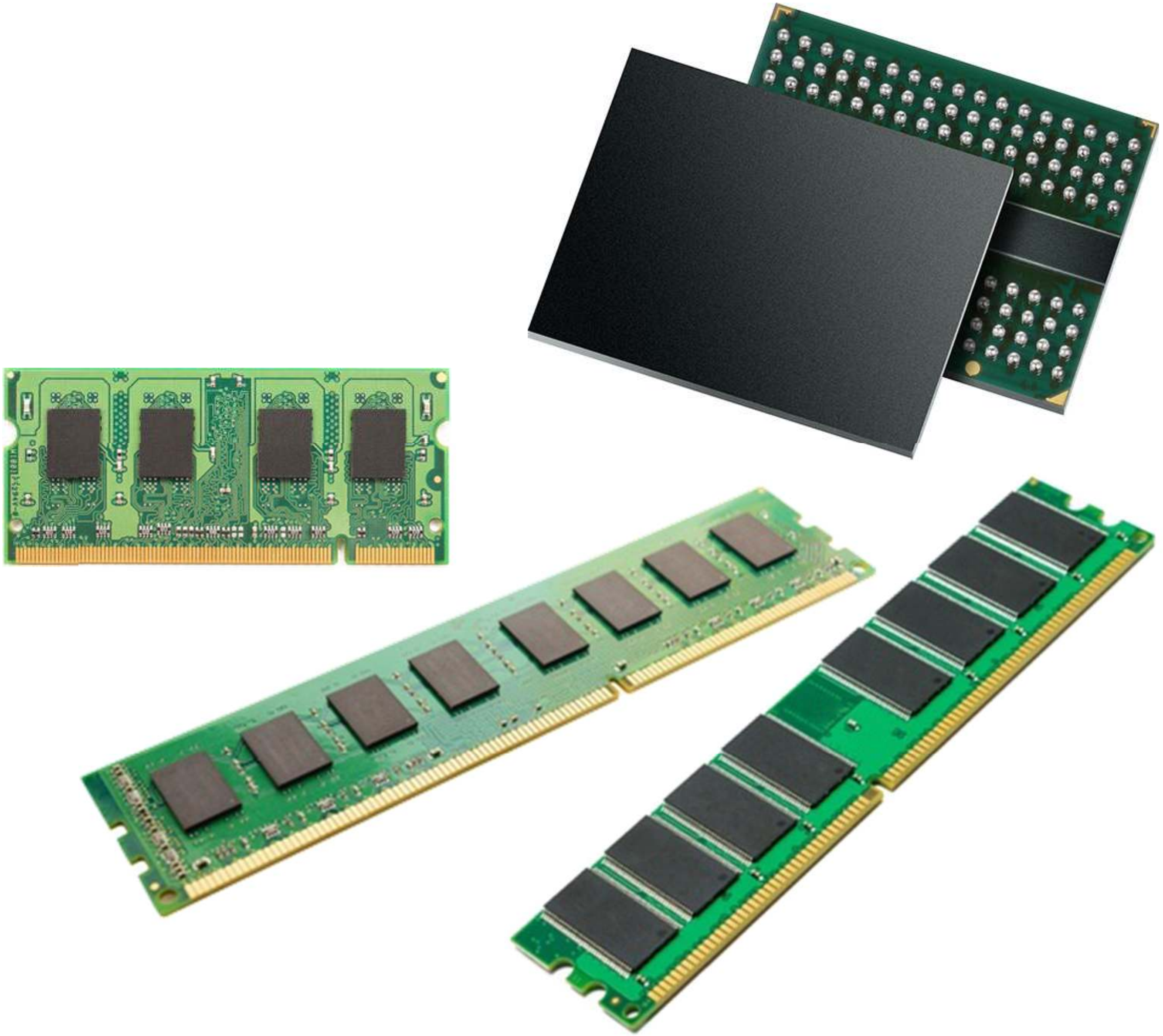
SRAM

DRAM

- SDRAM
- DDR
- DDR2
- DDR3
- DDR4
- GDDR5

EEPROM

Flash



Testing DRAM?

Typical questions:

- How can you test DDR memory at JTAG speeds?
- Data retention time
- Number of accesses

- Actual data retention time?

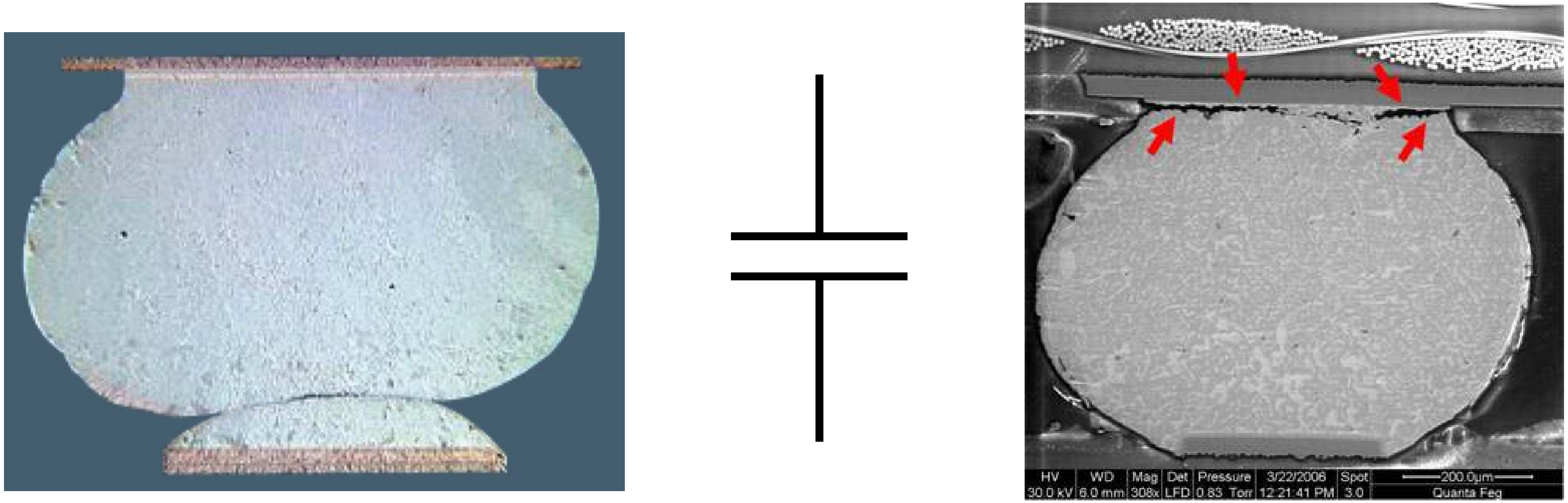
Testing at DC / testing “at speed”

- JTAG testing of memory is at low speed
- DDR memory operates at GHz frequencies
- Circuit behaviour changes at different speeds.

So – how can we claim to test DDR using JTAG?

Testing DDR with JTAG

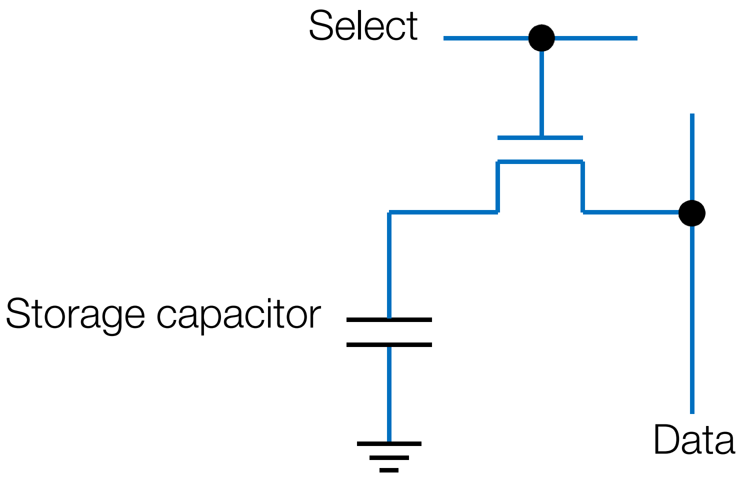
- Low-speed testing finds some faults that a short high-speed test may not



- Most faults will be found by both modes
- High-speed testing in a JTAG system can be achieved by using on-board programmable devices

But DDR is dynamic memory!

- A DRAM cell stores data as charge on a capacitor



- JEDEC specification for DDR→DDR3 gives a minimum retention time of 64 ms.
- After this the DRAM cell is **not guaranteed** to hold the value written to it

Testing dynamic memory

A typical JTAG scan on a board takes 0.2 ms to 2 ms.

- This theoretically gives time for ~ 30 to 300 JTAG scans between writing and reading back

Writing a word to DDR may take around 20 scans.

However –

- 64 ms retention time is a minimum over all cells – i.e. everything will exceed it!
- Likely to be higher if not many accesses are happening
  (see recent "rowhammer" attack details)
- In practice retention time is hundreds of milliseconds through to tens of seconds or more (hence "cold-boot" attacks on PCs)
  – varies between cells & devices
- Retention time is easily adequate to test via JTAG

DRAM – so why refresh?

You’re telling me that I don’t need to refresh my memory for a second or two during testing…

… but that I need to do it every 64 ms when running normally?

Google, 2009:

“…an average DIMM experiences 3750 correctable errors per year…”

Newer DDR types (1): DDR4

- DDR4 can be tested in similar way to other SDRAM types

- However sometimes we can do better
    - DDR4 also has TEN (test enable) pin… sometimes!
    - Pin enables “Connectivity test mode” in device to allow verification of connection to board.
    - TEN pin not present on DIMM interface
    - TEN pin optional/not present on smaller devices
        - But if putting memory directly onto PCB, use it!

Connectivity Test Mode in DDR4

- Divides pins into 2 sets – input and output
- Places asynchronous logic between these pins
- Logic functions are defined in DDR4 spec
- Should be quicker/easier to test than connectivity tests used with previous memory types

But:

- DDR4 test mode places requirements on JTAG implementation of differential signals in CPUs:
  - Unclear whether manufacturers will get this right!

Newer DDR types (2): LPDDR4

- LPDDR4 does not have a test enable (TEN) pin
- Timing requirements for refresh remain at 64 ms
  (same as DDR3 and previous)
- So testable in the same way as DDR3 or earlier, by writing and reading values to test busses and control signals.

Newer DDR types (3): GDDR5

GDDR5 supports “boundary scan” operation

- NB this is **not** JTAG Boundary Scan (1149.1)!
- We can set signals to the pins (address, data, control signals) and then read out the signals as-received at the device pins for verification

Allows for easy automation of connectivity test via a JTAG-compliant device.

Testing non-volatile memory with JTAG

- Because the data is not volatile, timing requirements are easier
- JTAG systems all support most types of non-volatile memory

- Pre-programmed devices
    - Most pins on a device can usually be tested
    - Various options allow testing – without wiping the device

Programming

- JTAG can be used to program most sorts of non-volatile memory.
- JTAG is fine for programming configuration partition, serial number, MAC address etc.
- But conventional JTAG is too slow for programming large amounts of data

Accelerated Flash programming

Different vendors may have different meanings of this term.

- The PC may download some code to the JTAG controller to reduce latency/communication between them.
- Better solution is to program a device on the board and use that to program the flash.
  - Embedded s/w program on CPU
  - Image for FPGA to configure it as programmer
  - Flash image can be loaded via USB/Ethernet, or for an FPGA streamed over the JTAG signals

NAND Flash

- Easily testable via boundary scan
- Could be programmed via JTAG as well…
  - But it is not practical
  - Not only slow but complicated due to bad block management
- Sensible NAND flash solution:
  - Use JTAG to load a boot loader or FPGA programmer image
  - Then load flash image via a standard high speed interface (e.g. Ethernet, USB)

Conclusions

- JTAG is a fast and reliable way to test memory
- Latest memory types (DDR4, GDDR5) are no problem
- Other types of memory can be functionally tested at low speed via JTAG
- If at-speed testing is required, JTAG can be used to program h/w to perform the at-speed test
- Similarly, large flash devices can be programmed at-speed using programmable devices on the PCB