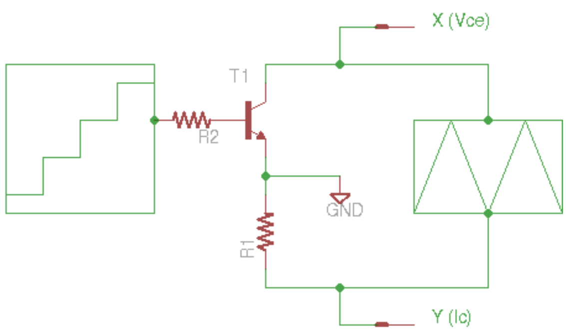
Yet Another Semiconductor V/I Curve Tracer – I

September 2, 2013 kwong

Semiconductor curve tracers were frequently used in analog circuit designs many decades ago, at a time when discrete semiconductor devices were in dominance and ICs were scarce. While curve tracers are no longer widely used nowadays due to the ubiquity of digital circuitry and computer aided designs, they are still quite popular in the educational world and among hobbyists. In this and the next post, I will discuss some of the design considerations of a curve tracer that I built and show some real-world measurement results.

A typical use of a curve tracer is to generate the V-I characteristics of the device under test (DUT). The following diagram illustrate how a typical configuration works when the DUT is a BJT.



Principle of Operation

In the figure above, a staircase waveform is fed into the base of a NPN transistor via a base resistor R2. For each step in the waveform, a different base current (Ib) is generated. When the input waveform voltage is sufficiently large, the voltage drop between the base and emitter junction (Vbe) can be ignored and thus:

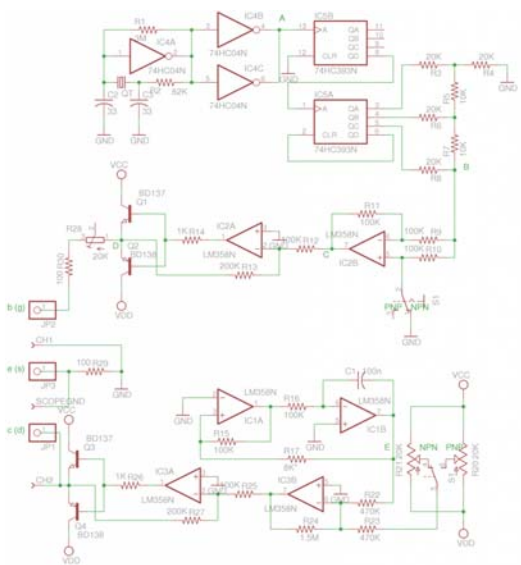
$$I_b\approx\frac{V_{bi}}{R_2}$$

where Vbi is the voltage of the i*th* staircase. This arrangement is usually good enough for most cases and this is what I used in my design as well. If we need more accurate characterizations of different Ib's, we can either use a variable current source (which increases the circuit complexity) in place of the staircase waveform generator or simply use the same design but add a current sensing OpAmp to read back the actual base current.

A sweep voltage is applied between the collector and the emitter. The sweep waveform is usually a symmetrical triangular wave so that characteristic curves for both sweep directions can be captured. When we only care about the characteristics in one direction (say when the voltage is ramping up) we could use a sawtooth waveform instead. The linearly of this swept waveform is not significantly important, and in many curve tracer implementations (like this one) the sweep waveform is simply bridge-power devices however, the common emitter gain (hfe) can be quite small (e.g. in the teens) and the collector current measurement can be off by quite a bit when the measurement is done on the emitter side.

The main reason we prefer the emitter side Ic measurement is because it is very easy to implement. When we use this implementation, we could use the emitter as a virtual ground and measure both Vce and the voltage drop across R1 using the two channels of an oscilloscope easily. Of course if your intention is to get the most accurate measurement in all circumstances at all cost, you could move the resistor to the collector side and use a differential amplifier to measure the current flowing through instead. There is another hurdle to the collector side current measurement implementation. Since the DUT can be either N-type or P-type, it will require the differential amplifier to have a common-mode input range that includes both rails. Bi-directional current-sensing OpAmps that have this characteristics tend to be very expensive.

After we covered the basics, let us take a look at my design below:

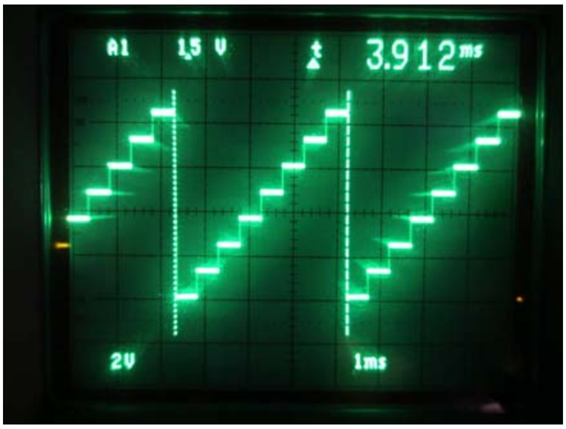


Parameter Tracer

Staircase Waveform Generator

IC4A (1/6 74HC04) along with the watch crystal (32768 Hz), R1, R2, C1 and C2 form a inverting amplifier is unity. IC2A also forms an inverting amplifier but with a gain of 2, so the overall output signal is either between 0 to 10V or 0 to -10V depending on the polarity selected. Finally the output from IC2A is used to drive a simple push-pull stage so that we can draw more current from the circuit than the OpAmp alone could provide. Note that the feedback loop of the output OpAmp (IC2A) is connected to the collectors of the push-pull stage and not the output pin (pin 1), this is important as the OpAmp will be able to compensate the Vbe drop in the push-pull stage and thus avoiding signal distortion.

Here is a picture showing the output staircase waveform:



Staircase Waveform

Triangular Waveform Generator

The staircase waveform generator mentioned above is used to generate the discrete Ib's in the V-I plot. Now let us take a look at the generation of the Vce sweep signal.

Triangular Waveform

The vertical sweep frequency does not need to be in sync with the staircase waveform. In my case, these two frequencies happened to be pretty close. In general, you want both waveform generators to operate in the hundreds Hertz range. Too low of a sweep frequency makes the display flicker and too high of a sweep frequency would require higher bandwidth and slew rate OpAmps in order to avoid waveform distortion.

Other Considerations

As mentioned earlier, my design philosophy was to keep everything simple and yet make the curve tracer fully functional. So some of the aforementioned sacrifices were deliberate design choices to make keep the curve tracer as simple as possible.

One thing you could easily improve upon this design is to use better spec'd OpAmps. I used LM258/LM358 because that's the OpAmp I happen to have on hand. They definitely get the job done, but there are some limitations that you should be aware of

‹ UPS Transfer Time Measurement

Yet Another Semiconductor V/I Curve Tracer – II ›

4 Thoughts on "Yet Another Semiconductor V/I Curve Tracer – I"

- November 5, 2013 at 1:59 pm

  Excellent, I was just thinking about building an octopus style tracer but this seems

much better.
A simplification would be to replace ic4 and ic5 with a single cd4060 or 74hc4060 which combines an oscillator suitable for direct use with your crystal with a 14 stage ripple counter in pretty much the exact configuration you want.

John Meacham

Reply

- November 5, 2013 at 9:18 pm

  Good to know. Thanks!

  Reply

  kwong

- March 17, 2014 at 10:33 am

  May i know what component is S1 under r10 and IC2b? and why pin 3 is not connected? Thank You.

  Jem

  Reply

  - March 17, 2014 at 7:55 pm

    S1 is a DPDT (double pole double throw) switch. The other part of the switch is next to R20.
    Pin 3 is float in PNP mode and IC2B is configured as a non-inverting amplifier.

    kwong

    Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

Post Comment

Notify me when new comments are added.

Categories

- AVR/Arduino (98)
- Coding (133)
- Electronics (224)
- Linux/BSD (34)
- Miscellaneous (190)
- MSP430G2 (10)
- PIC (4)
- Product Reviews (21)
- Raspberry Pi (5)
- Teardowns (59)
- Technology (57)

Tools & Libraries

- MCP2210 Library
- BCalc — A Hex/Bin/Dec Converter And Calculator

Most Popular Posts

- A Sensitive DIY Ultrasonic Range Sensor (286)
- Building a Constant Current/Constant Power Electronic Load (237)
- AVR LC Meter With Frequency Measurement (214)
- AD7705/AD7706 Library Revisited (91)
- A Digitally Controlled Dual Tracking Power Supply -- II (88)
- Hacking DTM0660L Based Multimeters (76)
- A Self-Balancing Robot – III (62)
- Reverse Engineering the Syma S107G IR Protocol (56)
- Arduino Development Using NetBeans (51)
- Dual Purpose Spot Welder With Pulse Duration Control (49)

Archives

Select Month

My Favorite Sites

- Arduino Forum
- Circuit Diagram
- Dangerous Prototypes
- EEVBlog
- Electrical Engineering Forum
- Electronics Lab
- Electroniq.net electronics projects and circuit diagrams
- Hack A Day
- Hacked Gadgets

Recent Comments

- PIEROGI on Visualizing a Full Bridge Rectifier

- PIEROGI on Visualizing a Full Bridge Rectifier
- kwong on Visualizing a Full Bridge Rectifier
- PIEROGI on Visualizing a Full Bridge Rectifier
- PIEROGI on Visualizing a Full Bridge Rectifier
- kwong on Teardown of a Hantek 2D72 3-in-1 Handheld Oscilloscope/DMM/AWG
- kwong on Visualizing a Full Bridge Rectifier
- David on Teardown of a Hantek 2D72 3-in-1 Handheld Oscilloscope/DMM/AWG
- PIEROGI on Visualizing a Full Bridge Rectifier
- kwong on Visualizing a Full Bridge Rectifier

Tags

74HC595 555 Accelerometer AD/DA Algorithm Arduino ASP.Net ATMega328 Atmega328P AVR C++ C Sharp (C#) DAC dasBlog Data Logging DS3232 Electronic Load FPGA GridView HP 8566B I2C Keithley 196 LCD Linux MCP2210 MSP-EXP430G2 MSP430 LaunchPad MSP430G2 MSP430G2231 Multi-threading Open Source Physical Computing Programmable Power Supply Q9450 Reflection Security SHT21 SPI Teardown Temperature Sensor Ubuntu Ubuntu Server VMWare Windows Vista WordPress