TestConX™

Archive

DoubleTree by Hilton
Mesa, Arizona
March 3-6, 2024

Turbo Charging Active Laser Trim for Precision & Throughput Optimization

Joseph Yehle & Yazdi Contractor

Texas Instruments

Mesa, Arizona ● March 3–6, 2024

What problem are we trying to solve here?

- Active laser trimming delivers maximum precision with a trim methodology that consumes very little (or no) die area.
  - **Sign me up for that!**
- Optimizing the trim settings to optimize throughput (it is not fast) and precision is device specific and far from easy.
  - **Well we knew there had to be a catch somewhere.**

Can we make that optimization quicker/easier without sacrificing the precision?

Active Laser Trim Cliff Notes – 1 of 4

Laser Trimming

Off-Line Laser

- Trim & Test are separate manufacturing steps.
- Trimming needs to be very fast to be cost effective.
- Ex: Memory Repair
- Ex: Probe - Trim - ReProbe

On-Line Laser

- Trim & Test are combined into a single manufacturing step.
- Trimming needs to be well integrated with the tester.
- Ex: Precision resistors
- Ex: Probe with integrated trim

Active Laser Trim Cliff Notes – 2 of 4

On-Line Trimming

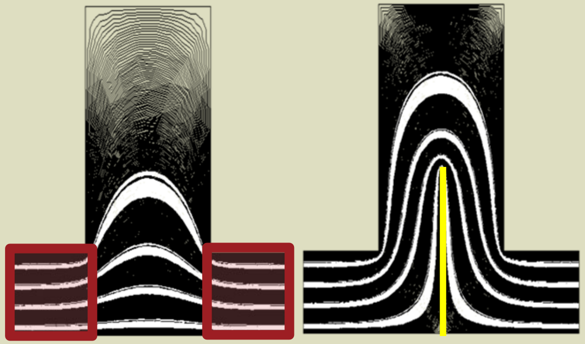
Link Trims

- Fuses or Links like Off-Line trimming
- Trim weight per fuse/link is not predictable enough to create the trim code map required by Off-Line.

Active Trims

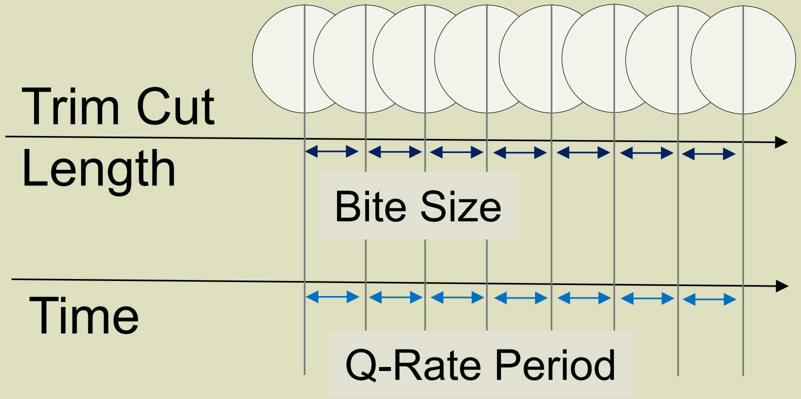
- Almost always a resistor.
- Trims have significant variation.
- Trim result is measured between every laser blast.
- Decision to execute/stop the next laser blast is made just before the next laser blast.

Active Laser Trim Cliff Notes – 3 of 4

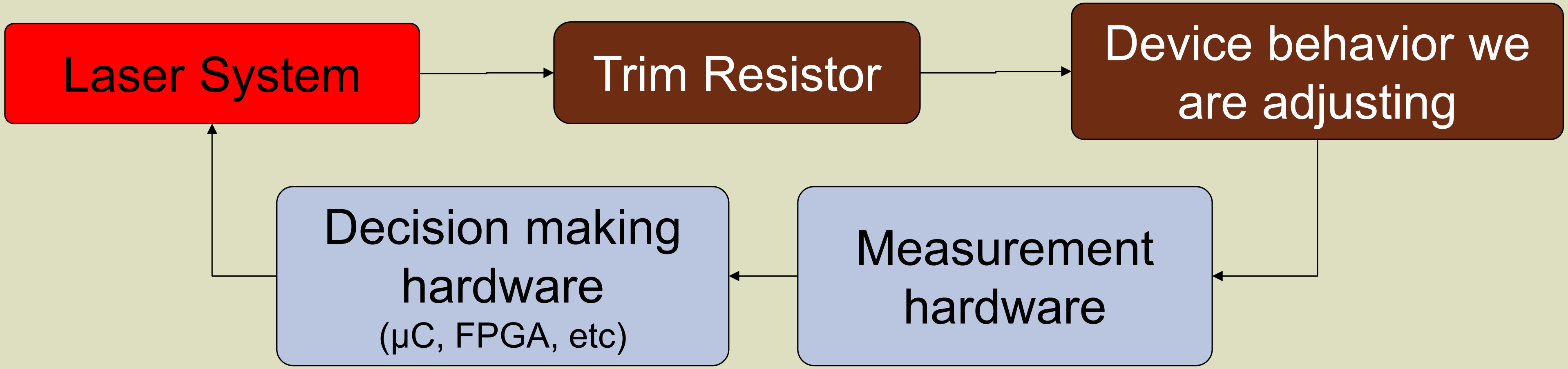


Bite Size:	Larger	– coarser trim, but faster
	Smaller	– finer trim, but slower
Q-Rate:	Faster	– less time for feedback loop
	Slower	– more time for loop to stabilize

Trim is achieved by making a series of overlapping laser blasts in a resistor to divert current flow into a longer path, increasing the effective resistance



Active Laser Trim Cliff Notes – 4 of 4

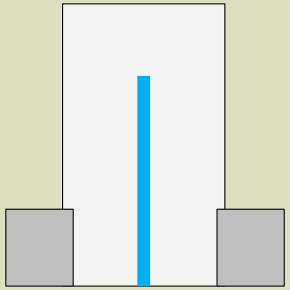


- We go around this loop for every laser blast of the trim.

Where can we optimize Bite Size & Qrate?

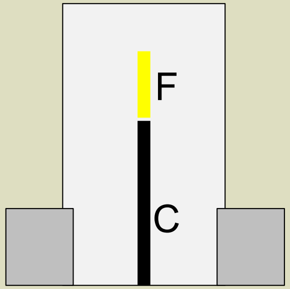
- Remember: Trim speed and precision are inversely proportional to each other.
  - Fine precision = smaller bite sizes = slower trims
  - Faster trims = larger bite sizes = worse precision
  - All of this is relative and the magnitude and slope of the relationships are device specific
- Bite Size: Optimization ability is limited.
  - Combination of datasheet parameters and IC design dictates the bite size at the end of the trim.
  - The test engineer has to determine a bite size that provides the required final precision.
- Q-Rate: Optimization ability is even more limited.
  - The test engineer will typically run this as fast possible, minus a little of bit of margin, that keeps the trim loop stable.

Optimizing Bite Size – How people do it today



No Optimization

- Trim the *entire* feature at the bite size required for the final precision
- Pros: Easy to setup for a beginner user. This is the simplest trim method available.
- Cons: Worst test time

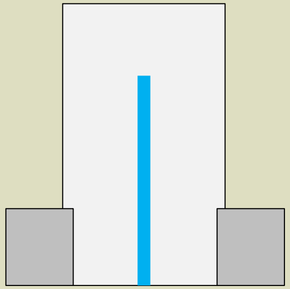


Coarse/Fine Optimization

- Split the feature into two legs.
- Coarse Leg: Larger bite size to save test time at the beginning of the trim.
- Fine Leg: Smaller bite size for the required precision at the end of the trim.
- Pros: Less test time.
- Cons: Much more complicated to set up even for an experienced user. Every device will be different. Adds overhead in the trimmer to execute two legs vs one.

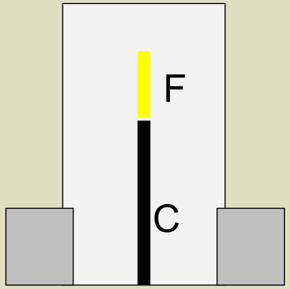
TestConX 2024

Optimizing Bite Size – Let us Turbocharge it!



No Optimization

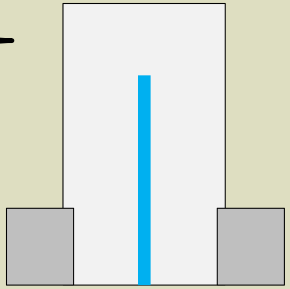
- Simple to create recipe
- Slow to execute in production



Coarse/Fine Optimization

- Pain to create recipe
- Faster to execute in production

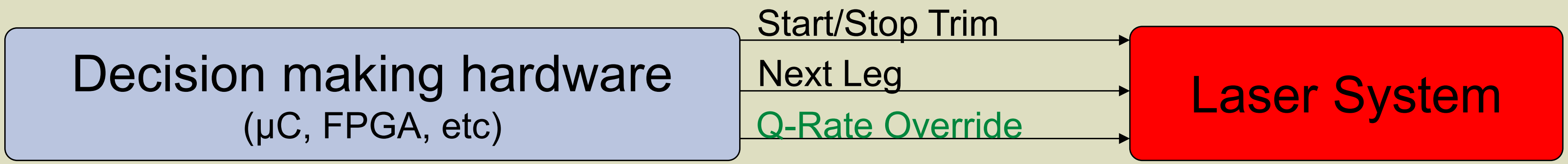
Add a new control signal *(Q-rate Overide)* to gain the benefits of the two trim methods while avoiding the costs.



Turbo Mode

- Patent Pending
- Same simple to create recipe
- *Applies a Q-Rate override* at the beginning of the trim
  – 10 kHz opposed to say 2 kHz.
- Uses the default Q-Rate and Bite Size at the end of the trim to achieve the required precision.

Control lines for Turbocharging the trim



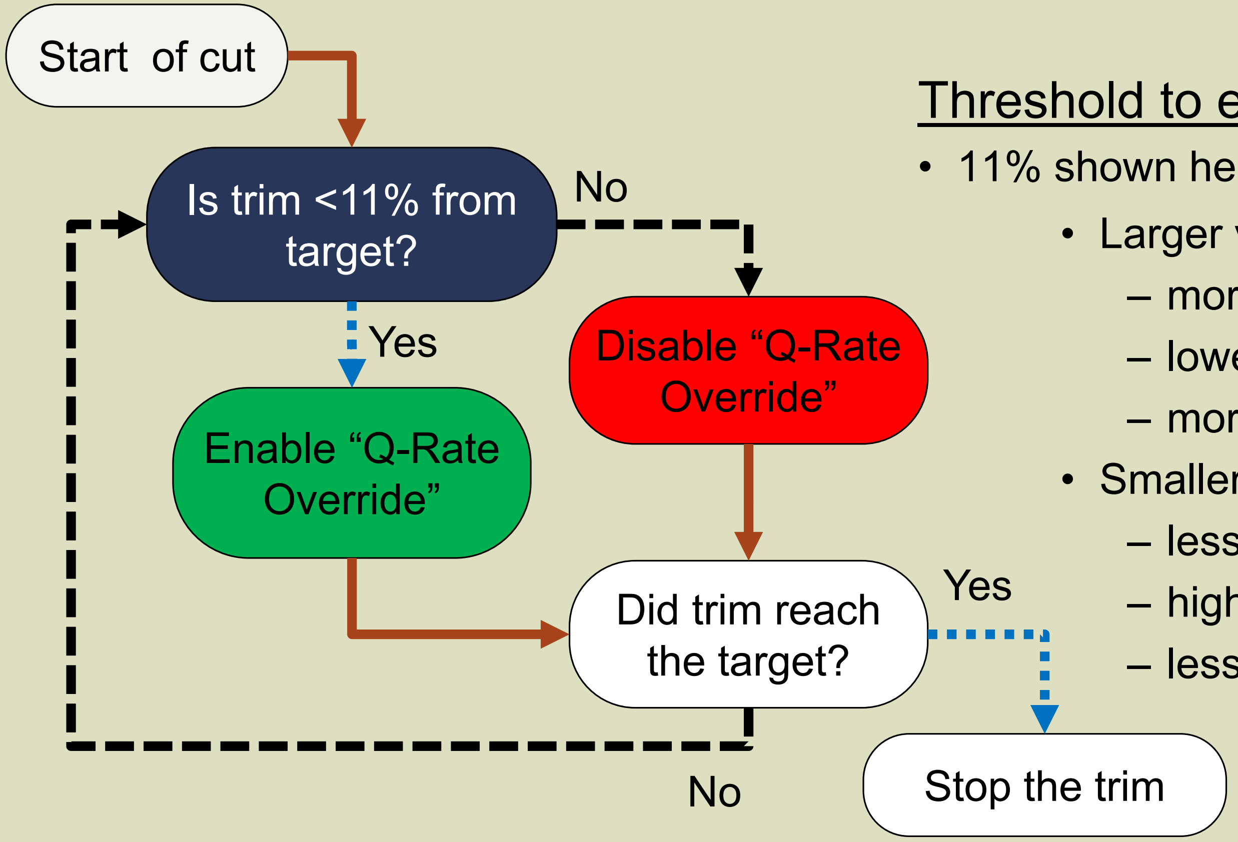
Standard trimmer control signals

- Stop Trim: Tells the trimmer to stop the cut when the target result has been achieved.
- Next Leg: Tells the trimmer to jump from the course to fine leg.

New trimmer control signal

- Q-Rate Override: Tells the trimmer to use a much faster Q-rate while the control is active.

Control block diagram for Turbocharging the trim



Threshold to enable/disable Q-Rate Override

- 11% shown here as an example.
  - Larger value
    - more of trim executes at recipe settings
    - lower risk of disabling too late & overshoot
    - more trim time overall
  - Smaller value
    - less of trim executes at recipe settings
    - higher risk of disabling too late & overshoot
    - less trim time

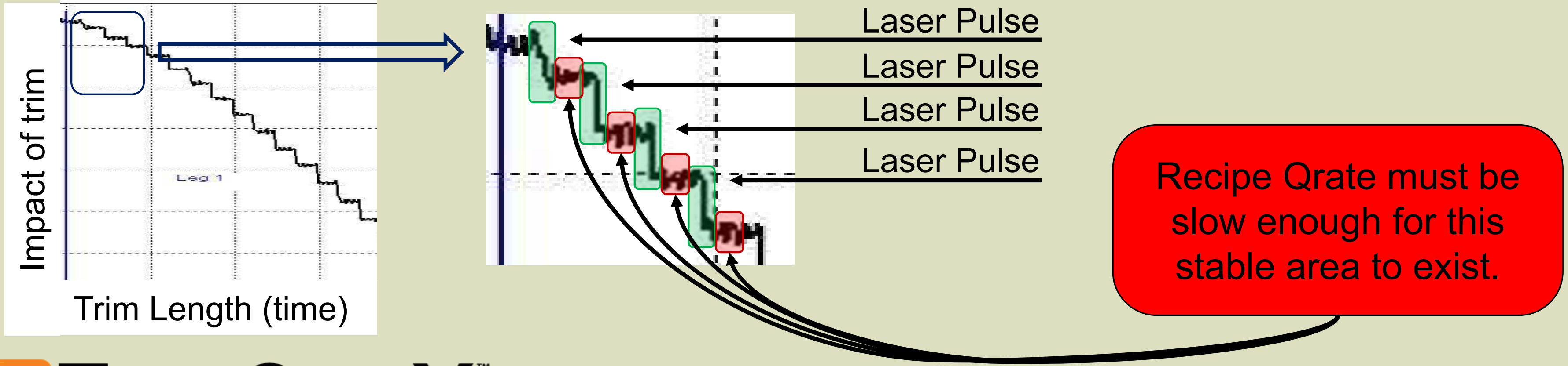
So why not just trim faster all the time?

- This is a lot of effort to dynamically use a faster Qrate for part of the trim.
  - New control input on the trimmer
  - New output on the tester integration (that measures the trim impact and controls the trimmer).
  - New firmware on the tester integration to determine when to enable/disable Turbocharging.
  - User training
- Wouldn't it just be simpler to use the maximum Qrate all the time in the recipe?

- In reality, everyone already uses the maximum Qrate all the time in the recipe.
  - that is stable
  - Remember, active laser trimming has a real-time control loop around the trimming operation.
  - The trim impact is measured between every laser pulse to know when the target is reached.
  - This loop (Trimmer + device + trim impact + measurement + decision making) can only run so fast.

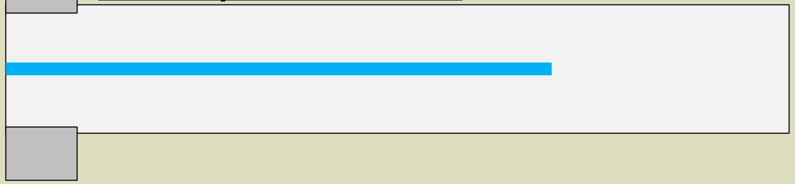
Trading trim stability for trim speed.

- Turbocharging uses a Q-Rate that is much faster than the recipe.
  - For example: Turbocharging at 10 kHz compared to normal trimming at 2 kHz.
- Turbocharging trims so fast that the real-time control loop is not accurate.
  - We stop <11% (again, an example) short of the final target because we are not accurately measuring the trim value while moving this fast.
  - We need some amount of trim range for the loop to stabilize to achieve a precise final result.



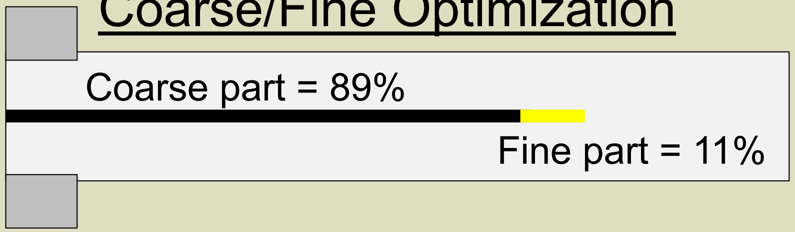
Results with Turbocharging the trim

No Optimization



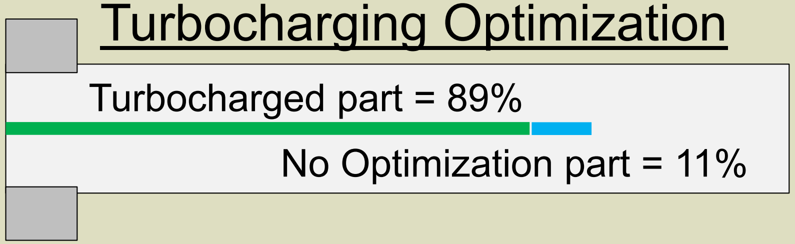
- Trim length: 225 microns
- Bite Size for final precision: 1 micron
- Qrate: 2 kHz
- Total trim time: 113 milliseconds

Coarse/Fine Optimization



- Coarse Bite Size: 2 microns (2x)
- Fine Bite Size: 1 micron (no change)
- Overhead to use a second cut: 2.5 milliseconds
- Total trim time: 65 milliseconds (-42%)
- **Quite tedious to optimize and unique for every product.**

Turbocharging Optimization



- All recipe settings: no changes from simple baseline version
- Overhead to use the TurboCharge 2.5 milliseconds
- TurboCharge Qrate: 10 kHz
- Total trim time: 35 milliseconds (-69%)
- Uses the simple recipe with a better control system.

Turbocharging an active laser trim - Conclusions

You can have your cake, with icing, and eat it too.

- “Cake” An active laser trim recipe that uses one bite-size per feature and one cut per feature. Easy to create. Will always work with stable yield and precision.
- “Icing” You can use that recipe for early customer samples *and* final production.
- “Eating it too” The trim speed can be significantly improved via control automation without a tedious optimization procedure for every device