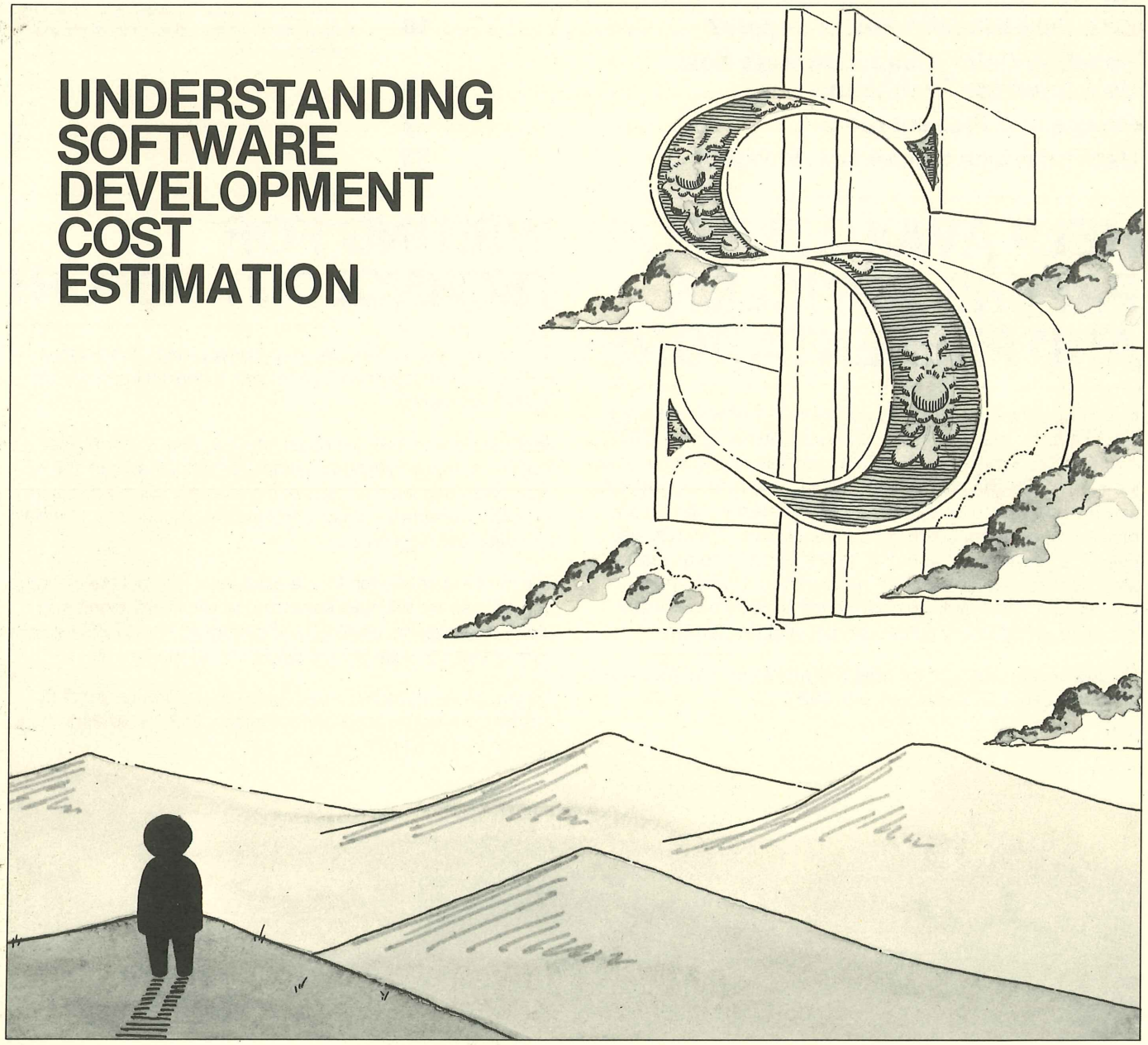
TEK INFORMATION FOR TECHNOLOGISTS

JUNE/JULY 1983

TECHNOLOGY report

COMPANY CONFIDENTIAL

UNDERSTANDING SOFTWARE DEVELOPMENT COST ESTIMATION



Tektronix®
COMMITTED TO EXCELLENCE

CONTENTS

Volume 5, No. 3, June/July 1983. Managing editor: Art Andersen, ext. MR-8934, d.s. 53-077. Cover: Joe Yoder; Graphic illustrator: Jackie Miner. Composition editor: Jean Bunker. Published for the benefit of the Tektronix engineering and scientific community.

Why TR?

Technology Report serves two purposes. Long-range, it promotes the flow of technical information among the diverse segments of the Tektronix engineering and scientific community. Short-range, it publicizes current events (new services available and notice of achievements by members of the technical community).

HELP AVAILABLE FOR PAPERS, ARTICLES, AND PRESENTATIONS

If you're preparing a paper for publication or presentation outside Tektronix, the Technology Communications Support (TCS) group of Corporate Marketing Communications can make your job easier. TCS can provide editorial help with outlines, abstracts, and manuscripts; prepare artwork for illustrations; and format material to journal or conference requirements. They can also help you "storyboard" your talk, and then produce professional, attractive slides to go with it. In addition, they interface with Patents and Trademarks to obtain confidentiality reviews and to assure all necessary patent and copyright protection.

For more information, or for whatever assistance you may need, contact Eleanor McElwee, ext. 642-8924. □

WRITING FOR *TECHNOLOGY REPORT*

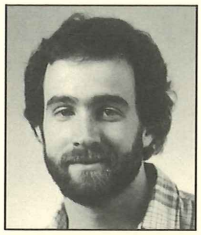
Technology Report can effectively convey ideas, innovations, services, and background information to the Tektronix technological community.

How long does it take to see an article appear in print? That is a function of many things (the completeness of the input, the review cycle, and the timeliness of the content). But the minimum is six weeks for simple announcements and as much as 14 weeks for major technical articles.

The most important step for the contributor is to put the message on paper so we will have something to work with. Don't worry about organization, spelling, and grammar. The editors will take care of that when we put the article into shape for you.

Do you have an article to contribute or an announcement to make? Contact the editor, Art Andersen, 642-8934 (Merlo Road) or write to d.s. 53-077. □

UNDERSTANDING SOFTWARE DEVELOPMENT COST ESTIMATION



Michael J. Demshki is a software engineer in the Software Center, part of the Technology Group. Michael joined Tektronix in 1982. He received the MSE degree in software engineering from the Wang Institute of Graduate Studies and the BS in business administration from Colorado State University. Michael was employed by the Hewlett-Packard Co. in Loveland, Colorado before graduate studies at the Wang Institute. He is co-author of WICOMO, a software cost estimation tool. Michael has presented seminars on software economics and project management to the Associao Portuguesa de Informatica in Lisbon, Portugal.

This article identifies the objectives of software cost estimation models, examines the fundamental assumptions upon which these models are based, discusses model limitations and use, presents two examples of models, and provides criteria for the reader to use in selecting a model.

Most software professionals have experienced the pressures of trying to meet an unrealistic development schedule. It is not uncommon in the software industry for development schedules to be underestimated by factors ranging from two for a "good" estimate to ten or more for a poor one.

After schedules are developed, it is often the software engineer who bears the burden of trying to meet an unrealistic schedule by working long hours, producing products that seldom satisfy personal or professional standards. During this process, motivation and job satisfaction suffer.

The effect of a poor estimate is not limited to those producing the software. It ripples through the entire organization. If the software is part of a larger product, then manufacturing, engineering, and support schedules will be missed and budgets will be overrun. For a stand-alone software product, the missed schedule means resources will not be available as planned for other projects. As the importance and demand for software increases, so does the need for accurate development estimates.

Poor estimates result from industry-wide inexperience – software engineering is still an emerging profession. Producing the next unit of software is not like producing the next widget. There are no well-defined units for measuring software project productivity and quality. Human factors compound the problem. Software development depends on human factors that are not understood well enough to use in predicting resource requirements.

Today's Estimating Techniques

The development schedule, while being very visible, is only one component of a complete estimate. A complete estimate usually states:

- Effort in units such as man-months, man-years, etc.
- Schedule in units such as weeks, months, or years
- Number of software professionals by life-cycle phase
- Costs in dollars

Estimated effort is the factor that must drive the estimate of the other components. The effort estimate, in turn, must be driven by the software product requirements and the development environment. There are many techniques used today to estimate development costs; some are subjective, others objective.

Subjective Estimating

Most estimates are produced by the project leader using informal, subjective techniques. The subjective technique is usually driven by two factors: *software product requirements* or *project constraints*.

Requirements-driven techniques examine a set of requirements, then, in the context of estimator's own experience, best-guess the time and personnel necessary. In the process of estimating, the estimator may consult others and incorporate their experience into the estimate to increase his confidence in its accuracy.

Although requirements-driven techniques have the advantage of combining past experience with present requirements, this advantage is offset by gaps in human recall. Almost invariably, project records inadequately fill these gaps. They simply don't yield accuate history associating resource requirements with actual usage. The second problem is that the estimating logic is seldom recorded. If ten people were to estimate the same project, you would get ten different estimates. Furthermore, nobody would be able to account for the discrepancies.

Other techniques are driven by project constraints, such as manpower, dollars, and time. With *constraint-driven techniques,* the estimator is given the software product requirements *and* the required completion date; these are absolutes, there is no estimating. The schedule has been fixed by factors independent of the job, by the competition, or by other schedules. In such situations, the estimator often lacks good, objective information to rebuff the dictated schedule; therefore, he accepts it, contrary to his better judgment.

Objective Estimating

Less commonly used are objective techniques such as quantitative software-development-cost modeling. These techniques predict resource requirements using algorithms and tables based on past projects. The advantage of quantitative models over the more subjective techniques is that identical inputs produce the same outputs every time. However, they have a disadvantage: They do not fully reflect the present project environment. Quantitative models heavily depend on historical data and cannot easily incorporate new factors.

No software cost estimating technique is perfect; each has its strengths and weaknesses (and some don't estimate at all). The best estimating technique, then, is to combine subjective and objective techniques that complement one another. Most people are familiar with the subjective techniques, but software-development-cost models have only recently become viable alternatives.

Quantitative Software-Development-Cost Models

A quantitative model is an abstraction of a process used to explain its behavior. To make the process more easily understood, models make simplifying assumptions about the process and environment, thus removing less important factors and emphasizing those of interest.

Most software cost estimation models in use today were developed as early as the late 60s for use by those for whom accurate estimating was vital: giant software contractors and hardware vendors, many in defense work. For these companies, understanding the elements that determine software cost was crucial to their success.

More recently, software costs are an increasing percentage of total system cost. This trend and the increased number of software solutions have spurred software cost estimation into the foreground. The Tek 1240 Logic Analyzer typifies the increased incorporation of software into new products. Due to these trends, cost estimation models are popular products for consulting firms. For example, those who have provided development methodologies in the past, such as Yourdon, now provide an accompanying cost estimation capability.

With the proliferation of software-cost-estimation models and the importance of controlling software development costs, software professionals will soon be expected to select and use estimation models knowledgeably. It is important to examine the fundamental assumptions on which these models are based and to know the objectives and limitations of the considered model. We will look at these factors and at two examples of models next. We will also look at criteria that I believe to be useful in selecting a model.

Objectives of Software-Development-Cost Models

Software-development-cost models have two primary objectives. The first is to predict resource requirements. Models provide an objective basis for economic analyses such as return-on-investment, rebuff over-optimistic schedules imposed by the use of constraint-driven techniques, and permit meaningful project tracking and control.

The second objective of software-development-cost models is to provide insights into the software development process. With process insights, we should be able to identify problems earlier and correct them, and analyze the effects of development technologies on the development process.

Model Fundamentals

This section will focus on factors common to the best-known software models. These models are:

The 1965 SDC Model [Nelson, 1966]

The TRW Wolverton Model [Wolverton, 1974]

The Putnam SLIM Model [Putnam, 1978]

The Doty Model [Herd, 1977]

The RCA Price S Model [Freiman-Park, 1979]

The IBM-FSD Model [Walston-Felix, 1977]

The 1977 Boeing Model [Black, 1977]

The 1979 GRC Model [Carriere-Thibodeau, 1979]

The Bailey-Basili Meta-Model [Bailey-Basili, 1981]

The Boehm COCOMO Model [Boehm, 1981]

The Jensen JS-1 Model [Jensen-Fletcher, 1983]

A brief overview of these models (excluding JS-1) is presented in [Boehm, 1981]. (Refer to the references for detailed descriptions.)

Quantitative models produce a software cost estimate by employing variables that are considered to predict development effort. Development effort is usually measured in man-months, man-years, etc. The development cost of a system is arrived at by assigning a dollar amount to each unit of effort, say man-months (MM), and taking the product, i.e., Estimated Cost = dollars per MM × Estimated MM.

The models imply that product size is the primary indicator of effort needed to complete a software project. The models differ somewhat as to which measurement of size most closely correlates with required effort, for example, source instructions, object instructions, number of routines. Most models, however, use *delivered source instruction* as their measure of product size [Boehm, 1981]. This conclusion is strongly supported by studies of hundreds of software projects. These studies find a strong correlation between the number of lines of source instructions and development effort [Walston-Felix, 1977], [Nelson, 1978], [Boehm, 1981].

The concept that men and months are not interchangeable is also common in the models. This follows the finding of [Brooks, 1975]. While most models offer equations relating effort to development time, no model offers a good explanation for the relationship.

Using Models

Quantitative models were designed for, and work best with, projects that follow the standard development life-cycle phases: concept, specification, design, implementation, and evaluation and verification (as defined in [Software Center, 1981]). Projects that follow other approaches, such as rapid prototyping, require adaption of the model to the approach.

The models reviewed by the Software Center were recommended by the models' authors for projects ranging from 2,000 to 500,000 delivered source instructions. For projects smaller than 2,000 source instructions, the models' estimates provide some guidelines, but the effort and schedule will be largely determined by the skills of the project leader.

Estimate accuracy depends on the inputs to the model, and the underlying model itself. Within the standard life-cycle, estimate, accuracy increases from phase to phase, as the software product is more precisely defined (see figure 1).

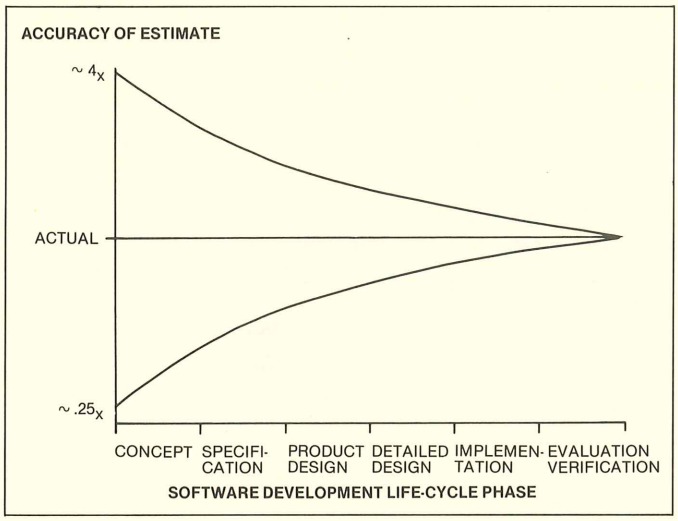


Figure 1. Estimating accuracy increases with knowledge.

In early life-cycle phases, when the product is understood only at the concept level, cost models provide the project manager with rough, order-of-magnitude estimates. These estimates are valuable for determining the feasibility of a project and for analyzing high-level trade-offs between cost, schedule, and system performance. But estimates to be used to schedule projects should be produced after testable requirements have been specified. Only then is enough known about the product to estimate accurately enough to schedule. In later phases, as still more details of the product are developed, accurate cost-to-complete estimates can be made.

Data based on actual usage of cost estimation models is not sufficient to draw sound conclusions about the accuracy of estimates produced during the design phase. But [Boehm, 1981] states that current models should produce estimates accurate to within 20 percent of actuals, 70 percent of the time (assuming accurate inputs). Promotional literature for some models claim accuracies around 95 percent, but often fail to state how consistently the model performs. Collecting model-accuracy data is a slow process; even studies of small projects take 9 to 12 months.

Before using any model, the estimator should thoroughly understand the definitions and assumptions behind the model. None of the models reviewed by the Software Center were sufficiently simple so that a person could just sit down and begin estimating without some study or training. Study and self-training, in turn, can be difficult as the material describing the models differs widely in clarity and completeness.

Regardless of which models you use, some basic steps will improve the accuracy of an estimate:

1. Define the objective for the estimate by asking:
   - What is the purpose of this estimate?
   - How accurate must it be?
   - How detailed and complete is the information about the software product being developed?
   - What information that could affect the estimate is not available?
   - Is the estimator's level of knowledge reasonable and accurate for the purposes?

   When you can answer "yes" to the last question, continue.

2. Estimate using at least two complementary techniques. First estimate using a familiar technique. Document the method and results. Next, estimate using a cost model and again document the method and results.

3. If the difference between the estimates is unacceptable, identify the reason for the discrepancy and re-estimate. Repeat step 2 until a single estimate can be derived based on the two techniques.

4. Track the progress of the project against the estimate, identifying and analyzing variances as they occur. Re-estimate cost-to-complete as more information becomes available.

By following this procedure and carefully documenting the results, project leaders will soon develop a base of information. With this information, estimating techniques and the software development process can be improved.

Examples of Models

Software-development-cost-estimation models can be classified as *empirically derived* or as *theoretically derived.* Empirically derived models employ historical data and formulas to predict future projects. Theoretically derived models base their estimates on assumptions about human behavior and project phenomena.

The COCOMO Model

The COCOMO model, developed by Boehm, is empirical. The model starts with a baseline effort equation of the form:

$$\text{EFFORT} = a \times \text{SIZE}^b$$

where SIZE is measured in thousands of delivered source instructions and *a* and *b* are constants determined by regression analysis of historical data. (The COCOMO model supplies default values for *a* and *b* derived from a database from 63 projects.) EFFORT is expressed in man-months. The baseline effort value is then modified using 15 multipliers that model the effect of environmental factors (cost drivers in figure 2). Each multiplier is assigned a value by using a rating system. The rating system is as objective as possible, using quantitative measures for most multipliers. Figure 3 gives an example of "programming language experience" (LEXP) ratings and effort multipliers. These are intended to help the estimator intuitively understand why each multiplier might significantly affect the project effort.

COCOMO Cost Drivers

- Product Attributes
  - Required Software Reliability
  - Data Base Size
  - Product Complexity
- Computer Attributes
  - Execution Time Constraint
  - Main Storage Constraint
  - Virtual Machine Volatility
  - Computer Turnaround Time
- Personnel Attributes
  - Analyst Capability
  - Applications Experience
  - Programmer Capability
  - Virtual Machine Experience
  - Programming Language Experience
- Project Attributes
  - Modern Programming Practices
  - Use of Software Tools
  - Required Development Schedule

Figure 2. The 15 environmental factors considered in modifying the COCOMO baseline effort value. Each factor is categorized as product, computer, personnel, or project attribute.

Once effort is estimated using the number of source instructions and the effort multipliers, the development schedule can be determined. The COCOMO model calculates total development time (DEV.TIME) as a function of the modified effort value (EFFORT.MOD) using the following equation:

$$DEV.TIME = c \times EFFORT.MOD^{d}$$

where c and d are empirically derived constants.

As a last step, time and effort are allocated to each development phase using tables derived from historical data and expert judgment.

The SLIM Model

Theoretically derived models are exemplified by Putnam's SLIM model, described by several papers in [Putnam, 1980]. The SLIM model is based on early work by Rayleigh and Norden. This work describes the pattern of manpower buildup and phase-out in human problem-solving efforts (see the Rayleigh curve in figure 4).

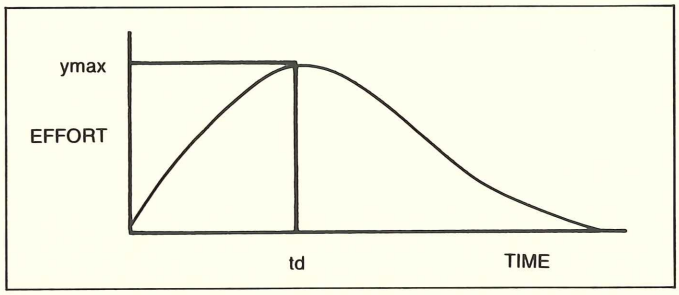


Figure 4. The Rayleigh curve. Putnam's work describing the distribution of project personnel (effort) over the software product life-cycle (time) is based on this curve.

Putnam observed that the software development process is a problem-solving effort and developed a Rayleigh-type curve of project personnel versus time to describe the software life-cycle. The SLIM model uses the following effort equation based on this curve:

$$S_s = C_k K^{1/3} t_d^{4/3}$$

where S_s is the size of the software product in source statements; K is the total life-cycle effort in units relative to t_d; t_d is the development time stated in equal units; C_k is a "technology factor" calculated from data on past projects or estimated based on factors of the development environment.

Putnam used empirical data to show that development effort is typically 40 percent of the total life-cycle effort, or 0.4K.

RATINGS		EFFORT MULTIPLIERS				
LEXP Rating	Phase	Requirements and Product Design	Detailed Design	Code and Unit Test	Integration and Test	Overall
Very Low:	⩽ 1 month average experience	1.02	1.10	1.20	1.20	1.14
Low:	4 months average experience	1.00	1.05	1.10	1.10	1.07
Nominal:	1 year average experience	1.00	1.00	1.00	1.00	1.00
High:	⩾ 3 years average experience	1.00	0.98	0.92	0.92	0.95

Figure 3. This example of programming language experience ratings and multipliers is for COCOMO (LEXP). Such ratings and multipliers are meant to coincide with an estimator's experience; that is, they should be intuitively perceived as "correct" by the estimator.

Putnam's curve is used by the SLIM model to determine first the minimum time required for the project, based on size and environment parameters. Then, cost and schedule trade-offs extending beyond the minimum period can be investigated to determine their optimal combination. The SLIM model also incorporates a software sizing technique adapted from PERT.

The use of empirical models such as COCOMO identify environmental factors that influence the software development process. Such models allow us to study the relationships of the environment on the process. Theoretical models such as SLIM provide insight into the underlying behavior of the software development process and allow the estimator to test his basic assumptions.

Selecting a Model

There is little agreement within the industry as to which software-cost-estimation model is best. Therefore, project leaders are on their own when evaluating and selecting the most appropriate model. To aid in this process, the Software Center has identified four areas for model evaluation: *definition, fit to the environment, contribution,* and *usability.*

Definition requires precise and complete information. This information should:

- Define terms, parameters, algorithms
- Describe model assumptions
- Define phases included in the estimate
- Define activities estimated in each phase
- Identify items not included in the estimate

An estimate is not of value for prediction if it cannot be determined whether or not the estimate included activities such as project management or documentation, or whether the modeling started in the specification or in the programming phase. Also, models that use "secret" or "proprietary" algorithms or that require extensive training to comprehend the documentation provide little insight into the development process.

Once the model is well-defined, its *fit to the environment* can be determined. A model designed specifically for use on large aerospace projects may not work well at Tek. Here, we are concerned with how well a model explains our development process and environment. When considering a model, look for these "fit" factors:

- Intended project profile
- Ease of model adaptation and calibration to our environment
- Input can be sufficiently detailed to reflect factors affecting the project
- Availability of the data required by the model

For example, a model will not fit the environment if we define systems and requirements at the module level, but the model's inputs and outputs are at the system level. Likewise, if a model requires historical data that is not available or data not present in our environment, it does not fit.

Contribution refers to the relative importance of the model in project planning and in understanding the development process. Will the model do these things:

- Provide consistently accurate estimates when compared to actuals?
- Allow you to intuitively understand *why* the model gave a particular estimate?
- Maintain relative stability of estimates despite small changes in inputs?
- Reasonably balance the impacts of objective and subjective factors on the estimate?

An example of a model with low contribution would be one that does not help the estimator intuitively resolve a discrepancy between the model's estimate and the estimate from another technique – or between the model's estimate and actual data.

Finally, a model that meets all of the above criteria must be *usable.* It must:

- Be readily available to the project manager
- Be automated
- Have a user interface that allows easy input and data modification
- Document the estimate
- Provide estimates and analyses that are not too terse nor cluttered and confusing. (Model use should aid the project leader, not create obstacles, extra work, or confusion.)

Conclusions

Greater use of software-development-cost-models can benefit individual projects and Tektronix as a whole. By using models in estimating development costs, project leaders and managers will:

- Better understand the development process
- Create realistic schedules to work and track against
- Control project costs
- Have quantitative bases for reviewing projects and transfering lessons learned to future projects.

For Tektronix, model use would:

- Create opportunities for a quantitative project data base
- Allow us to observe and analyze the effects of changes to the software development environment
- Help us manage our resources

Which model is most appropriate for Tektronix? The field is too new to select one model as being the most appropriate. There is plenty of room to use a variety of models without serious conflict. The important thing is to experiment with using models and to document these experiences well. The greatest value of models today is to increase our understanding of the estimating and development processes. In the end, only the project manager can accept responsibility for producing the final estimate.

Getting More Information

Tektronix has limited experience with both the SLIM model and the COCOMO model. For further information on experiences using SLIM, contact George Tice in MDP Language Software Evaluation 629-1310.

The COCOMO model is available within the Technology Group on the CRD Vax/Unix computer as a software product called the WICOMO tool. WICOMO was developed at the Wang Institute. The WICOMO User's Manual, extra copies of [Boehm, 1981], and training are available from the Software Center by contacting Michael Demshki at 627-3068, d.s. 50-487. □

References

[Bailey-Basili, 1981]. J.W. Bailey and V.R. Basili, "A Meta-Model for Software Development Resource Expenditures," *Proceedings, Fifth International Conference on Software Engineering,* IEEE/ACM/NBS, March 1981, pp. 107-116.

[Basili, 1980]. V.R. Basili, *Tutorial on Models and Metrics for Software Management and Engineering,* IEEE Computer Society, Catalog No. EHO-167-7, October 1980.

[Black, 1977]. R.K.D. Black, R.P. Curnow, R. Katz, and M.D. Gray, *BCS Software Production Data,* Final Technical Report, RADC-TR-77-116, Boeing Computer Services, Inc., March 1977, NTIS No. AD-A039852.

[Boehm, 1981]. B.W. Boehm, *Software Engineering Economics,* Prentice Hall, Inc., Englewood Cliffs, NJ, 1981.

[Brooks, 1975]. F.P. Brooks, Jr., *The Mythical Man-Month,* Addison-Wesley, Reading, MA, 1975.

[Carriere-Thibodeau, 1979]. W.M. Carriere and R. Thibodeau, *Development of a Logistics Software Cost Estimation Technique for Foreign Military Sales,* Report CR-3-839, General Research Corp., June 1979.

[Freiman-Park, 1979]. F.R. Freiman and R.E. Park, "PRICE Software Model-Version 3: An Overview," *Proceedings, IEEE-PINY Workshop on Quantitative Software Models,* IEEE Catalog No. TH0067-9, October, 1979, pp. 32-41.

[Herd, 1977]. J.R. Herd, J.N. Postak, W.E. Russell, and K.R. Stewart, *Software Cost Estimation Study-Study Results,* Final Technical Report, RADC-TR-77-220, Vol. I (of two), Doty Associates, Inc., Rockville, MD, June 1977.

[Jensen-Fletcher, 1983]. R. Jansen and B. Fletcher, *Software Development Cost and Schedule Estimating with the JS-I System,* Computer Economics, Inc., Marina del Rey, CA, 1983.

[Nelson, 1966]. E.A. Nelson, *Management Handbook for Estimation of Computer Programming Costs,* AD-A648750, Systems Development Corp., October 31, 1966.

[Nelson, 1978]. R. Nelson, *Software Data Collection and Analysis at RADC,* Rome Air Development Center, Rome, NY, 1978.

[Putnam, 1980]. L.H. Putnam, *Tutorial on Software Cost Estimating and Life-Cycle Control: Getting the Software Numbers,* IEEE Computer Society, Catalog No. EHO 165-1, October 1980.

[Putnam, 1980]. L.H. Putnam, "A General Empirical Solution to the Macro Software Sizing and Estimating Problem," *IEEE Trans. Software Engr.,* July 1978, pp. 345-361.

[Software Center, 1981]. *Software Project Management Guidelines and Project Review Checklist,* Software Center, Tektronix, Inc., Beaverton, OR, September 1981.

[Walston-Felix, 1977]. C.E. Walston and C.P. Felix, "A Method of Programming Measurement and Estimation," *IBM Systems Journal,* Vol. 16, No. 1, 1977, pp. 54-73.

[Wolverton, 1974]. R.W. Wolverton, "The Cost of Developing Large-Scale Software," *IEEE Trans., Computers,* June 1974, pp. 615-636.

HIGH-SPEED MONOLITHIC HORIZONTAL AMPLIFIER YIELDS HIGHER PERFORMANCE, LOWER COST

Art Metz is a senior engineer in the Transient Mainframe group, part of the Instrument Systems Division. He joined Tektronix in 1968 from the Argonne National Laboratory. Art has designed parts of the 7D14, 7D15, 7104, 7B10, 7B15, and the 7104 calibration fixture. He received his AAS from the DeVry Technical Institute.

Market pressures demand oscilloscopes with faster sweep speeds, greater accuracy and lower cost. The monolithic horizontal amplifier (M215) for the new Tektronix 2400 series portable oscilloscopes was developed to address these competing issues. The special performance of this amplifier required both innovative circuit techniques and a new, high-frequency, high-voltage integrated circuit process. To take full advantage of the new IC process, the SPICE computer simulation program was modified to accommodate the characteristics peculiar to high-voltage transistors.

Many Tektronix engineers contributed to the success of the M215 project. Among these were Carl Battjes for pioneering work in high slew-rate amplifiers, Ken Schlotzhauer for developing SCALX (a computer program to generate transistor models for SPICE simulation) and Binoy Rosario who developed the SPICE B simulation program. Ralph Ulrich and Richard Wood developed the SHHV process. Richard Wright of the Solid State Group developed the DIP package for the M215 amplifier.

Monolithic Solution to Horizontal Requirements

An oscilloscope horizontal amplifier is built of blocks of circuitry having divergent requirements, some of which are common to most amplifier designs. The horizontal amplifier of the 2400 series consists of a four-input multiplexer; two transconductance amplifiers (one a 10X magnification amplifier); a variable gain stage; and a differential, high-slew rate, output amplifier.

In a horizontal amplifier, the large output-voltage swing required to drive the CRT mandates the use of high-voltage transistors. Normally, high-voltage transistors have a lower current/gain-bandwidth product (F_T) and lower current density than do low-voltage devices. Tektronix had previously developed a high-voltage integrated circuit process that offered high-voltage devices that were faster than discretes. The capability of this process indicated that a monolithic design could potentially meet design goals if CRT tradeoffs were made to reduce the

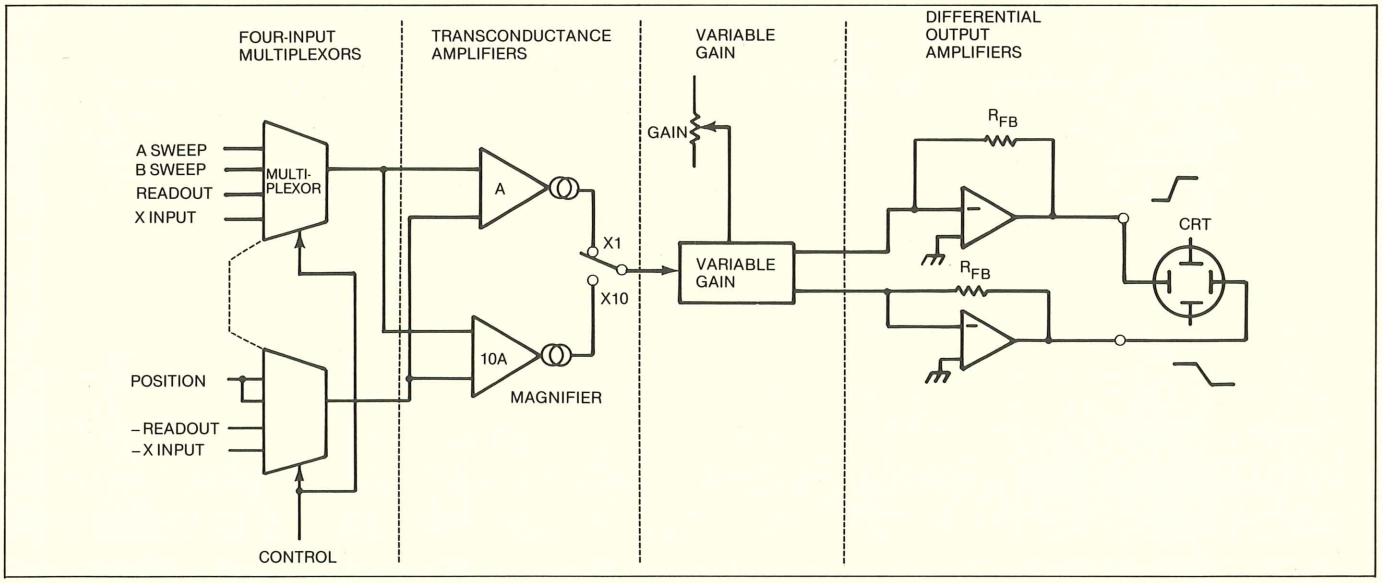


Figure 1. The M215 monolithic amplifier used in the 2400 series consists of a four-input multiplexer; two transconductance amplifiers (one a 10X magnification amplifier); a variable gain stage; and a differential, high-slew rate, output amplifier.

high-voltage requirements for the transistors. These tradeoffs increased CRT sensitivity and amplifier speed and thus made timing accuracy and dynamic range easier to achieve.

Accurate sweep speed ratioing (amplifier gain independent of sweep speed) at the faster speeds dictates that the output amplifier have a clean transient response into the capacitive load presented by the CRT. The highest sweep speed of the 2465 scope (0.5 ns/division) requires the amplifier to slew faster than 4000 volts per microsecond and to quickly achieve a linear CRT sweep deflection without wasting dynamic range. The dynamic range must be somewhat larger than the deflection voltage for all 10 divisions because the output ramp voltage can not instantly achieve linearity, and because of CRT deflection centering tolerances.

To reduce packaging costs and improve component reliability, power levels must be kept low. But high slew rate, wide bandwidth, and a large dynamic range all increase power consumption. Integration offered a way to resolve the performance/power conflicts.

Lower stray capacitance can be achieved because monolithic circuits eliminate many packaging parasitics inherent in discretes. Reducing currents required to drive capacitance at critical nodes saves significant power. Moreover, the lower capacitance combined with the high F_T of the IC process allows the use of higher impedances, further reducing power dissipation.

Because the 2400 series features CRT readout of control settings, cursors, and displays of results, jitter cannot be tolerated. Even slight thermal distortion, caused by resistor or transistor self-heating or by thermal coupling between elements, will cause jitter. Integration offers techniques that can significantly reduce thermal distortion.

The integrated approach also offers size benefits; monolithic design consumes much less board real estate. Monolithics, because of superior high frequency transistor characteristics, device matching, and lower stray capacitance, do not require the many high-frequency adjustments inherent in discrete approaches.

Circuit Development — Beginning at the End

To minimize interface problems, it was logical to design the output stage first. Then the other stages could be designed, working backwards towards the input.

The output stage design was most critical for several reasons. First, the output speed was to be ten times faster than previous monolithic horizontal amplifiers developed at Tek. Second, to meet the fastest speed and breakdown voltage requirements, new circuit techniques would be needed. In addition, the output amplifier would dictate process requirements, such as high breakdown voltage, which would affect critical transistor parameters that would impact the design of the remaining horizontal circuits.

Because the horizontal amplifier's most stringent requirements are for ramp inputs, all stages were simulated with ramp inputs and analyzed by differentiating the ramp output. Differentiation makes linearity and transient response problems much more visible.

Output circuit development was begun using SPICE 2, a computer-aided circuit simulation program developed at the University of California at Berkeley.[1] This tool was essential in designing the output circuit design. SPICE 2 proved that our circuit concepts were feasible and demonstrated the need for a higher speed, high-voltage process.

Because the rolloff of beta and F_T, as a function of current and collector voltage for the high-voltage devices, did not closely match SPICE 2 predictions, a new version of SPICE 2 was developed. This version, known as Spice B, modified the transistor models so that the simulated beta and F_T rolloff curves closely matched the measured parameters. This was necessary because of the wide range of device operating conditions in the class AB output stage.

Early simulation demonstrated the need for a new, higher speed process. To meet this need, the Solid State Group developed the SHHV (super high high-voltage) process. SHHV is a hybrid of the Tektronix SHF-III 6-GHz process. To achieve a 65 V collector base breakdown voltage, a thick, higher resistivity epitaxial layer with deep-implanted guard rings around the base was used. Although some speed was traded off, SHHV devices still have minimum F_T ratings of 2.5 GHz.

The Solid State Group also developed a thin-film Sichrome (silicon/chromium) process that was added to SHHV to produce stable, thin-film on-chip resistors for critical applications such as feedback resistors. Because these resistors have small stray capacitances and low voltage coefficents of capacitance, high-speed linearity and high slew rates were possible at lower power levels. Because these resistors have tighter tolerances and better temperature coefficients than diffused resistors, performance is more consistent and fewer adjustments are required.

Using SCALX-predicted device parameters, circuit development was concurrent with process development. The positive going output stage evolved from the simple Cascode shunt feedback stage to the configuration shown in figure 2A.[2] Battjes F_T doubling stages[3] were used to improve the high-speed response, increase the open-loop gain, and improve the slew rate without increasing power consumption.

Well-known techniques of bootstrapping and stacking were used to further reduce the power consumption and voltage stress on the high-voltage devices. Additionally, the feedback network was designed to be beta dependent to compensate for beta (or alpha losses) elsewhere in the horizontal circuit.

The negative-going output stage is very similar to its positive-going complement except for small changes to optimize for negative going signals. Both stages are compensated with a same-value MOS capacitor, an important factor in this design.

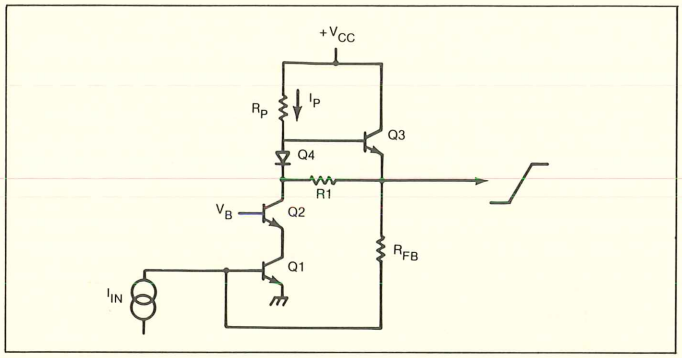


Figure 2A. Basic shunt feedback stage. Transistor Q1 is the control device, and Q2 is used to reduce Miller capacitance. Q3 provides gain for positive transitions to minimize Ip. Q4 provides bias such that the voltage across R1 is nearly zero so that nearly all of Ip is available to charge the capacitance at the slew rate limiting node. R1 is used to stabilize the amplifier and to provide a path for pulldown current for rapid negative transitions. Output voltage swing for the stage approaches I_{IN} times R_{FB}.

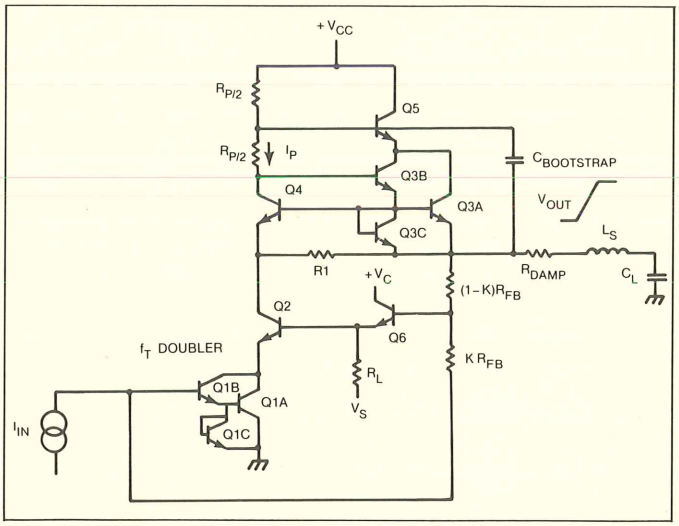


Figure 2B. Several techniques to improve performance. Control device Q1 was replaced by an F_T doubler to increase open loop gain, to increase the gain bandwidth product, and to reduce input bias current. An F_T doubler was used to replace the output transistor Q3 to reduce capacitive loading at Q2 collector, improving slew rate. Since Q_{3A} and Q_{3B} share the output load current, Q_{3B} can be reduced in size without exceeding its current capacity. Also, because of the extra gain offered by the F_T doubler, the dynamic base current is cut in half for equal-size devices Q_{3A} and Q_{3B}, allowing more of the pullup current available to charge node capacitance. These improvements also increase the small signal bandwidth, reducing the settling time for the amplifier.

To bootstrap the pullup curcuit, the pullup resistor was split and transistor Q5 and capacitor $C_{BOOTSTRAP}$ were added. This causes the output voltage swing to be shared by Q3 and Q5, reducing voltage and power stress on Q3. Also, since Q5 base and emitter follow the output voltage during transients, Q3 Miller effect is eliminated. Further, during fast positive transients, the voltage across the lower portion of R_P is maintained, such that almost as much charging current is available at the top of the dynamic range as was initially flowing. This provides a better slew rate/power ratio.

The feedback resistor is tapped to bootstrap transistor Q2 by voltage K (V_{out}). This increases the dynamic range capability without exceeding Q2 collector-base breakdown voltage. Further, R_L can be chosen to load the feedback network as a function of beta. This technique is used to modify the gain of the amplifier to compensate for beta dependancy of gain of this stage or of previous stages.

To achieve maximum performance from feedback amplifiers, it is necessary to be able to adjust feedback compensation capacitance to compensate for tolerances of F_T, resistors, and capacitors. A new circuit[4] (patent pending) was developed that electronically varies feedback capacitance by means of a DC input (figure 2B). Without this internal electronic adjustment, off-chip variable capacitors would have been necessary. In addition to cost, the off-chip method would have added stray capacitances larger than the desired feedback capacitor and introduced stability problems from series inductance. The electronically adjustable circuit works very nicely for differential amplifiers (like the horizontal output amplifier) that have identical feedback capacitors. It also works well for single-ended amplifiers.

A linear electronic means of controlling feedback capacitance can solve other problems. In integrated circuits with multiple identical amplifiers, all amplifiers can be adjusted with one control. The compensation capacitance can be adjusted as a function of temperature or output voltage swing. Further, stored constants could be used by a processor and a digital-to-analog converter (DAC) to alter the amplifier response to fit a particular need.

Adjusting the Gain

Since scope CRTs have as much as a 10 percent deflection sensitivity tolerance, a variable gain stage with good linearity and low thermal distortion is necessary. Figure 4A shows a commonly used variable gain stage. The major limitation with this design is that thermal distortion, linearity, and transient response are a function of the gain setting, with optimum performance achieved only at full gain, if identical-size devices are used.

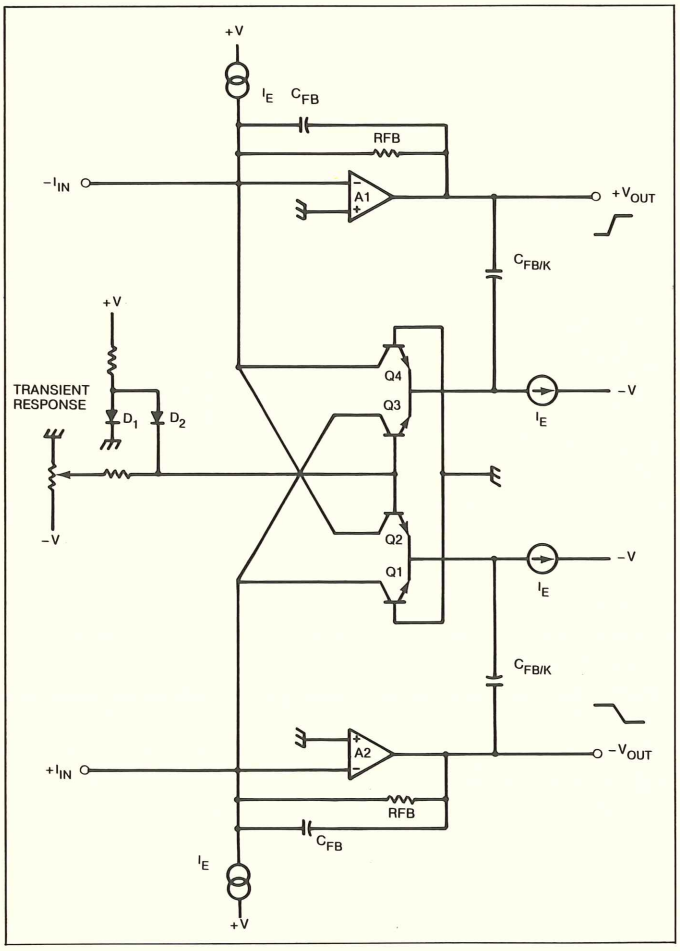


Figure 3. The method used to adjust transient response of the output amplifiers. Symbols A1 and A2 are used to represent the positive-going and negative-going amplifiers. The typical value of C_{FB} required to compensate the amplifiers is connected in the usual manner. Another capacitor, $C_{FB/K}$, whose value is some fraction of C_{FB}, is connected from each amplifier output to the emitters of a Gilbert multiplier. Appropriate current sources are added to insure transient current through these capacitors does not turn off any transistor and to maintain the desired DC operating conditions.

When the transient response control is centered and the currents flowing in D_1 and D_2 are identical, transistors Q_1 and Q_4 will all conduct equal current ($I_{E/2}$). Under this condition, the transient current from the capacitor $C_{FB/K}$ connected to the positive output will split equally between Q_3 and Q_4; similarly, transient current from the other capacitor will split between Q_1 and Q_2. Since the amplifiers are symmetrical, cancellation occurs and the net feedback current will be zero; hence, the effective feedback capacitance is C_{FB}.

It can readily be seen that if the transient response control has the range to transfer all of I_E to either side of the multiplier, then the effective range of feedback capacitance is from

$$C_{EF}\,(MIN) = C_{FB}(1 - 1/K)$$
$$C_{EF}\,(MAX) = C_{FB}(1 + 1/K)$$

Thus, this method offers a symmetrical adjustment of effective capacitance around the desired center value.

This technique can be used for single-ended amplifiers with simple modifications. However, for this case, the adjustment is not symmetrical but has an effective range of

$$C_{EFF}\,(MIN) = C_{FB}$$
$$\text{to}$$
$$C_{EFF}(MAX) = C_{FB}\,(1 + 1/K)$$

The circuit shown in figure 4B,[5] derived from the Gilbert gain cell,[6] can be designed nearly free of thermal distortion, nonlinearity, and transient response variations, if proper device scaling and layout are used. An additional benefit is that mid-range current gain is unity and symmetrically adjustable. Furthermore, the total output current does not vary with gain adjustment. The output of the gain cell drives the summing node of the output amplifier. (This design is also used in the 2400 series' vertical system.)

Magnifying Amplifier Design Achieves Unexpected Benefit

A magnifying amplifier is typically used in oscilloscopes to expand the display ten times. This feature is used to display the fastest sweep speeds or expand the display around an event of interest. Since the magnifying amplifier is always overdriven, its overdrive recovery is critical. In addition to excellent limiting and overdrive recovery, it must have good linearity, low thermal distortion, and stable gain. These are essential for timing accuracy in magnified displays.

In the 2400 series scopes, these characteristics were achieved by improving the basic Quinn Cascomp amplifier[7] (figure 5A). Because of its inherent linearity and low thermal distortion, the Cascomp configuration was the basis for many designs in the 2400 series scopes. It was used in such diverse applications as the trigger, horizontal output, and all vertical system IC designs.

Improvements[8,9] in the Quinn Cascomp are shown in figure 5B. Resistors were added to the base connections of the common-base stage to compensate for transistor alpha loss. Because overdrive recovery was critical, clamping transistors were used to limit outputs and keep the common-base stages conducting at all times. To provide good thermal distortion characteristics under overdriven conditions, the common-base stages were bootstrapped. Finally, because the error amplifier contributes a significant portion of the output signal current, an improved buffered error amplifier was used to improve linearity and limiting characteristics.

A SPICE B simulation of this modified Cascomp stage revealed an overshoot on the differentiated response to an input ramp. This overshoot is typical for almost any method of limiting or clamping in circuits amplifying ramps. After some effort to reduce this overshoot, it was more productive instead to peak the output stage with this overshoot. This peaking reduced the time

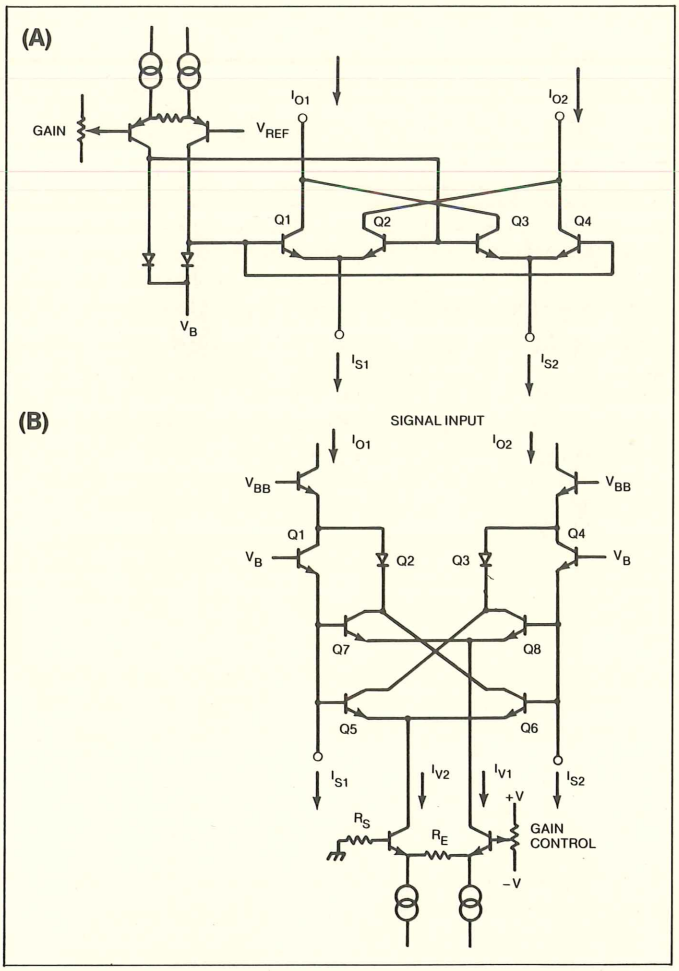


Figure 4. The chief limitations of the commonly used gain control stage shown in figure 4A are that linearity and thermal distortion are a function of gain.

The circuit shown in figure 4B solves these problems. Gain of this stage is defined as

$$A = 1 + \frac{I_{V2} - I_{V1}}{I_{S1} + I_{S2}}$$

For minimum gain, transistors Q_7 and Q_8 are off. The active circuit is now the Gilbert gain cell, and thermal distortion and non-linearity are ideally zero if transistor areas are properly ratioed. This also occurs at minimum gain (Q_5 and Q_6 off) if $Q_5 - Q_8$ are all the same size. For midrange gain, both pairs $Q_5 - Q_6$ and $Q_7 - Q_8$ will amplify and produce distortion; however, the signal and the distortion are of opposite polarities and will cancel. It can be shown that this error cancellation occurs throughout the gain range.

and voltage required to get the output stage ramping at the correct speed. After being verified by SPICE B, this peaking was incorporated into the design.

Multiplexing Input Stage

To complete the design, a high-speed analog multiplexer input stage was developed [8]. This stage provides very high input

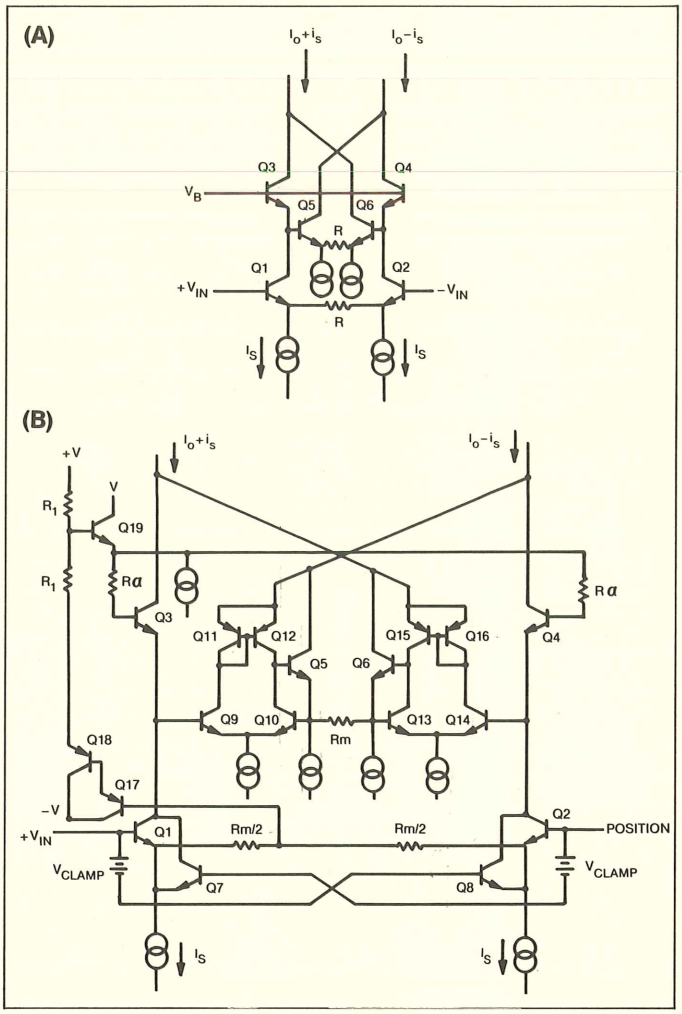


Figure 5. In basic Quinn Cascomp amplifier (A), the error amplifier Q_5 and Q_6 senses the error signals at Q_3 and Q_4 emitters, producing an error correction signal. This signal is summed at the output nodes, linearizing the amplifier.

For overdriven amplifiers, the error correction amplifier produces a large percentage of the output signal. This requires that the error amplifier itself must be very linear and free of thermal distortion. The buffered error amplifier in (B) shows the configuration chosen to meet this constraint.

Other improvements shown are the addition of alpha compensation resistors in the bases of Q_3 and Q_4. These resistors can make the amplifier gain independent of transistor alpha and improve linearity.

Transistors Q_7 and Q_8 are biased by Vclamp (Vclamp $< I_s$Rm) to limit the current signal to Q_3 and Q_4. This prevents transistors Q_3 and Q_4 from turning off, improving transient response and more precisely establishing the limits on the output current of the amplifier.

Transistors $Q_{17\text{-}19}$ are used to bootstrap the common base stages to adapt the error correction scheme for the single ended horizontal application and to reduce thermal distortion when the amplifier is overdriven.

impedance and fast selection of one of four input channels (A sweep, B sweep, Readout, or X-axis input). The horizontal application requires fast settling and very low switching thermal effects. The calibration system requires very accurate channel matching to provide the excellent cursor timing measurement accuracy of the 2400 series. (This circuit is also used in the 2400 series' trigger and vertical systems.)

Design Choices Proven in Performance

The prototype parts were fabricated with the Tektronix SHHV process and packaged in a 24-PIN, power dual-in-line package. This new package was developed by the SSG packaging group in parallel with the integrated circuit. Part performance agreed with the SPICE B simulations (typically within 10 percent). For example, a differential output voltage swing of greater than 100 was achieved as predicted along with a slew rate exceeding 5000 volts per microsecond for each side of the output amplifier.

The transient-response control also performed as predicted, with optimum amplifier performance occurring near the center of the control range. Some minor discrepancies in transient response occurred at the fastest speeds, but these were corrected by slightly changing external damping components. The effect of the transient response control is shown in figure 6A. The middle waveform (optimum setting of the transient response control) easily meets the linearity requirements; the other two waveforms show the extreme settings of the transient response control. Figure 6B shows the optimum amplifier response at the fastest sweep speed. These differentiated ramp waveforms were monitored using a Tektronix CT-1 current transformer to sense current into the capacitive load.

The benefits of the integrated horizontal amplifier are reflected in the timing specifications of the 2400 series oscilloscope. Timing accuracy specifications are better than 1.3 percent unmagnified and better than 1.8 percent magnified for all sweep speeds, even including 0.5 ns/division. Only one transient response adjustment is required in the 2465 (for 0.5 nanoseconds/division). None were required for the 2445.

The major source of thermal distortion proved to be the output stage thin film feedback resistors. The large voltage swing and resulting large temperature change (about 50°C) in these resistors causes unacceptable distortion if the thin film temperature coefficient of resistance (TCR) exceeds 100 ppm. A second source of distortion is the variable gain stage which contributes about 0.15 percent distortion at the extremes of its range. A redesign of the M215 increased the size (area) of the feedback resistors by a factor of 10 to fix the first problem, and changed to the variable gain stage to reduce thermals. Prototypes of these parts indicate that the one thermal distortion compensation adjustment, presently required, might be eliminated.

Because of the tight timing specifications of the 2400 series, another adjustment to precisely set the ratio of magnifying amplifier gain to the unmagnified gain is necessary. The only other adjustments, centering and gain, are necessary to match CRT tolerances. Gain matching of the input ports is better than 0.1 percent. This is important because the gain from the readout

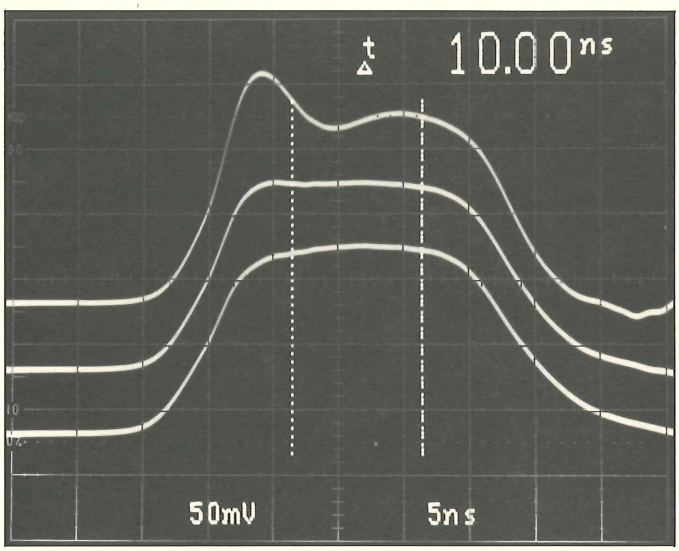


Figure 6A. The horizontal amplifier response at 1 nS per division over full range of transient response control. Vertical sensitivity is equivalent of 10 mA per division. Cursors indicate the on-screen portion of the waveform.

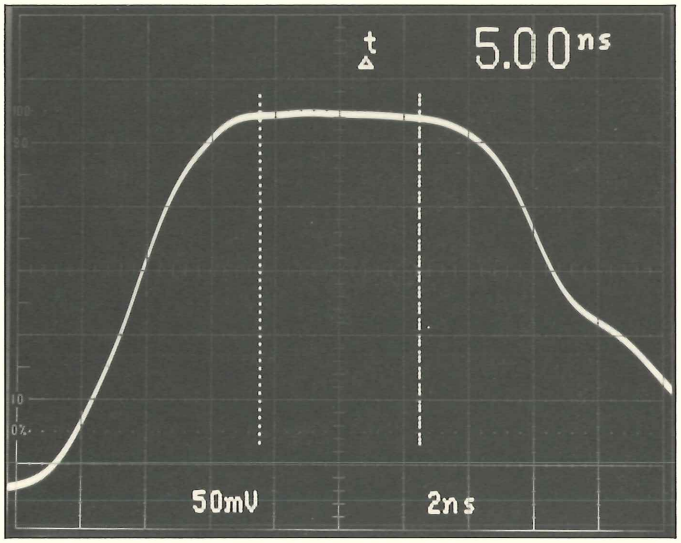


Figure 6B. Amplifier response at 0.5 nS per division. Equivalent vertical sensitivity is 10 mA per division. Again, cursors show the portion of waveform where good linearity is required.

must accurately match the gain from the A-sweep input to make possible accurate cursor delta-time measurements without adding another adjustment.

Total device power dissipation of the M215 is typically 2 watts — previous discrete designs capable of 0.5 ns sweep speeds required about 8 watts. This low power level minimizes power DIP heatsinking; the component is simply fastened to the printed circuit board.

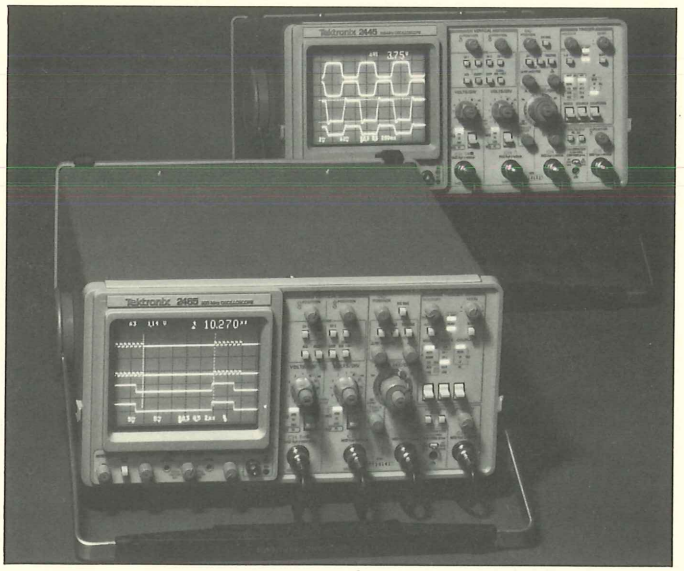


Figure 7. The M215 monolithic horizontal amplifier is a key component in the new 2400 Series. These instruments have the highest level of circuit integration ever used in a portable scope.

Summary

This component has made a significant contribution toward producing an instrument with solid timing specifications and excellent linearity, at an attractive price. In addition, some of the techniques developed have been applied in nearly all of the integrated circuits developed for the 2400 series oscilloscopes.

For More Information

For more information, call Art Metz, ext. B-3640. □

References

[1] Cohen, E., Program reference for SPICE 2, Memo ERL-M592, Electronics Research Laboratory, University of California at Berkeley, June, 1976.

[2] Metz, Arthur J., patent pending, Tektronix, Inc., Beaverton, Or.

[3] Battjes, Carl R., U.S. Patent 4,236,119, "Monolithic Wide-Band Amplifier."

[4] Metz, Arthur J., patent pending, Tektronix, Inc., Beaverton, Or.

[5] Metz, Arthur J., U.S. Patent 4,340,866, "Thermally Compensated Variable Gain Differential Amplifier."

[6] Gilbert, Barrie, U.S. Patent 3,689,752, "Four Quadrant Multiplier Circuit."

[7] Quinn, Patrick A., U.S. Patent 4,146,844, "Feedforward Amplifier."

[8] Schlotzhauer, Kenneth G. and Metz, Arthur J., U.S. Patent 4,322,688, "CASCODE Feedforward Amplifier."

[9] Metz, Arthur J., patent pending, Tektronix, Inc., Beaverton, Or.

NEW TOOLS SHOULD REDUCE GPIB EVALUATION TIME

The Automated Instrument Compatibility Evaluation (AICE) group now provides a portable package of GPIB evaluation tools. Called TestPak, the tools consist of questions and software tests to help you design GPIB products to comply with the IEEE-488 and the Tek Codes and Formats (C&F) standards.

TestPak follows the product through all NPI phases. The early phases are covered with check-list questions about hardware, firmware, and system features. In later phases, prototype devices are tested with 4050-based software procedures that check most GPIB and C&F standards issues. These are checked with AICE-provided test tapes. The tests are tailored for non-controller products employing a TI 9914A or INTEL 8291A GPIB chip.

These tapes are available:

- General C&F Semantics
- Syntactic Pattern Tests
- GPIB Interface Message Handling
- High-Speed Timing.

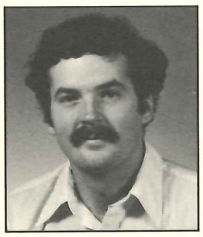
For all but the high-speed timing tests, all you need is a 4050-series controller with an R14 ROMPACK. AICE supplies the software tapes and a special test-fixture.

High-speed timing tests require an Interface Technology IT-488 Bus Analyzer/Monitor and a Tek 7D01/DF2 Logic Analyzer/Formatter. Both are available for short-term loans from the AICE group.

In July, AICE will incorporate the TestPak procedures into its evaluation plan for NPI products. Although this test package will greatly speed product testing by AICE, its real value is in enhancing GPIB evaluation within the product groups. Overall, evaluation time should shorten as firmware is more efficiently debugged, thus reducing the time required for hands-on AICE evaluation.

For more information, contact Bob Cram, manager, AICE group, 50-761, 627-1796. □

AN ELASTOMERIC INTERCONNECT FOR OSCILLOSCOPE PROBES



Ken Smith is a packaging engineer in Hybrid Circuits Engineering. He joined Tek in 1979. His degree is a BS in general Engineering from Oregon State University. Ken is presently project leader of High Performance Packaging, which is developing a new generation of high frequency, high power hybrids. Ken can be reached at B-6362.

A radically new design for packaging oscilloscope-probe tips has been developed. Conventional hand-soldered, discrete R-C components were replaced by a thick-film substrate with conductive elastomeric interconnects. The elastomeric interconnect was a new solution to a strain-gauge problem caused by miniturization. The new product outperforms past products in all respects: electrical, mechanical, reliability, and production costs.

In recent years the use of high-density packaging created a demand for a 350-MHz, 10X-attenuation probe capable of probing 100-mil center-to-center spacings.

The conventional 10X-attenuation probe is too bulky for probing high-density circuits. It is also inherently expensive to produce. Such probes are typically discrete hand-soldered RC networks. In the late 70s a second-generation probe was developed by Accessories Engineering that uses a laser-trimmed thick-film substrate. Although this substrate also had hand-soldered pins, it was electrically superior to discrete components.

In 1979, Accessories Engineering was asked to design a probe to match the performance goals for the 2465 oscilloscope. We decided that the probe tip was to be a threaded metal barrel 1.3 inches long with a 0.095-inch diameter. A method was required to interconnect a 9-Mohm resistor and a 10-pF capacitor in parallel within the 0.080-inch inside diameter of the barrel. Additional requirements were low parasitic capacitance, 2400-V voltage withstand, low dielectric absorption, and low strain-gauge effects. Figure 1 shows a photographic comparison of the second-generation probe (top) with the new probe (bottom). Figure 2 shows the three types of RC attenuators.

Initial packaging efforts centered on attaching a thin, chemically milled lead between the thick-film substrate and the connector pins. Soldering, spot welding, and laser welding were tried – with poor to fair results. It was also difficult to handle the fragile parts and to minimize lead lengths without exceeding the strain-gauge specifications. Experiments showed that as the lead length was reduced, more shear and moment forces were transmitted to the substrate when probing a circuit. This bent the substrate, stretching or compressing the resistor, causing a change in value. As much as a five percent strain-gauge effect was seen

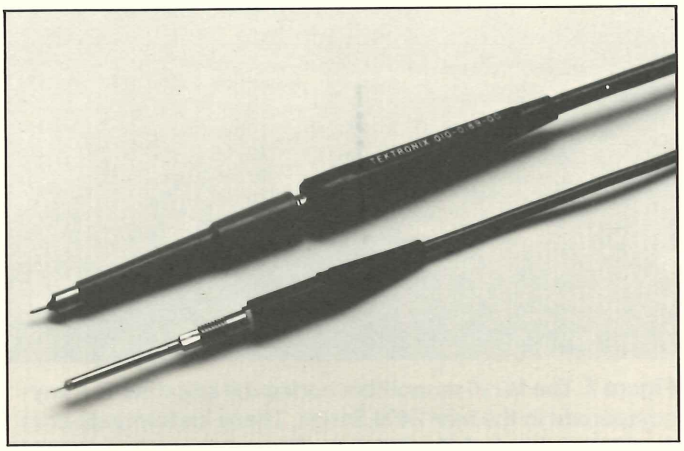


Figure 1. The P6010 series probe (top) and the new P6130 "microprobe" (bottom). The microprobe is small enough to probe high density circuits on 100 mil center-to-center spacings.

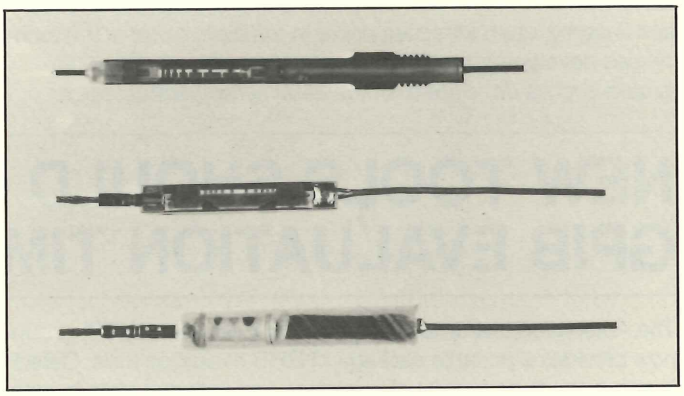


Figure 2. Top to bottom: discrete hand-soldered RC combination used in traditional probes, thick-film substrate with hand-soldered pins as used in more recent designs, and the thick-film substrate with elastomeric interconnect used in the P6130 series microprobes.

when the interconnect length was reduced to meet the tip capacitance goal. It was apparent that a radically new interconnect method was required.

Package Description

The new P6130 probe consists of a screw-machined beryllium-copper barrel, a thick-film ceramic substrate, and the tip and tail subassemblies. The substrate provides the front end of a 10X attenuator; the attenuator consists of a 9-Mohm resistor and a 10-pF capacitor. The tip and tail subassemblies include the contact wires and the conductive elastomers. The plastic is a low-dielectric constant, injection moldable fluoropolymer. The thin wall surrounding the elastomer provides electrical insulation

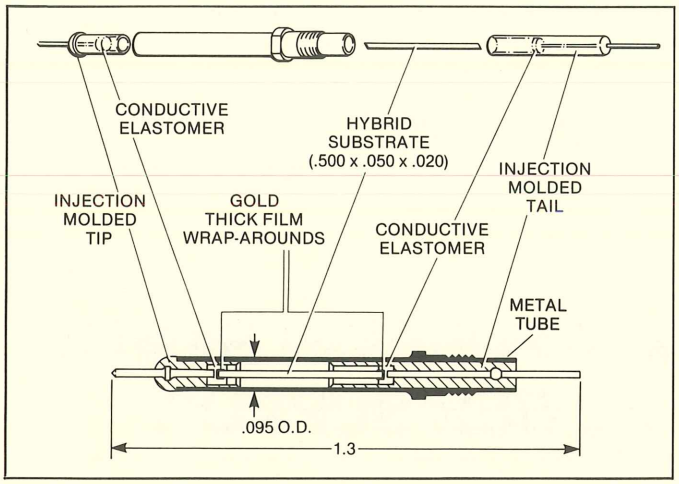


Figure 3. An exploded view of microprobe components. Parts are slipped in from each end and held in place by crimping the metal tube.

from the barrel and serves to guide the substrate into place during assembly. Figure 3 is an exploded view of the microprobe components.

The elastomer is a silver-filled silicone. It makes a low resistance contact (approximately 0.050 Ω) between the molded-in wires and the thick-film wraparounds on the substrate. A gas-tight seal protects contact integrity from environmental exposure. The substrate is totally isolated from bending moments, which cause the strain-gauge effects on ceramic substrates.

The assembly process used for the new probe is very simple and so results in a low-cost, high-throughput component. Labor-intensive soldering and flux-removal steps are eliminated by employing a Tektronix-developed pressure interconnect. The pins, substrate, and elastomer are sequentially slipped into the barrel, compressed, and crimped in position.

Assembly Process

A 1 × 60 array of substrates is printed on a 0.500 × 3.0-inch alumina substrate (see figure 4). This substrate area includes edge-printed wraparounds and a digitally trimmed, tri-plate capacitor. After thick-film processing, the resistors and capacitors are laser trimmed. Because the parts are fragile, teflon tape is applied to one side to facilitate substrate handling after laser scribing.

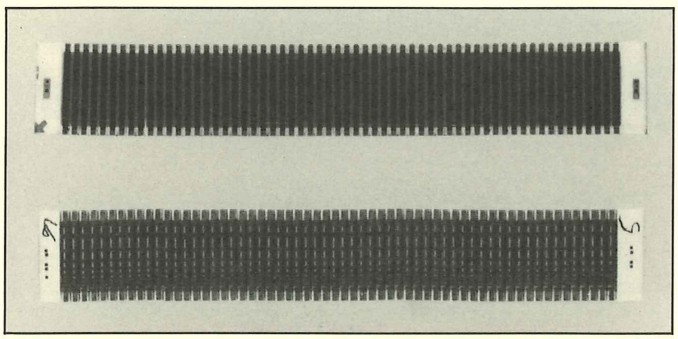


Figure 4. Twelve screen printing operations and two laser trims are done on a 1 × 60 array of substrates. Reduced parts handling reduces costs.

Separation of the 0.020 × 0.050 × 0.500-inch substrates is accomplished with very deep pulsed CO_2-laser scribing. Hole depth must be in the 0.016- to 0.020-inch range; these holes occasionally punch completely through the 0.020-inch substrate. Each strip is then placed on a thin neoprene pad and covered with steel shim stock; then, a roller is run over the top. This propagates the laser scribe and separates the substrates. Excellent yields have been achieved with this system.

Visual inspection is done on the teflon tape, which is then stripped off. The substrates are passivated by total immersion in a dilute conformal coating; this coating is then dried and cured. Surprisingly, this coating does not inhibit the pressure contact system. The injection-molded parts are loaded with conductive elastomer pellets, and the probe tip is inserted into the barrel and crimped in position. The substrate is inserted, properly oriented, and the tail is slipped in. A calibrated force is applied to the system during the final crimping operation to ensure low contact resistance.

Environmental Testing

Perhaps the only drawback to using an elastomeric interconnect in this system is the relatively large thermal coefficient of expansion (TCE) of the elastomer. It is an order of magnitude higher than the TCE of the plastic tips and tails, and two orders higher than that of the metal barrel. In the initial design, severe problems were seen during temperature cycling (−62°C to +85°C). Both ends shifted outward anywhere from 2 to 20 mils and contact resistances increased as much as 100 to 200 ohms.

In several reliability samples resistance did not initially shift, but their resistances increased as much as 20 ohms after 500 hours at 70°C load life. We assumed this long period before failure was caused by stress relaxation of the system. Failure analysis showed that because the elastomer was almost totally enclosed by the plastic and substrate, the volumetric expansion essentially acted as a hydraulic system. Hence, efforts to increase the holding power of the crimps were fruitless.

The solution was to alter the effective linear-expansion factors of the system by adjusting the lengths of the elastomer and the barrel. This allowed elastomer compression to be essentially constant with temperature change. Since the elastomer and barrel lengths were changed, no failures have been detected after temperature cycle or 1000-hour high-temperature load life testing.

The standard mil-spec ten-day humidity test initially caused severe dielectric absorption, or "hook," in the thick-film capacitor. Hook is manifested by a rounding off of the leading edge of a square-wave signal applied to the attenuator. This rounding off is caused by an apparent shift in capacitance, affecting response in the 10-20 kHz range. We determined the cause was ionic contamination of the dielectric during assembly. The effect was intensified by the exposed dielectric along the sides of the narrow capacitor absorbing moisture. Ultrasonic cleaning and vapor degreasing reduced the effect, but a new passivation was also required. A very thin coating of silicone solved the dielectric-absorbtion problem, presumably by acting as a moisture getter.

A second hook problem, unrelated to dielectric contamination, was successively resolved. The problem was traced to poor laser trimming of the digital capacitor. Slag left in the laser kerf offered a high-resistance current path in a humid environment.

This created an extra RC network in the attenuator, which acted very much like hook. A double-cut trim operation, combined with vacuum impregnation with silicone, solved the problem. Presumably, the vacuum pulls the passivation into the kerf and adjacent microcracks and thus removes the current path.

Conclusion

For the first time, an elastomeric interconnect has successfully been used in an oscilloscope probe tip. Superior performance and reduced cost were achieved by eliminating conventional soldered interconnects from the system. □

Ken Smith presented a paper on the elastomeric interconnect at the International Electronic Packaging Society, November 14, 1982. A patent application has been filed.

DO WE NEED TO MAINTAIN FOREIGN RIGHTS FOR A VERY EFFICIENT POWER SUPPLY?

Several years ago, Tektronix patented a very efficient power supply. At this time, Patents, Trademarks, and Licensing needs to either invest the resources to maintain foreign rights or let those rights go.

The key factor in the decision is the potential for further use in Tektronix products. The power supply is presently used in the GMA 103 Display. Does someone else at Tek want to employ the invention?

Tek's U.S. patent describes an "Efficient Power Amplifier with Staggered Power Supply Voltages." The power supply description (adapted from the patent description) follows:

The amplifier comprises output devices (emitter follower transistors) which are in parallel and are selectively enabled by power supplies of different output voltages so that the lowest usable power supply provides the load current.

An input signal V_{IN} is applied to the bases of the transistors by way of diodes. A switch supplies current from a power supply V_ϕ only to the base of transistor Q2 for output voltages less than the voltage of the power supply V_1 by a fixed amount and only to the base of the transistor Q1 for greater output voltages. The emitters of the output transistors are connected to the negative side of the power supply V_ϕ and to the load. If the input voltage V_{IN} is such that the switch provides current to the base of the transistor Q2, the current will elevate the voltage at the base of Q2 above V_{IN} by an amount equal to the voltage drop across the diode. The voltage at the emitter of Q2 will be less than the voltage at its base by an amount equal to its forward base-emitter junction voltage.

Thus, the voltage V_{OUT} is very nearly equal to V_{IN}. When the voltage V_{OUT} reaches a threshold value such as to cause the switch to select the transistor Q1, the diode D3 is reverse biased so that Q2 may be elevated above the power supply V_1.

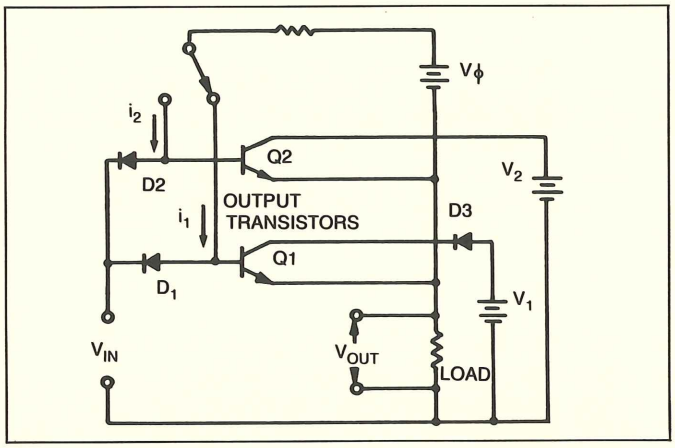


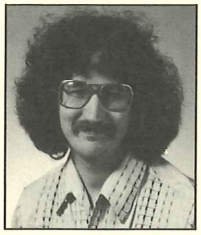
Figure 1. By employing only one output transistor at a time, this power supply is more efficient than the usual supply that employs two or more transistors as higher output voltages are required.

Because the currents I_1 and I_2 are supplied by V_ϕ instead of V_1 or V_2, they are independent of the output voltage V_{OUT}. Therefore, the amplifier can accommodate an output voltage from any power supply up to a saturation drop below that supply voltage.

Only one of the output transistors is enabled at any one time, and accordingly, the power consumption is lower than in the case of amplifiers in which one output device is enabled for low output voltages, two devices for higher output voltages, three output devices for yet higher voltages, and so on.

For further information, call John Smith-Hill, Patents, Trademarks, and Licensing, 643-8152. □

HANDLING COLOR SIMPLY THROUGH NEW USER INTERFACE TECHNIQUES



Gar Bergstedt is a software engineer III in Graphic Systems Products, part of IDD. Gar joined Tektronix in 1978. He holds a BA in physics/computer science from Evergreen State College, Olympia, Washington.

Color choices by users in computer graphics work vary with each application, individual perceptual differences, and personal preferences. A simplified user interface, such as that found in the new 4105 Color Display Terminal, gives the user the freedom and the power to easily identify and change colors, in order to achieve the "right" colors for an application.

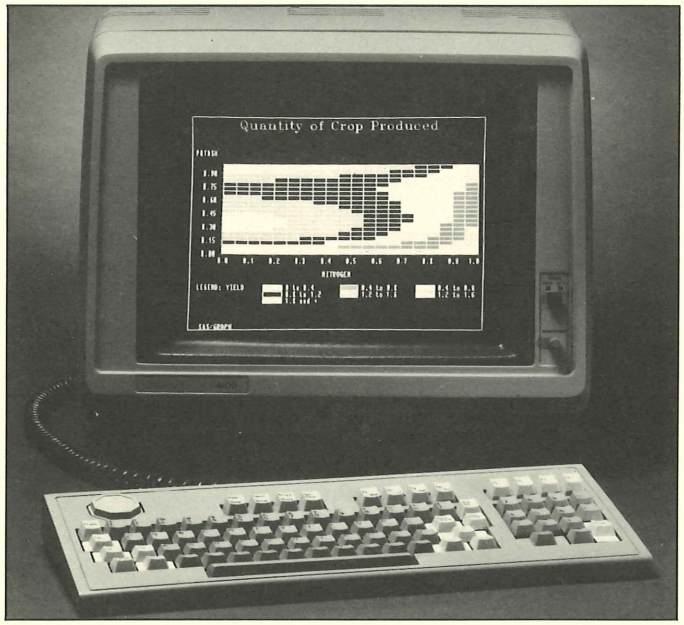


Figure 1. The 4105 Computer Display Terminal.

The process of color selection is becoming more important as the world moves toward better ergonomica. The user should not have to worry about color theory or color coordinates just to change a color. Most users do not have the time, and an increasing number do not have the knowledge, to input reams of data before results can be seen and used. The 4105 speeds the user past the tedious task of color setup to his ultimate goal: results.

The 4105 employs a simple method of selecting and modifying colors. The user interface does the job without exacting an inordinate amount of user time or effort. The 4105 is a new member of the 4100 Series of graphics terminals directed at professionals engaged in technical problem-solving. By handling both graphics and text-related functions, the 4100 Series terminals provide a low-cost solution to professional graphics needs by offering, in essence, two terminals in one.

Why Color?

A good man-machine interface fits the user's cognitive and perceptual capabilities. Cognitive considerations concern the manner in which information is translated from the display to the user; ideally, such translation should be instant and clearly understood.

Color can significantly enhance both the perceptual and cognitive aspects of the user interface. Perceptually, color allows the user to work in a more natural environment: We live in a colorful, not a black and white world. Color displays allow the user to assimilate information through his chromatic as well as his achromatic visual channel, more fully employing the user's information gathering and processing powers. Color can also be used to discriminate between similar data events that might be confused on a monochromatic display.

Cognitively, color aids tremendously in locating and distinguishing information. Finding a single element in a complex background is easier when colors are properly used; conversely, color usage can be used to signal disregard of unimportant information. Color can be used to organize information into logical groups, such as high- and low-priority items. Proper use of color takes advantage of culturally-ingrained color mnemonics: flashing reds for "warning" and quiet greens for "safe."

Although the appropriate use of color can boost comprehension and productivity, bad usage has the opposite effect. If misused – too many colors, inappropriate colors, color in the wrong places – color can complicate the man-machine interface.

The Requirements for a Color-Selecting Interface

Color enhances many applications. In fact, in some applications color is essential: Productivity in integrated circuit design, architectural drafting, and even textile manufacturing now depends on color terminals.

What makes or breaks a terminal as a productivity enhancer is how easy (or difficult) it is to use. Many color terminals now on the market demand a good understanding of color theory and much data entry before the user can even attempt to use the color feature.

The ease with which the user can select and modify colors at his terminal depends on the color model used. Although it is difficult for novice users, the traditional RGB (red, green, blue) model is

the generally accepted color standard in the computer industry. The red, green and blue vectors of the RGB model serve well in engineering equations, but they don't work well where the user works intuitively. To use the RGB model effectively, the user must understand color theory and how colors combine.

A color model should be simple enough for low-end users yet sophisticated enough to accommodate complex color applications. The model should specify colors in a way that is equally understandable to both people and machines, so that both can duplicate the colors desired. The user interface employing such a model should be able to grow with the user and with hardware upgrades. Certainly the model must be compatible with peripherals, such as hard copy devices.

Above all, the color model should simplify interactions between user and terminal. The Tektronix HLS color model satisfies these requirements. All Tektronix color display terminals use this hue, lightness, and saturation model (see sidebar).

The 4105 User Interface to Color

The premise underlying the design of the user interface to color in the 4105 terminal is that the color selection process should be fast and satisfying to the user. To place the elegance of the HLS model at the user's fingertips, designers employed special-function keys and a simple color display legend. Three keys control HLS components (coordinates), the color-menu key provides fast selection of one of nine hard-to-get colors, and the legend shows the color produced by keying in changes to those coordinates.

When the user turns on the 4105 and wishes to change the default colors, he simply pushes one button, SET COLOR. Instantly, a color legend is displayed. The legend presents a key to the function keys and a current status of which color is selected and what its HLS values are. A message is also displayed. This asks the user if he would like the current palette of colors to be displayed. If the user presses the erase key it will erase the screen and display all possible colors. If the erase key is not pressed, the interface assumes that the user wishes to modify an on-screen picture.

The terminal allows eight colors to be displayed at one time in the graphics area and an additional eight in the dialog area. The dialog area is an area reserved for commands; this separation prevents alphanumeric text from interfering with work in progress in the graphics area. Within the dialog area, the user can alter the foreground and background color, change an underline, boldface a label, slow-blink a graphic element, or implement reverse video.

The user selects a desired color by locating the crosshair cursor over it on the screen. To modify the index color, the user either presses the hue, lightness, and saturation keys or presses the color-menu key, which displays a menu close to the crosshairs. (We call this a pop-up menu because it pops up anywhere on the screen.) The user then locates the crosshairs over one of the nine menu colors and the previous screen color is instantly changed. To get rid of the pop-up menu, the user releases the color-menu key. As these keys are pressed, a legend color changes, and the corresponding numerical HLS coordinates of the color are displayed. The user can duplicate a desired color anytime by using these coordinates.

The user can also undo any changes made in the interface colors, just by pressing one key.

The simplicity and ease of use of the 4105 interface is appreciated more when compared to the user effort required by earlier color selection systems. Most color terminals demand several numerical inputs before a single color can be examined for possible use; in contrast, the 4105 user simply pushes the SET COLOR key. Some competitors' terminals require two values to specify the index, and three more to specify the color coordinates. Even when the user satisfies these requirements, the specified color is not displayed on the screen. For that to happen, the user must build an object on the screen. As an example, to build a polygon the user must specify a polygon color, then build a polygon, coordinate by coordinate – a multistep and often laborious process.

The 4105 color interface allows the user to select and view, on one screen, all the colors he will use. The system is iterative: The color being selected appears and changes on the screen as the user presses the H, L, S, and color menu keys. The updated display instantly gives the user the feedback needed to speed through the iterative process. If a change of color is desired, a few presses of the function keys do the job.

Once the user has completed an image, he will often need to transfer it to an output device for a permanent visible record. The 4105 interface allows on-screen colors to be easily aligned to match those of the output device. Since such devices usually have colors fixed by such invariables as plotting inks, it is necessary to adapt the colors of the terminal display to the capacity of the output device. The 4105 adapts easily and promptly to a peripheral's color limitations.

Engineering the 4105 User Interface to Color

The most complex aspect of the 4105's interface to color is saving and restoring the terminal environment and the color pop-up menu. About 25 percent of the interface code interface is devoted to these functions.

Tektronix HLS Color Model

The HLS model can be visualized as a figure consisting of two cones joined at their broad ends. Within the double cone, the visible color spectrum is encompassed and represented in values of hue, lightness and saturation (see figure 2).

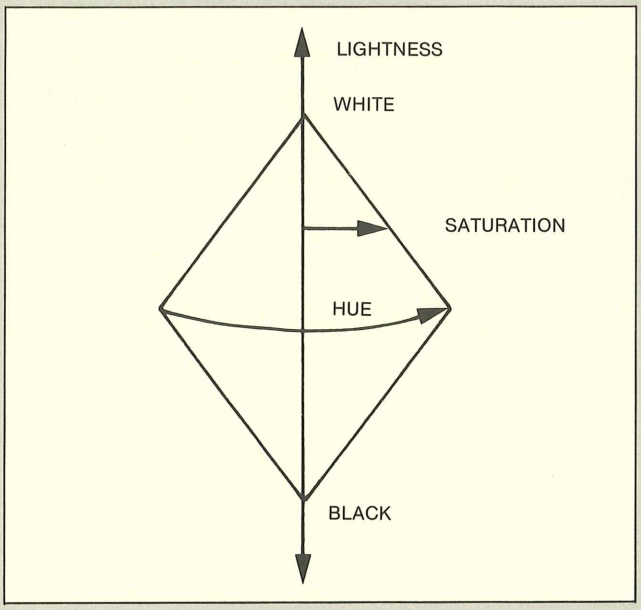


Figure 2. The Tektronix developed HLS model.

Hue is represented in the model as a degree of rotation around the axis of the conical figure. A hue represents a single color (red, green, blue . . .). The axis represents lightness or how dark or light it is. The lower point of the model, representing black, is valued at zero percent; the upper point of the model represents the lightest color, white, at 100%.

The length of the hue radius represents saturation, or the color purity. Saturation is valued from no color purity (gray or zero percent) to maximum color purity (full red or full green = 100 percent).

Any color offered by the terminal can be exactly represented by three HLS coordinates. Equally important, this spatial model is considerably easier to work with than the vector-based RGB model.

When the user initiates the interface with the SET COLOR key, the 4105 interface first saves the current terminal environment: the screen contents, any active terminal modes, and both graphics and dialog areas. This means saving from 20 to 50 terminal components such as gin cursor color, polygon environment, and keyboard ID. Saving and restoring the terminal environment makes the interface transparent to the user. You can enter and exit without losing a keystroke.

After saving the terminal environment, the interface adjusts its environment to suit itself, then it displays the legend and default-color message. Next it activates graphic input. Then it waits for a state change, such as a key being pressed.

All keys related to the interface are single-function keys in that they are independent of mode. This makes it easy for the user to remember what each key does without having to worry about whether or not he is in the correct mode.

The most complex of the special-function keys is the color-menu key. When this key is pressed a menu of nine color names pops up near the crosshairs. The current crosshair position is saved and then the 4105 loops, changing the crosshair position while the user has the color-menu key pressed. If the crosshair is placed over a menu-color name, the screen color and the legend HLS values are changed. If the color-menu key is released, the menu is replaced by the original display.

Operating the hue, lightness and saturation keys is straightforward. Each keystroke increments or decrements the respective value. The RESTORE key restores HLS values for a particular index to the values before the cursor touched that pixel. A (SHIFT) RESTORE restores HLS values for all indices to their values before the interface was entered.

The interface took only two man weeks to program once the specs were outlined. The interface requires direct access to the bit map, keyboard and other resources. This is not a simple task to implement on any color terminal.

For More Information

For more information, call Gar Bergsted, 685-3811. □

PAPERS AND PRESENTATIONS

The table below is a list of papers published and presentations given during recent months.

While providing recognition for Tektronix engineers and scientists, the presentation of papers and articles contributes to Tektronix' technological leadership image.

If you plan to submit an abstract, outline, or manuscript to a conference committee or publication editor, take advantage of the services that Technology Communication Support (TCS) offers.

TCS provides editorial and graphic assistance to Tektronix engineers and scientists for papers and articles presented or published outside Tektronix and obtains patents and confidentiality reviews as required.

Call Eleanor McElwee on ext. MR-8924. □

JANUARY

TITLE	AUTHOR	PUBLISHED	PRESENTED
Developing a Reconfigurable Computer-Aided Test System	Steve Jumonville	**Electronics Test**	
Laser Optometric Assessment of Visual Display Viewability	Gerry Murch		Society of Photo-Optical Instrumentation Engineers (SPIE) Meeting, Los Angeles, CA
A Dual-Color AC TFEL Display	Dick Coovert Chris King		Society of Photo-Optical Instrumentation Engineers (SPIE) Meeting, Los Angeles, CA

FEBRUARY

TITLE	AUTHOR	PUBLISHED	PRESENTED
Portable Scopes Born of Chip, CRT, and Assembly Automation	Clint Brannon	**Electronics**	
Computer-Aided Manufacturing (CAM) Semiconductor Processing Monitoring	Dave Cole		Portland State University

MARCH

TITLE	AUTHOR	PUBLISHED	PRESENTED
IEEE-802 Standards Committee Selects Multiple LAN Techniques	Maris Graube	**Electronics**	
Control and Evaluation of Resistivity in N – /N + Epitaxial Deposition	Vugranan Kannan Jack Sachitano	**Journal of the Electrochemical Society**	
To Ease High-Speed Logic Analysis, Start With Measurement Basics	Steve Palmquist Dave Chapman	**EDN**	
Drilling and Plating of Teflon/Glass Printed Circuit Boards	Terry Smith		NEPCON West '82, Anaheim, CA
CAD/CAM for GaAs ICs	Tom Reeder W.B. Wylie		Conference on Microwave Systems and Applied Technology, Washington, DC
Selection Criteria for Graphics Hardware	Henk de Groot		Computer Graphics Applications for Management and Productivity (CAMP83), Berlin, Germany
Present and Future Color Display Technologies for Graphics	John McCormick		Computer Graphics Applications for Management and Productivity (CAMP83), Berlin, Germany

MARCH

TITLE	AUTHOR	PUBLISHED	PRESENTED
Color Graphics Hard Copy: Where Can Ink Jet Win?	Chuck Davis		IGC Ink Jet Conference, Amsterdam, Holland
Tight-Tolerance Trimming of Megohm-Value Resistors	Oscar Olson		ISHM Chapter Meeting, Wilsonville, OR
Venturing Beyond Traditional Power Measurements with Waveform Processing	Gary Kirchberger Clark Foley		POWERCON 10, San Diego, CA
User Interface Aspects of a Desktop CAD System	John Harms		National Design Engineering Conference

APRIL

TITLE	AUTHOR	PUBLISHED	PRESENTED
The Future of American Instrumentation (Guest Editorial)	Tom Long	**Microwave Journal**	
Logic Analyzer Suits All Levels of Expertise, Adapts to Many Needs	Doug Boyce Dave Moser	**Electronics**	
Human Factors in Computer Systems: The Affective Area	Herb Weiner		CSG/RMC Workshop on Human Factors in Computer Systems, West Vail, CO
Advancements in Commercial ETE Technology Through the 80s	Kevin Considine		American Defense Preparedness Association (ADPA), Washington, DC
High-Level Language Debugging Aids	Doug Johnson		Electro '83, New York, NY
A Misconception Concerning Normal and Intrinsic Curves of Permanent Magnets	George Pratt		National Relay Conference, Oklahoma State University

Technology Report

MAILING LIST COUPON

☐ ADD

☐ REMOVE

Not available to field offices or outside the U.S.

MAIL COUPON TO 53-077

Name:_____ D.S.: _____

Payroll Code: _____

(Required for the mailing list)

For change of delivery station, use a directory change form.

USER INTERFACE GUIDELINES AVAILABLE

The User Interface Coordination Committee highly recommends *Design Guidelines for the User Interface to Computer-Based Information Systems* to all engineers involved in user interface evaluation.

This 279-page document not only includes a bibliography, glossary, and index; it includes 580 guidelines covering these interface concerns:

- Data entry
- Data display
- Sequence control
- User guidance
- Data transmission
- Data protection

A copy of the Guideline is available in the Wilsonville Library, 685-3986. Committee member Herb Weiner, 685-3586, also has a copy.

The Tektronix User Interface Coordination Committee has been investigating user interface design guidelines for some time. If you are interested in contributing to these efforts, please contact Herb Weiner (63-356, 685-3586). □