# Test Trimming

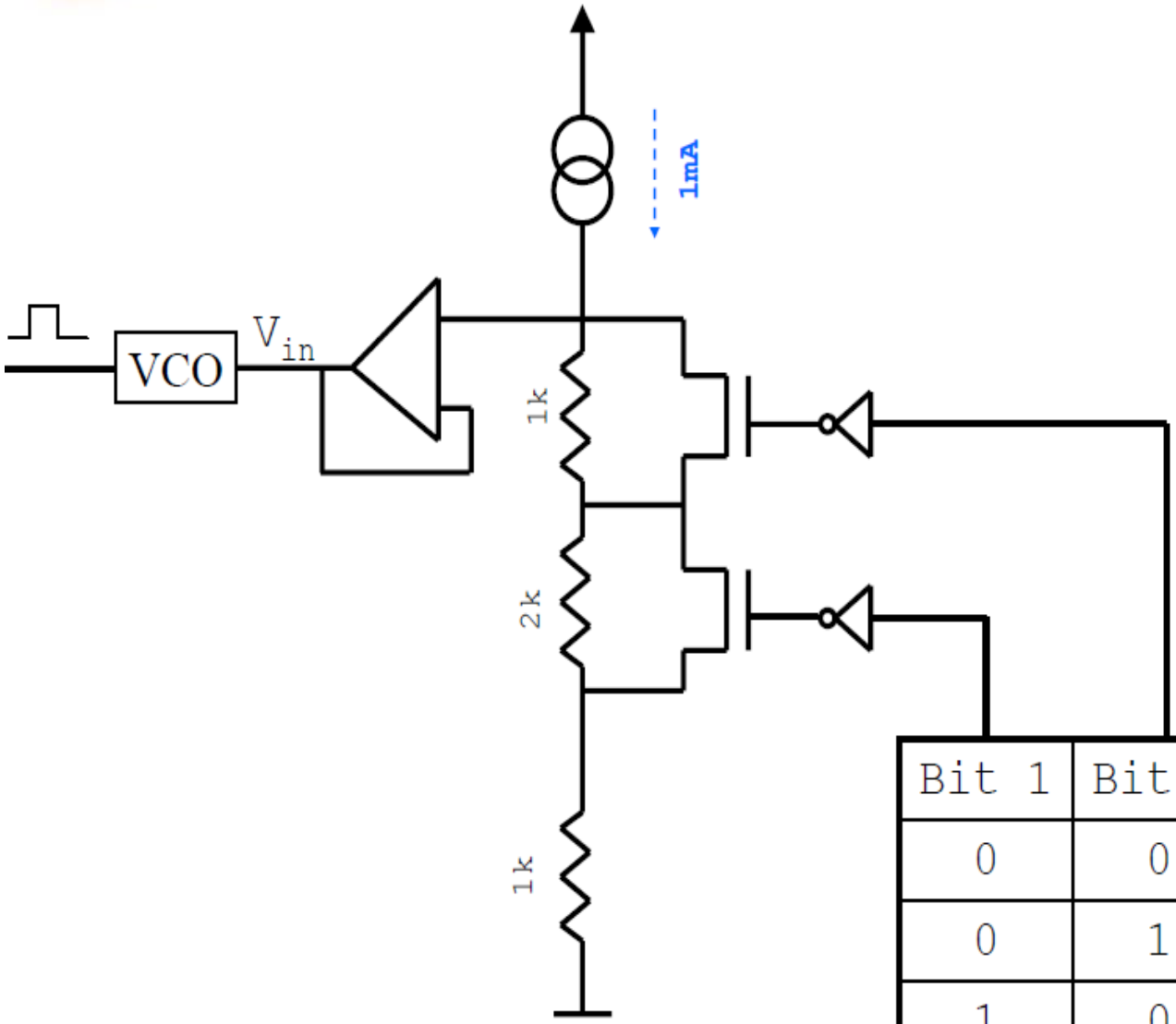

Jerry Gao
2014-9-13

# Why Trimming ?

# Why Trimming

- Trimming allows the quality of a product to be enhanced during testing.
- This is the only aspect of testing that adds value to the device.
- Trimming is frequently performed after packaging to compensate for packaging effects.

# Simple Trimming Circuit




| Bit 1 | Bit 0 | $V_{in}$ | Abs | Rel |
|---|---|---|---|---|
| 0 | 0 | 1V | -1V | -50% |
| 0 | 1 | 2V | 0V | 0% |
| 1 | 0 | 3V | +1V | +50% |
| 1 | 1 | 4V | +2V | +100% |

# Trimming Type ?

# Trimming Type

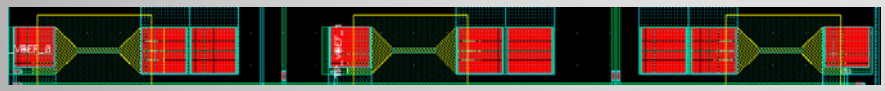
- Fuse ( poly, metal )
- Laser
- Zener
- E/EEPROM

# Trimming Type

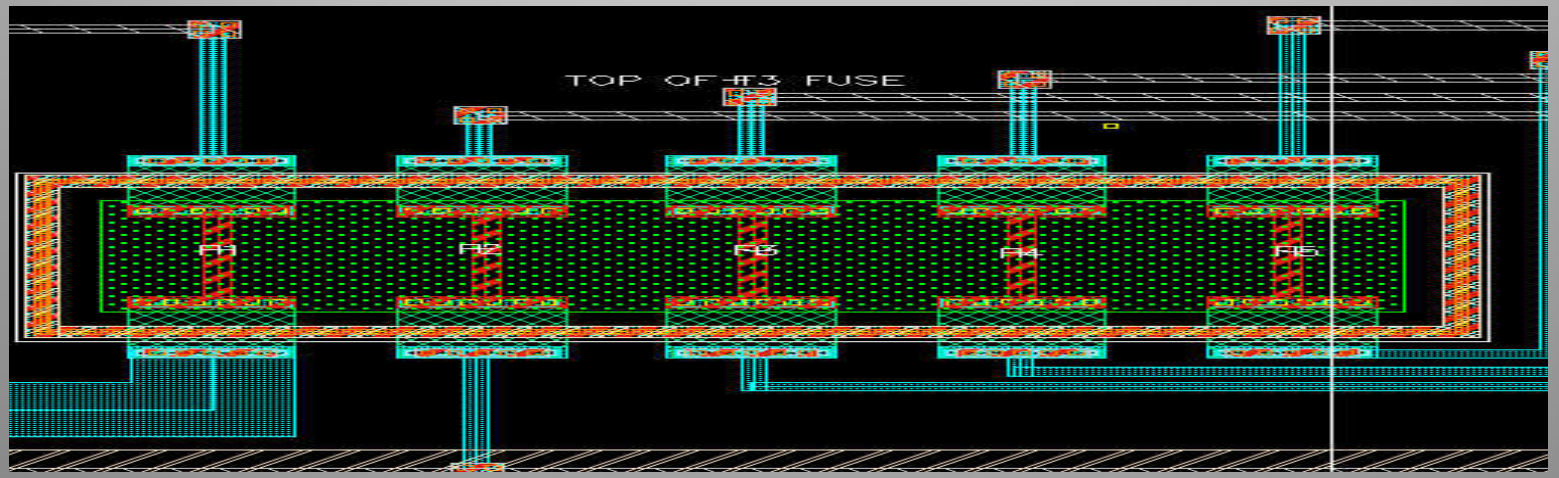
- Fuse

# Fuse

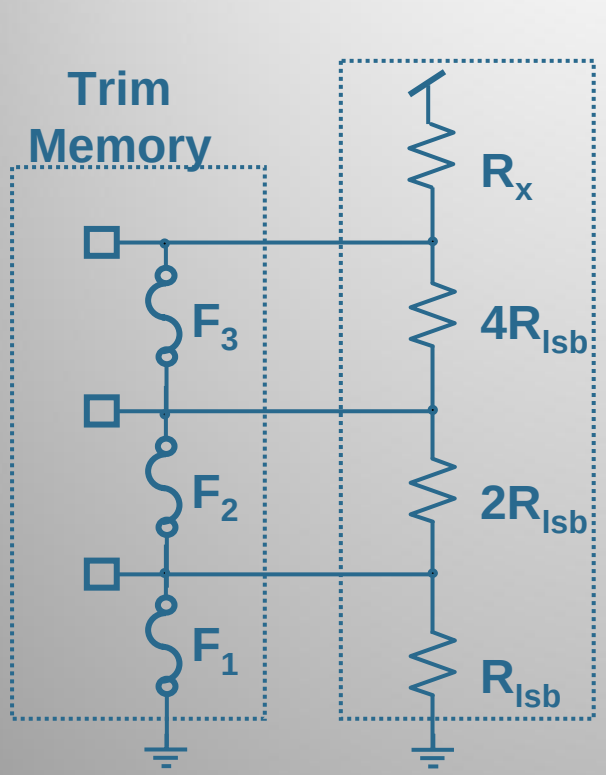
- Poly Fuse



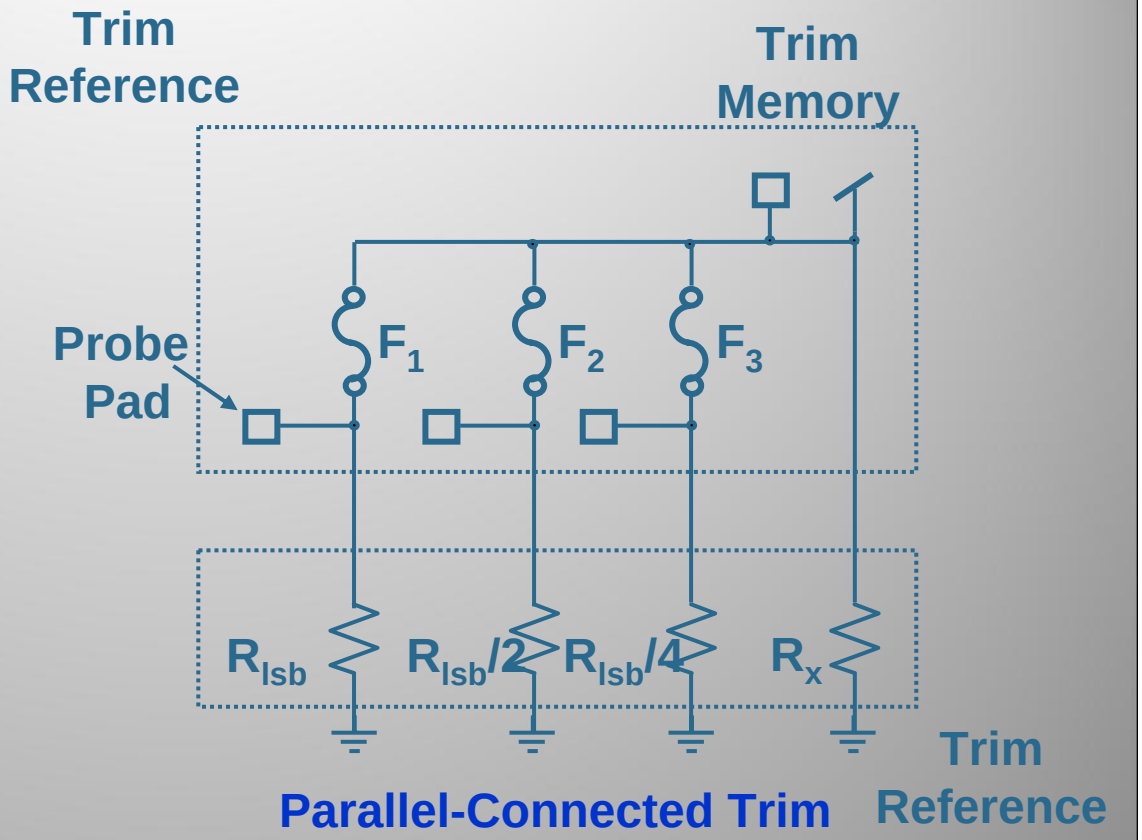
- Metal Fuse



# Binary Weighted Resistor Trim Schemes Using Fuses



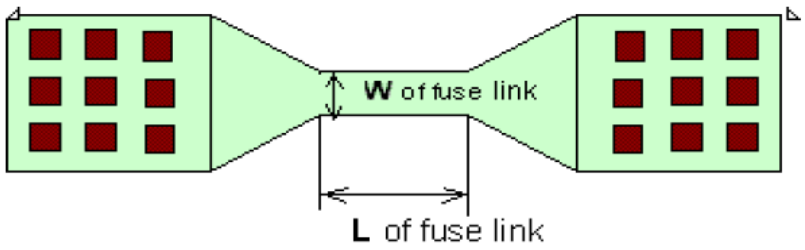

Series-Connected Trim Memory/Ref.

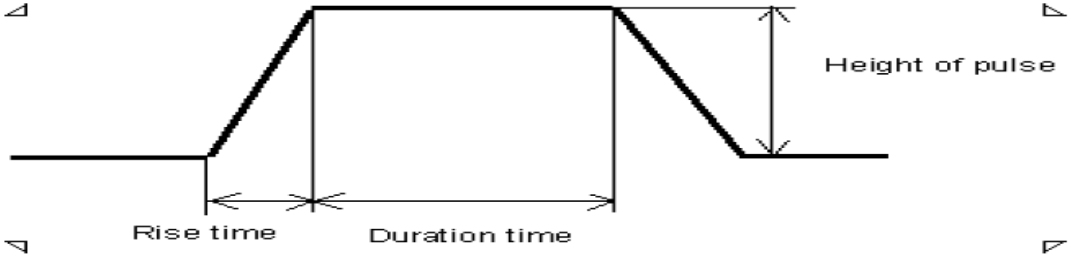



Parallel-Connected Trim Memory/Ref.

- Fuses are blown by forcing a controlled current through them.
  - A blown fuse appears to be an open circuit.
  - Probe pads and control pads for each fuse are required.
- No simulation available (difficult to simulate an open circuit).

# Poly Fuse Specification




## 1. POLY FUSE [RES_FUSE] v1

PDK Device name-RES_FUSE; SPICE Model Name-RES_POLY_L_LBC7; SV Model Name-FUSE

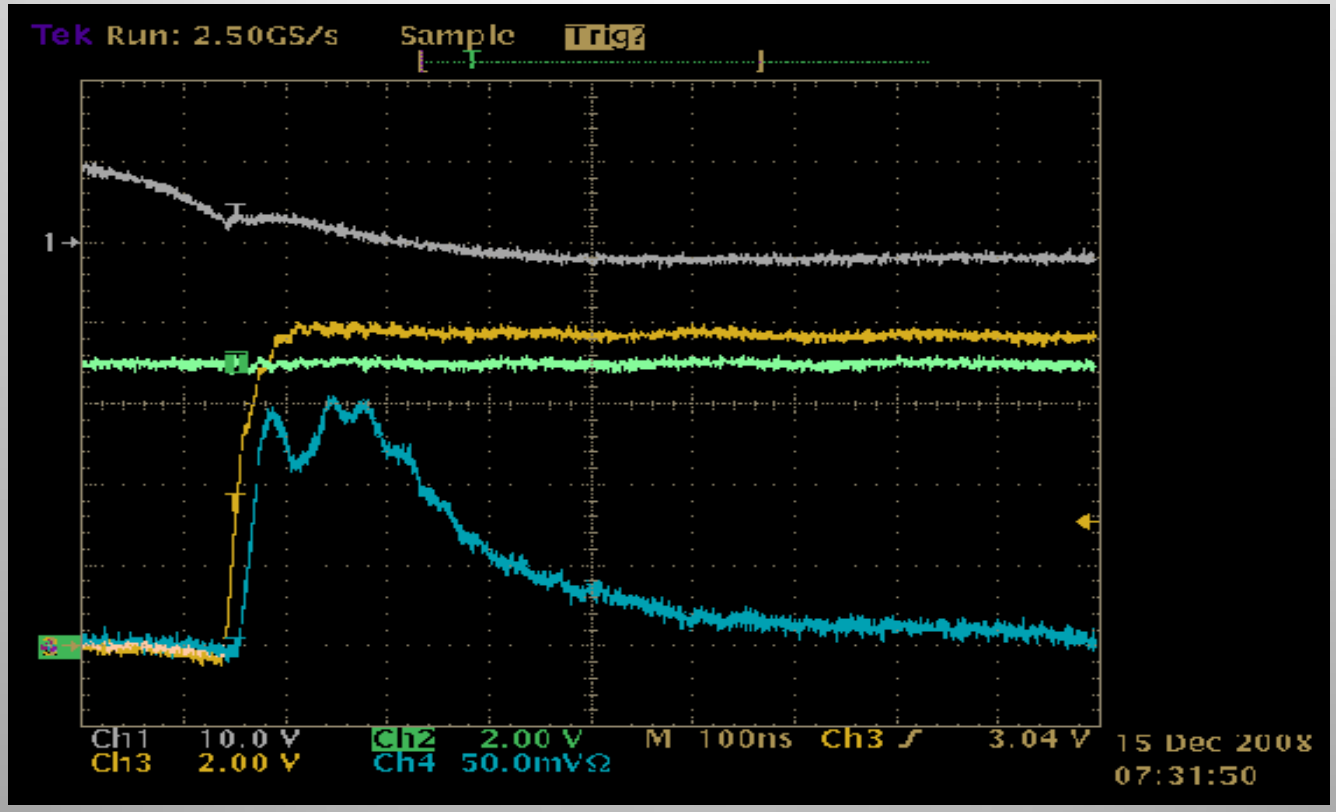
| ALLOWED DEVICE DIMENSIONS | TARGET | LSL | USL | UNITS | COMMENTS | FOOT NOTES |
|---|---|---|---|---|---|---|
| Fuse link W (drawn) | 0.7 | | | um | Exact | |
| Fuse link L (drawn) | 6.5 | | | um | Exact | |

| ALLOWED OPERATING CONDITIONS | TARGET | LSL | USL | UNITS | COMMENTS | FOOT NOTES |
|---|---|---|---|---|---|---|
| Temperature Range | | -40 | 150 | C | | |

Note: Temperature=27C unless specified otherwise.

| FAIL CRITERIA | SVN | ELECTRICAL PARAMETERS | TARGET | LSL | USL | UNITS | COMMENTS | FOOT NOTES |
|---|---|---|---|---|---|---|---|---|
| Monitor | FUSE_P1I | Fuse Resistance Initial | 25 | | | Ohm | @0.1V | |
| Characterize | FUSE_I_I | Leakages Fuse-substrate(well) pre-blow | | | | A | T=27C. Apply 0.7V, measure I. | |
| Monitor | FUSE_R_B | Fuse Resistance Blown | 100 | 1 | | Mohm | @0.1V | |
| Characterize | FUSE_I1I | Leakages from first end of the Fuse to substrate(well) post-blow | | | | A | T=27C. Apply 0.7V, measure I. | |
| Characterize | FUSE_I2B | Leakages from second end of the Fuse to substrate(well) post-blow | | | | A | T=27C. Apply 0.7V, measure I. | |
| Characterize | FUSE_IPL | Fuse Current Pulse Height | 30 | | | mA | | |
| Characterize | FUSE_VPL | Fuse Voltage Pulse Height | 6 | 5.5 | 8 | Volts | | |
| Characterize | FUSE_TR | Fuse Voltage Pulse Rise Time | 50 | | 500 | ns | | |
| Characterize | FUSE_TDU | Fuse Voltage Pulse Duration | 10 | | 50 | ms | | |

# Poly Fuse Blown Waveform

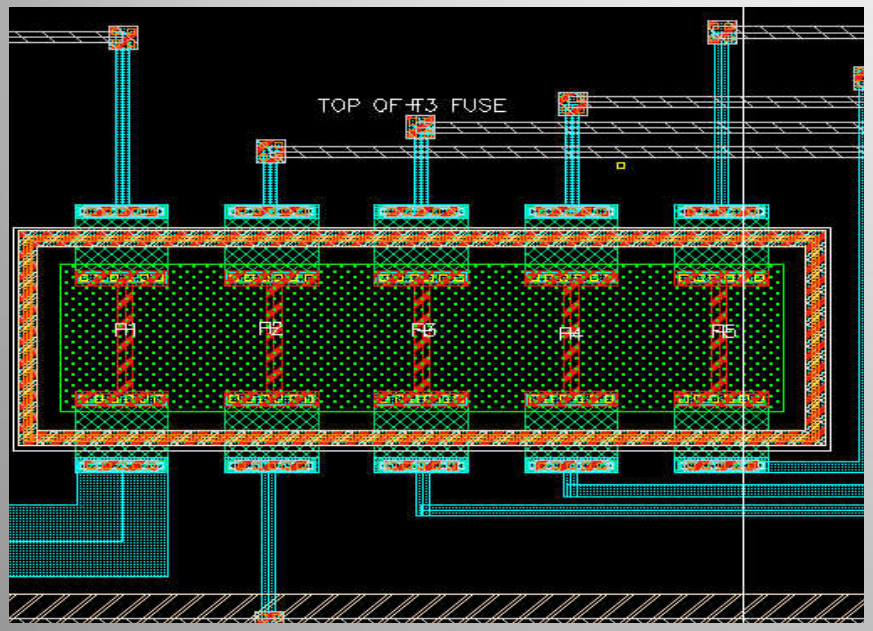


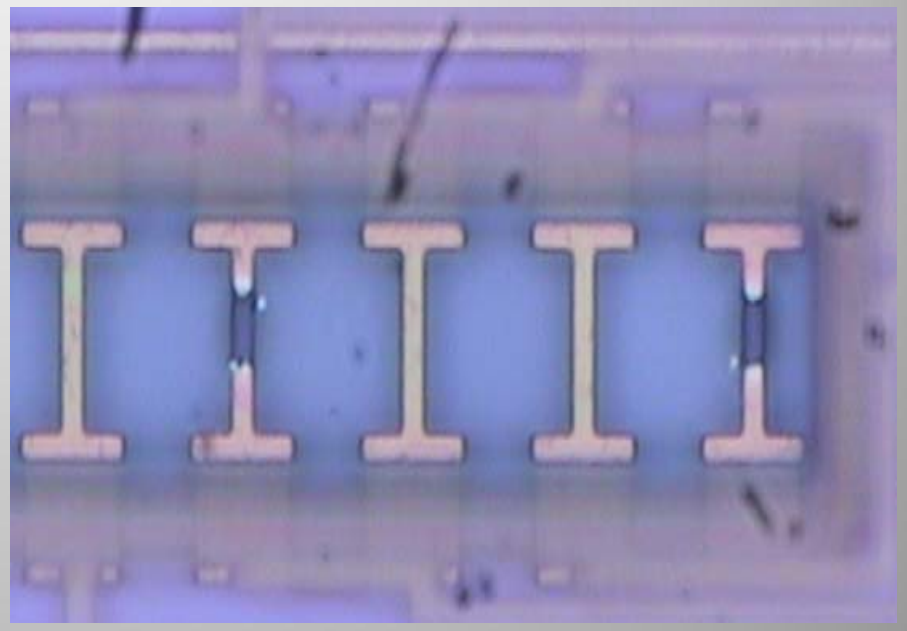

- Approximate 500ns duration of currents pulse during fuse blowing
- Current pulse provide

$$i = C\frac{du}{dt} \Rightarrow U(t) = U(0)e^{-\frac{t}{RC}} \Rightarrow C = \frac{t}{R \times ln\frac{U(0)}{U(t)}} = \frac{500\times10^{-9}}{25\times ln\frac{7}{6.99}} = 14 \times 10^{-6}(uF)$$

For 3 fuses, take one capacitor 47uF

# Fuse Layout
## Before and After Trimming



Layout Of View Of Fuse Memory
**Before Trim**

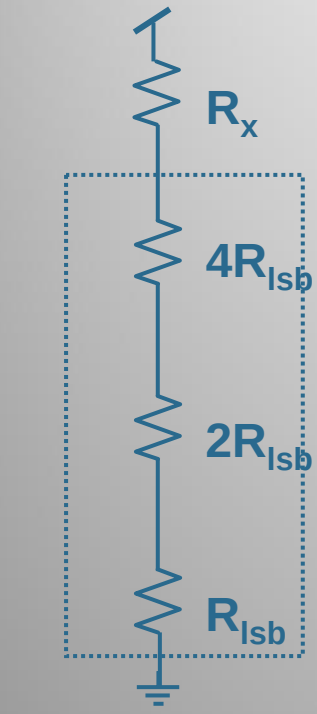
Microphotograph Of View Of Fuse Memory
**After Trim**

(Fuse links blown by laser)

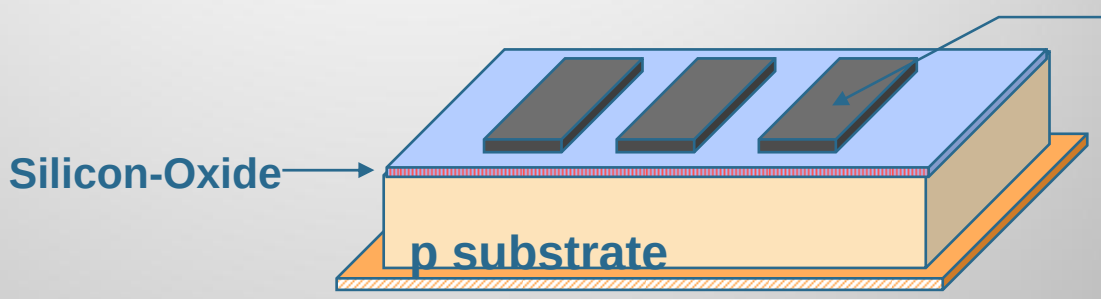
# Trimming Type

- Fuse
- Laser

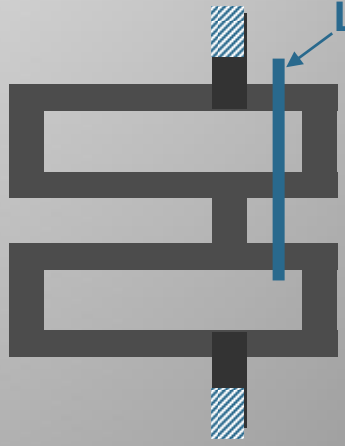
# Laser Trimming Thin-Film Resistors




Trim Memory/Ref.



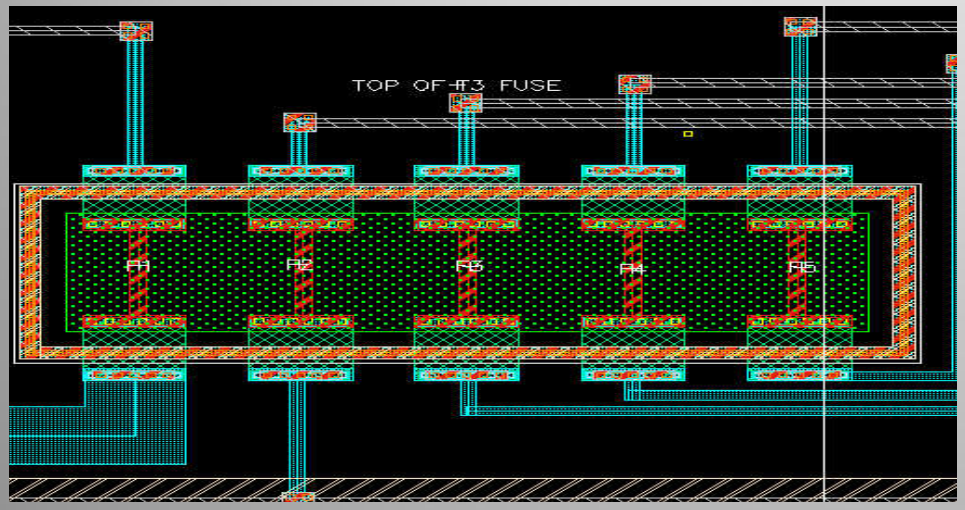


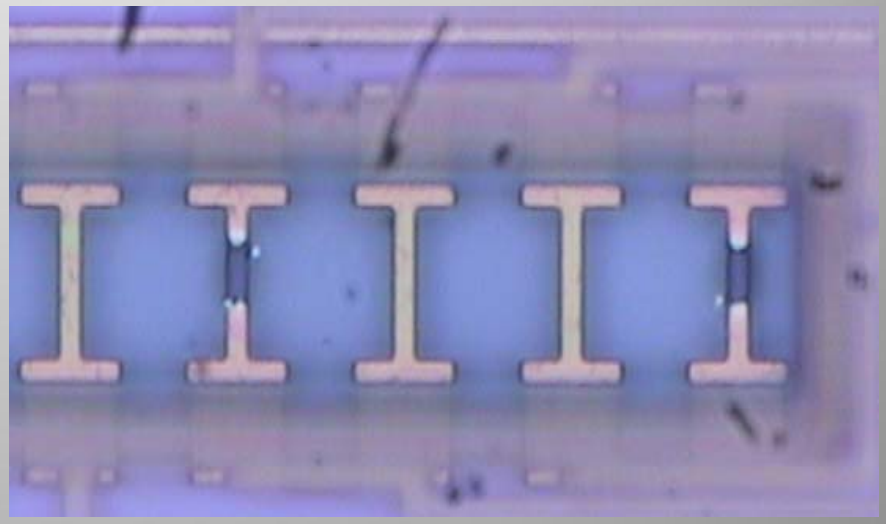



Example Laser Trims

- Cut a thin-film resistor with a laser to increase its resistance value.
- Performed at the wafer level.

# Laser Trimming Metal Fuse

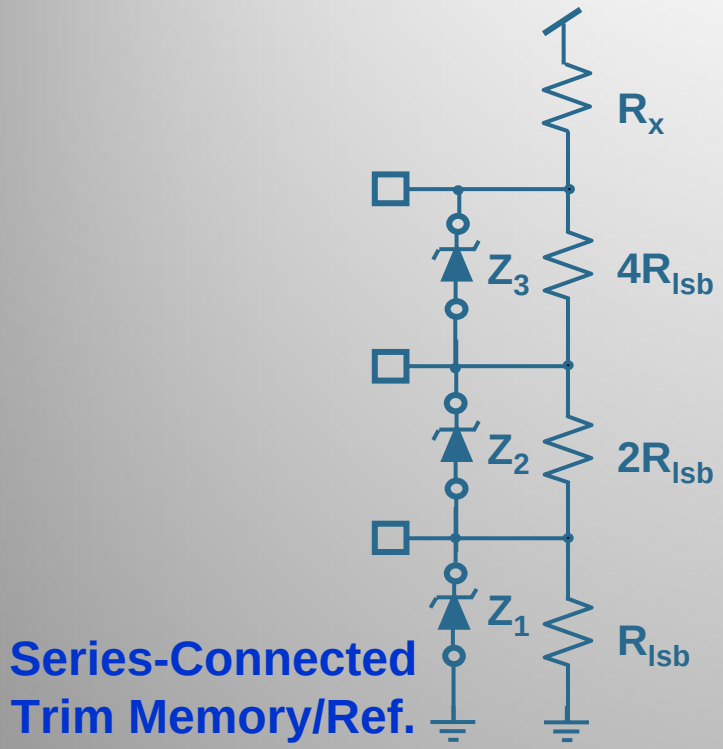


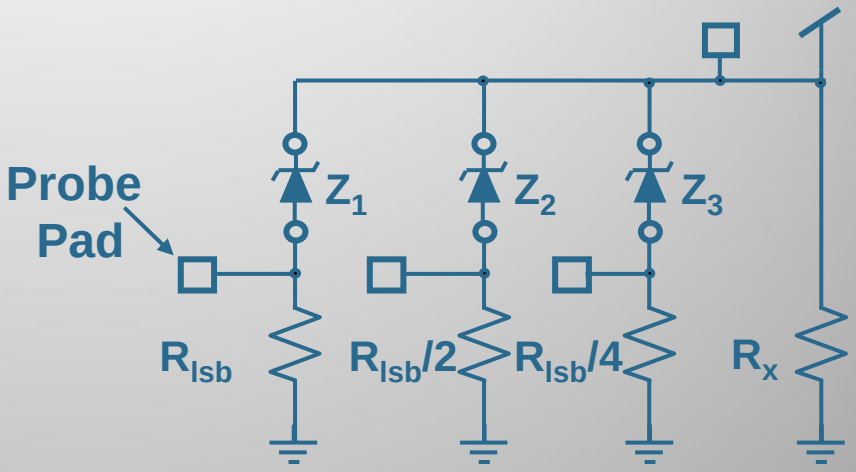
# Trimming Type

- Fuse
- Laser
- Zener

# Binary Weighted Resistor Trim Schemes Using Zener Diodes



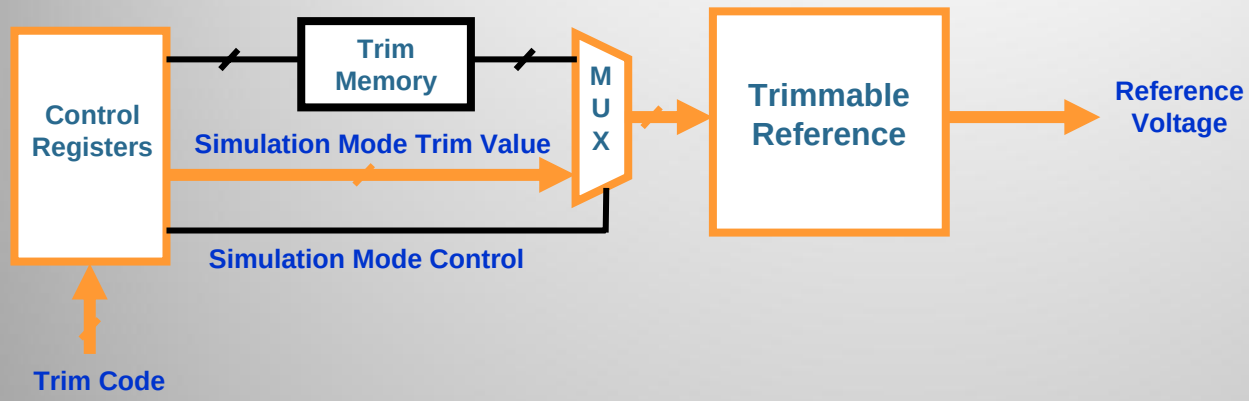

**Series-Connected Trim Memory/Ref.**



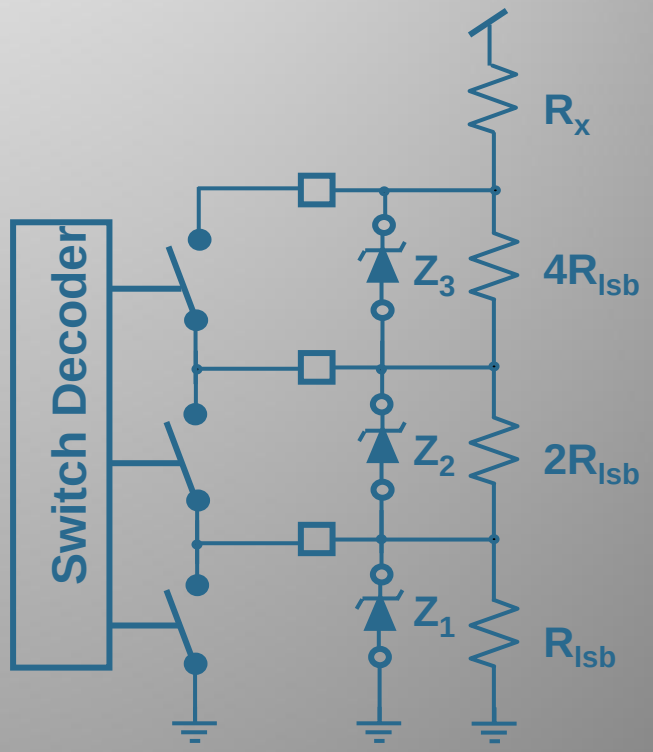

**Parallel-Connected Trim Memory/Ref.**

- Zener diodes are blown by forcing a controlled current through them.
  - A blown zener diode appears to be a short circuit.
  - Probe pads and control pads are required for each zener diode.
- Can be simulated to verify best trim code.

# Trim Simulation Using Zener Diodes




- Zener trim technology provides the ability to simulate zap effects by temporarily shorting individual zener diodes with test hardware relays.
- There are no simulation techniques available for Fuse technology.
- E/EEPROM technology can be simulated.

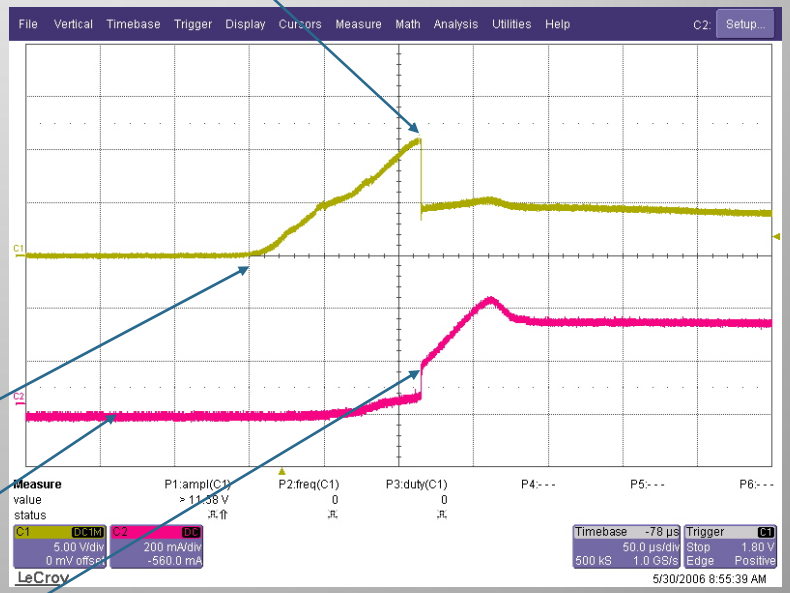


# Zapping Zeners

- Typical I/V Requirements:
  - ABCD150 - ~13 V, 300 mA
  - PVIP50 - ~5 V, 20 mA

**Zener breakdown ~11v in this case**

**Voltage in yellow, Current in red.**

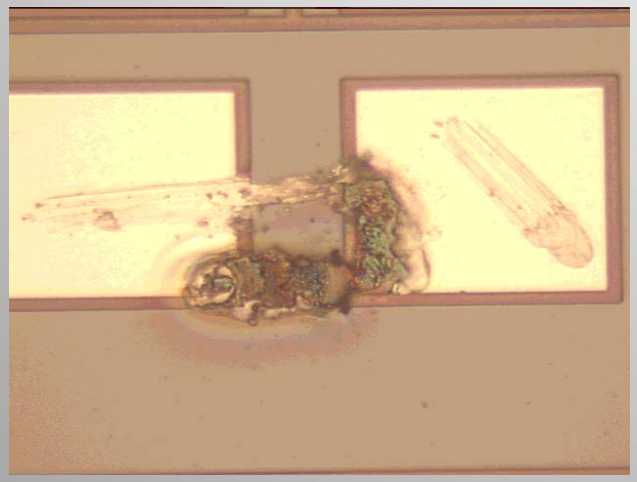



**Voltage rises until zener breakdown**

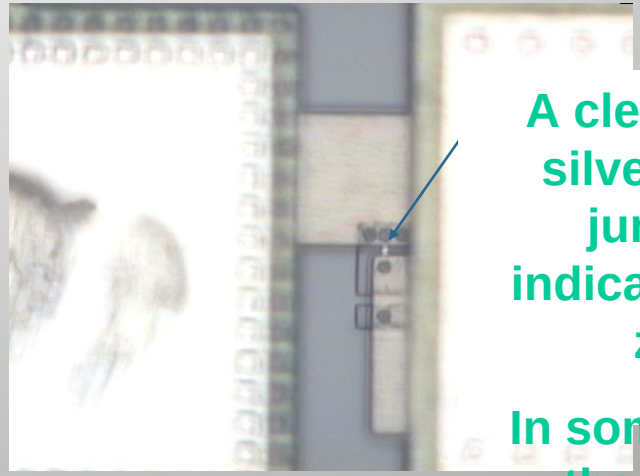
**Current probe had offset, absolute values are shifted (notice negative current before the zap)**

**At breakdown, current increases quickly**

# After Zener Zap



Bad zener zap,
excessive current



Good zener zap

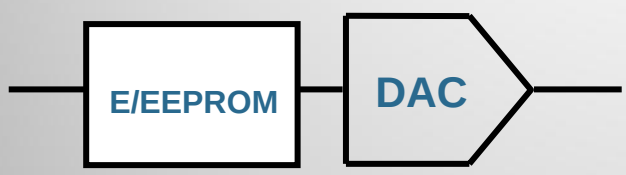
A clear bright silver line in junction indicates good zap.

In some cases, the flow is subterranean, and appears as a fuzzy line.

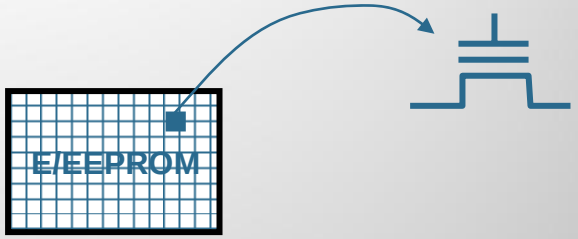
# Trimming Type

- Fuse
- Laser
- Zener
- E/EEPROM

# E/EEPROM Trimmable Reference Circuit



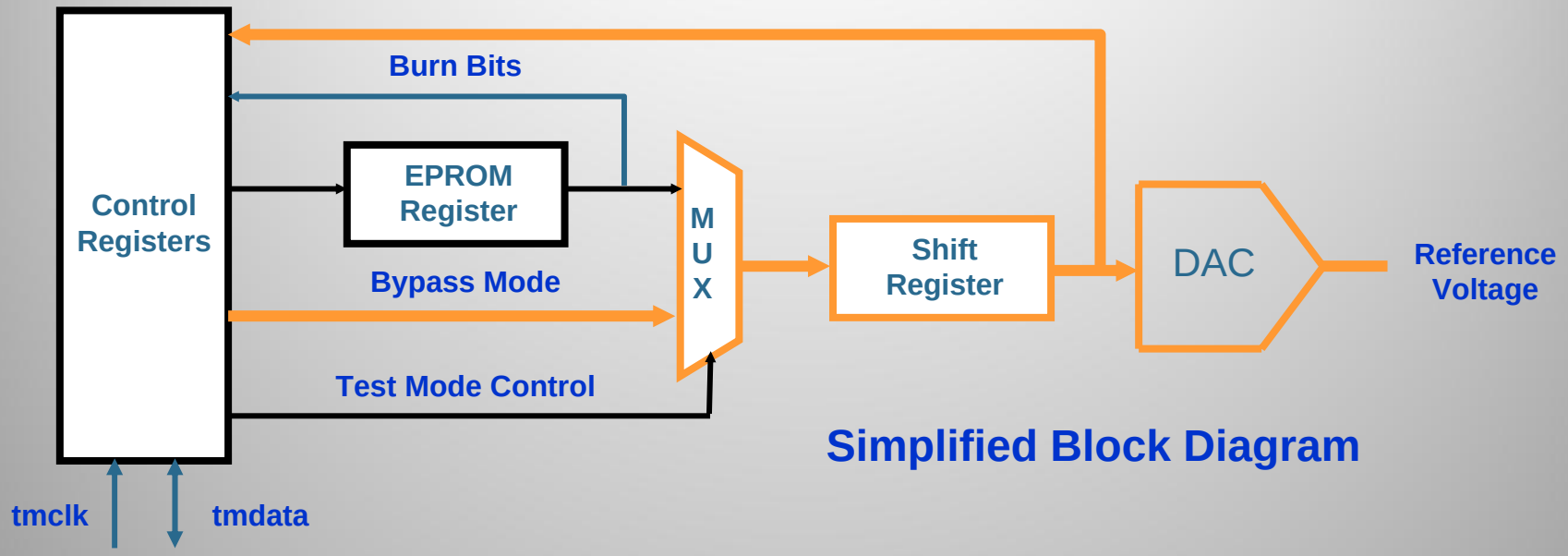

Trim Memory




Floating Gate MOS Technology

- E/EEPROM registers store digital values in memory that allow a DAC to adjust current or voltage values.
  - Except for the control pads, no probe pads are required.
- EPROMS (One-Time Electrical Programmable Read Only Memory) can be programmed but not erased.
  - They can be programmed one time only, so these are used after the code is bug free.
- An EEPROM (Electrically Erasable Programmable Read Only Memory) is similar to an EPROM but it can be erased.
  - Can be programmed many times, hence trim errors can be easily corrected.

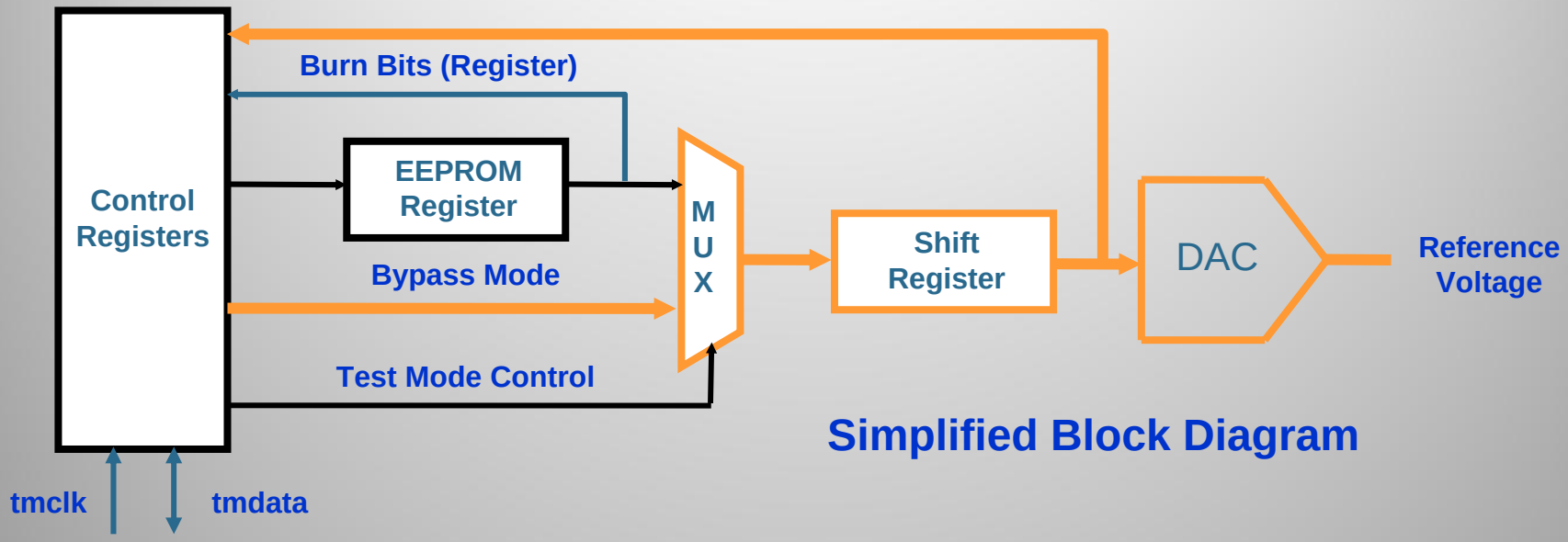
# EPROM Programming




Simplified Block Diagram

- Simulate and find the best-fit trim code.
- Program the EPROM register and verify result.
- Cycle power and read back programmed data. Verify that it is the correct data.
- Verify all electrical parameters.

# EEPROM Programming



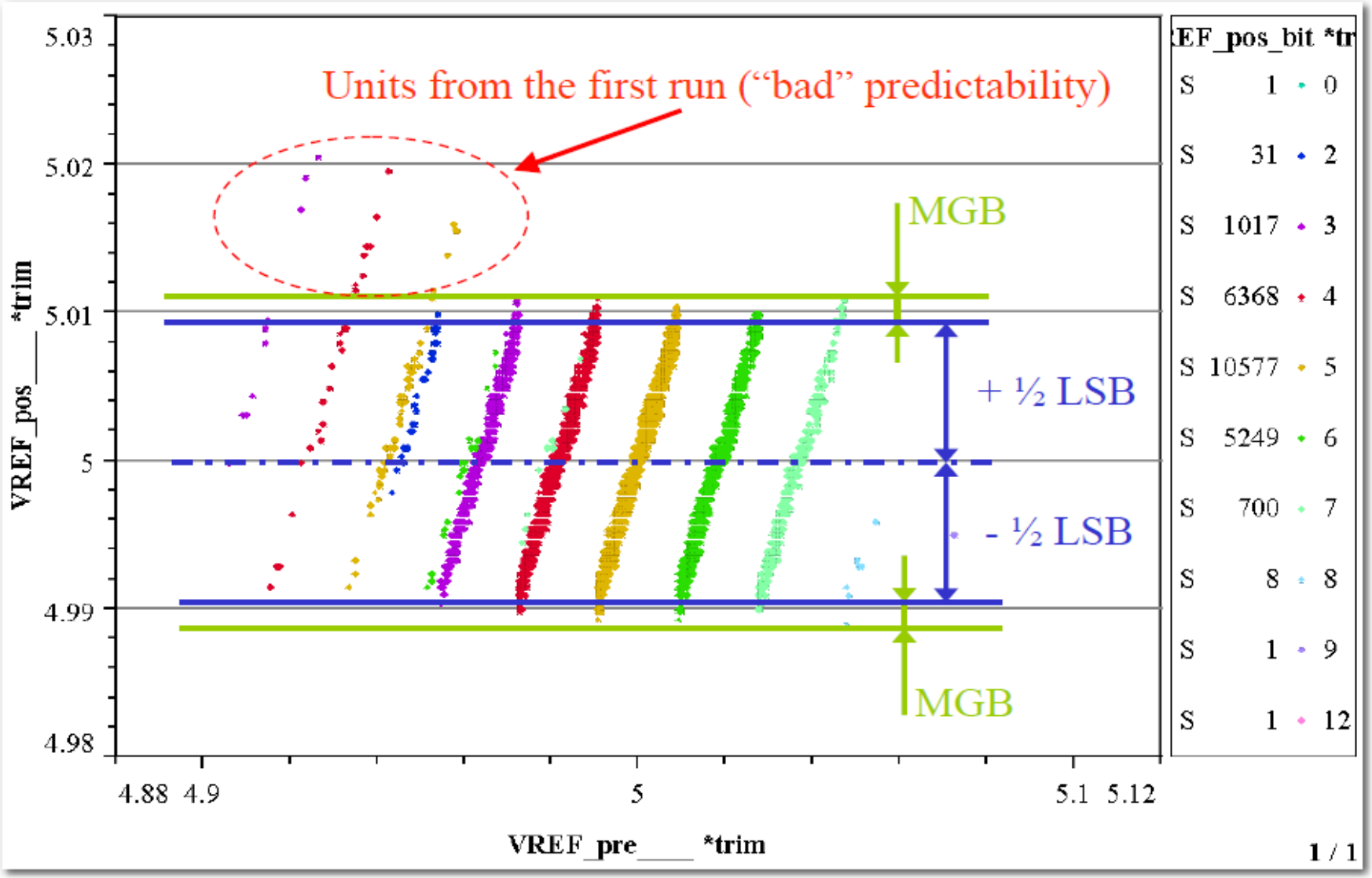

Simplified Block Diagram

- Simulate and find the best-fit trim code.
- Re-program the EEPROM register. Verify result.
- Cycle power and read back programmed data. Verify that the data is correct.
- Verify all electrical parameters.

# Trimming

- Trim parameter folding

# Trim parameter folding



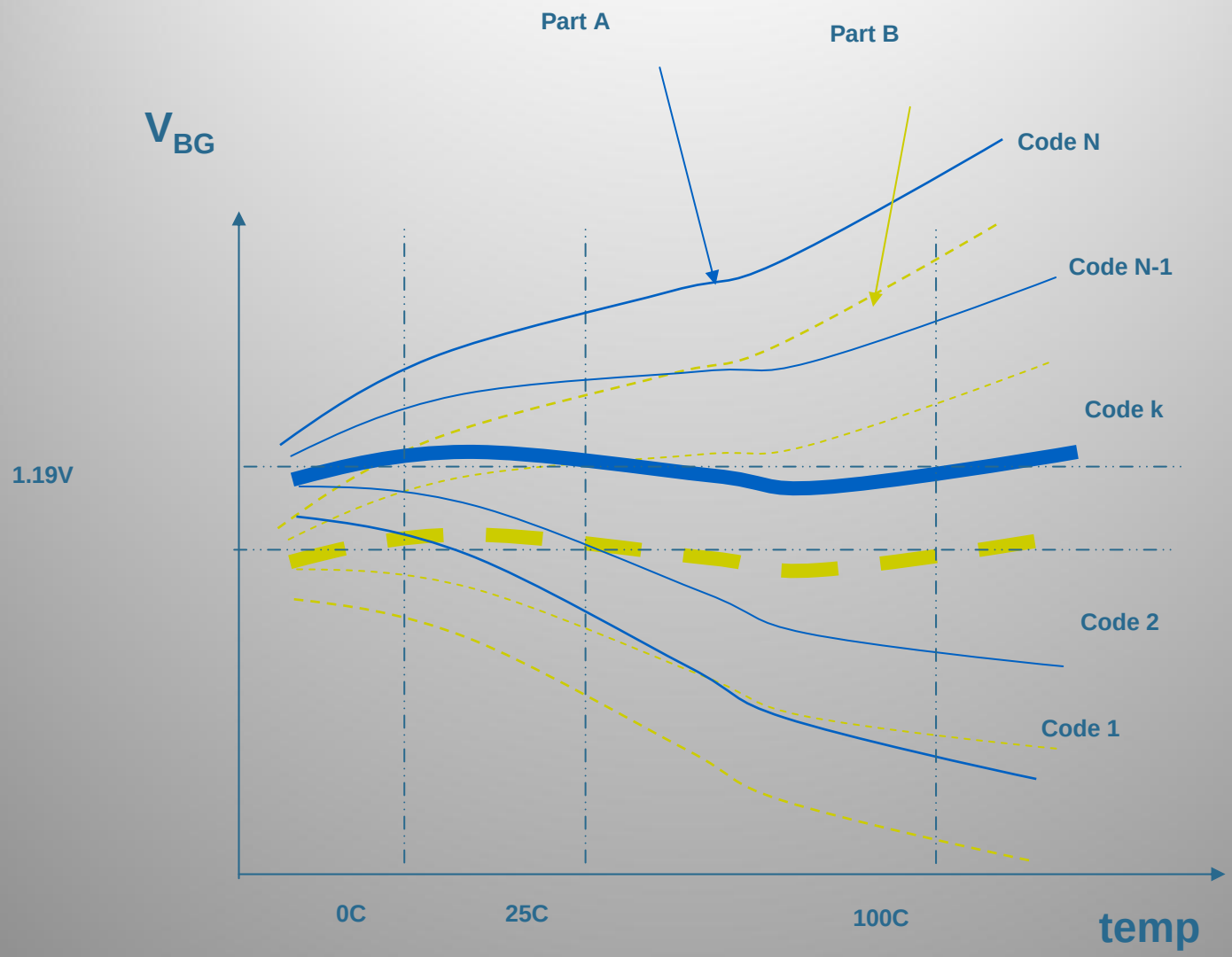

Note: The bad predictability can be avoided by doing the trim table char before the device trimming (execution of 'pre()').

As all the trim steps are measured before the trimming the class can use this values to trim accurately from the first unit on

# Trimming

- Trim parameter folding
- Reference Distributions

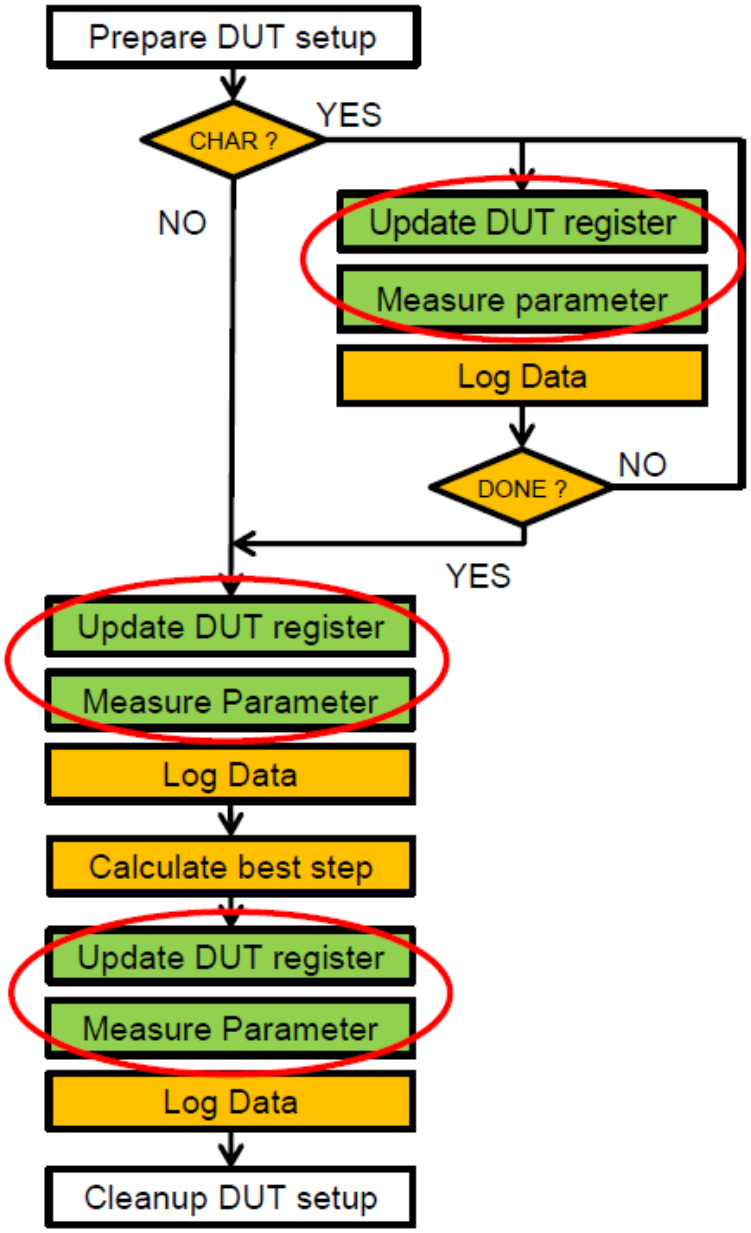
# Reference Distributions

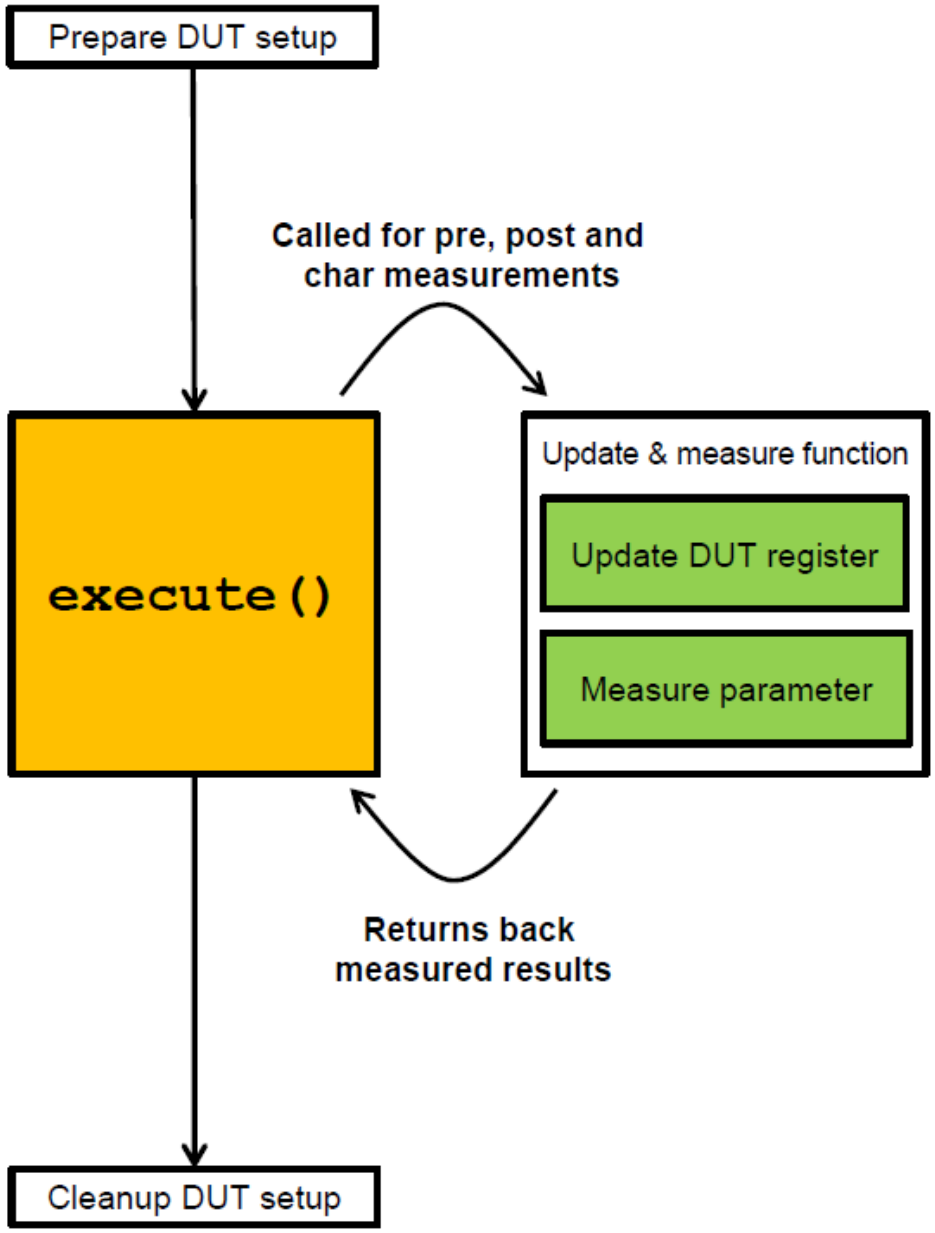


# Trimming

- Trim parameter folding
- Reference Distributions
- Trim Flow

# Trimming flow



# Execute() function

- Execute() …
  - Implements correct sequence of TREG calls (e.g. pre(), post(),…)
  - Ensures that trim learning works out of the box
  - Easies trim step characterization (for device char and during production)
  - Skips post trim measurement if not necessary (→ saves test time)
  - Takes care of datalogging
  - Test engineer can focus on implementing the measurement

```
// Test Function: iconst_trim
ETS_PRGFLOW_FUNC iconst_trim( int DSIndex, LPCTSTR TestLabel )
{
     // Prepare DUT Setup
     dut.sel("TM").set_working(TM3); // select testmode 3

     dut.trim("iconst").execute(measure_iconst_trim,1100,8200,TREG_LOG_STD);

     // Cleanup DUT Setup

     return( msSiteStat( MS_ALL ) );
} // END_ETS_PRGFLOW_FUNC
```

Update & measure function
First **production** datalog number
First **characterization** datalog number
Log level (optional)


Thank You !