An Engagingly Formatted If Thinly Annotated Listing of

FifoSim.ps

CRA:5151.ps

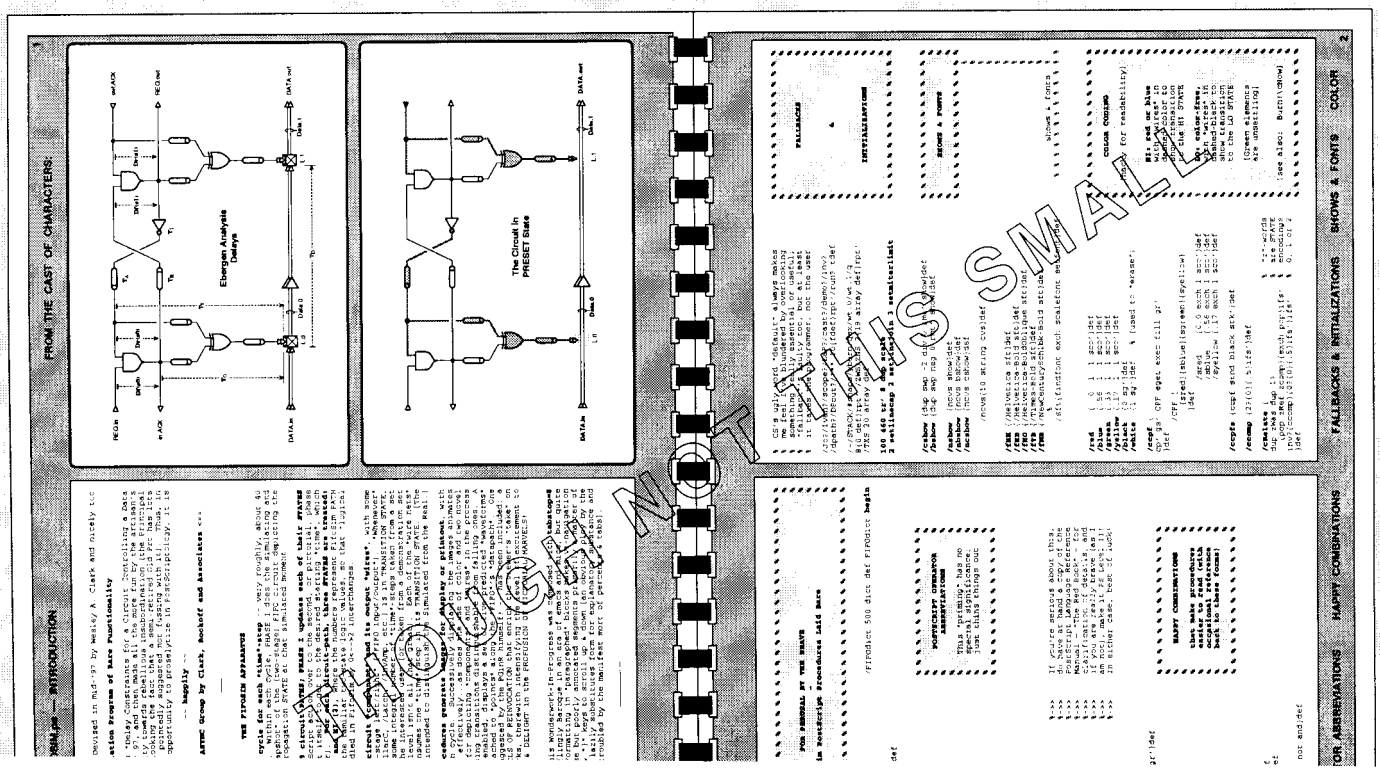
Printed In Convenient HALF-SCALE
for the Intrepid Reader with Normal Vision

and

PROFUSELY ILLUSTRATED!

by Clark, Rockoff and Associates

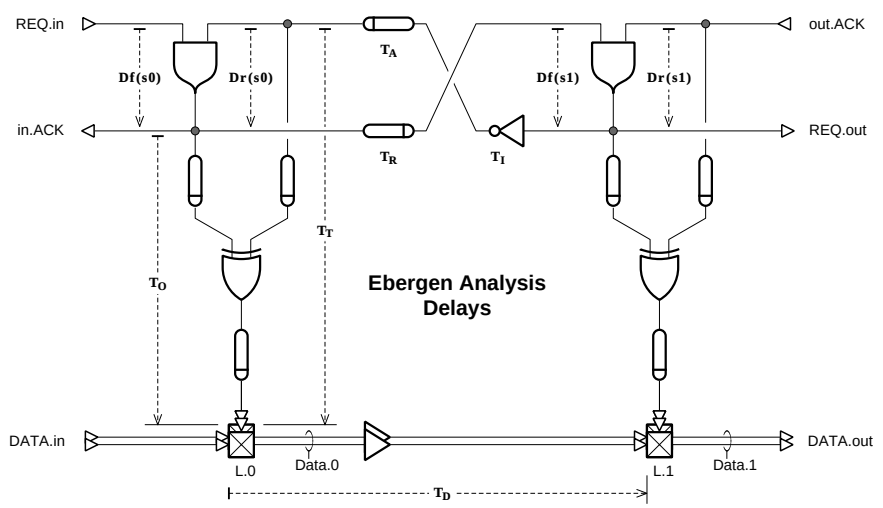
11-9-97 CRA:5152.ps

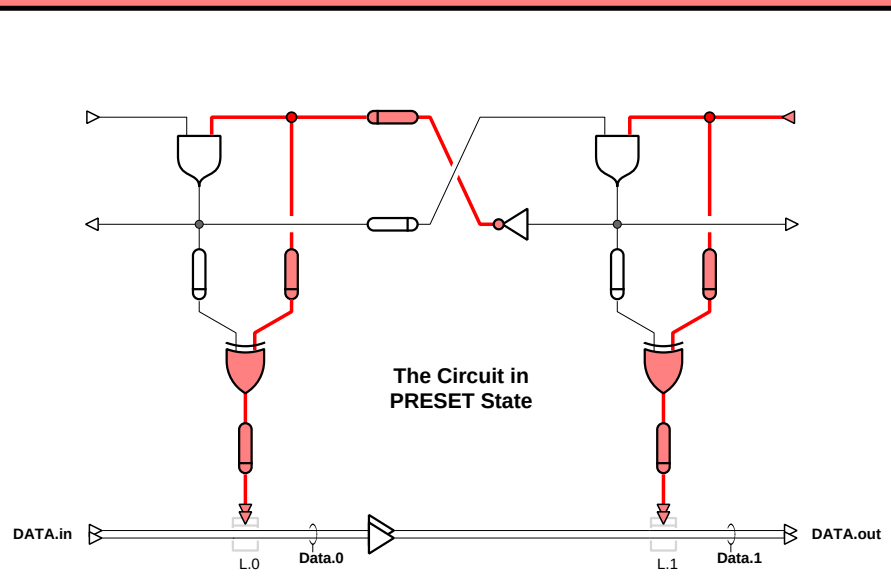


FIFOSIM.ps — INTRODUCTION

%! **FifoSim.ps**|CRA5151.ps Devised in mid-'97 by Wesley A. Clark and nicely too
%
% **A Simulation Program of Rare Functionality**
%
% Inspired by Jo Ebergen's "Delay Constraints for a Circuit Controlling a Data
% Path", SML#97:0155 1 Apr 97, and then made all the more fun by the artisan's
% indulgence of a proclivity towards rebellious insubordination (the Principal
% Guru in Residence, overlooking the fact that a semi-retired Old Pro has lots
% of time for fun & games, pointedly suggested not fussing with it). Thus, in
% anticipation of a bully opportunity to proselytize in PostScriptology, it is
%
% ~~ **happily** ~~
%
% ==> **Donated to SML's ASYNC Group by Clark, Rockoff and Associates** <==
%
% ___
%
% **THE FIFOSIM APPARATUS**
%
% **OPERATES in a two-phase cycle for each *time*-step** -- very roughly, about 40
% picoseconds of realtime. Within each cycle, PHASE I does the simulating and
% PHASE II produces a "snapshot" of the (two-stage) FIFO circuit depicting the
% circuit in its signal-propagation STATE at that simulated moment.
%
% **Each phase deals with 19 circuit-PATHS; PHASE I updates each of their STATES**
% and then turns the PostScript action over to the second, or pictorial, phase
% (unless it has to repeat itself to get to the desired starting *time*, which
% is specified by the user). **For each circuit-path, three STATES are treated:**
% **LO (0), TRANSITION (1), and HI (2)**, where the numbers represent FifoSim PATH
% encodings, rather than the familiar two-state-logic values, so that "logical
% complement" must be handled in FifoSim by 0<-->2 interchanges.
%
% **A PATH is defined as a circuit *component* and its output *wires***, with some
% *components* being "off-stage left/right" (*FIFO input/output*). *Whenever*
% a *component* [see: /MullerC, /Latch, /InvAmp, etc.] is in TRANSITION STATE,
% its *output* stalls by some integral number of *time*-steps taken from a set
% of values supplied by the interested user (or taken from a demonstration set
% if the user's interest level isn't all that great). Each of the *wire nets*
% between *components* consumes one *time*-step in its TRANSITION STATE. [The
% annoying asterisks are intended to distinguish the Simulated from the Real.]
%
% **PHASE II PostScript-procedures generate images for display or printout**, with
% one image per simulation cycle. Successively displaying the images animates
% the *FIFO action* quite effectively...as does the use of color and two novel
% visualization features for depicting *components* and *wires* in the process
% of transition, with rising transitions distinguishable from falling ones. A
% simulated 'scope, when enabled, displays a set of five predicted *waveforms*
% taken from *probes* attached to *points* along the *FIFO*'s *datapath*. One
% final enhancement -- suggested by the PGinR himself! -- has been included: a
% pair of clock-like WHEELS OF REINVOCATION that enrich the viewer's "take" on
% how the whole thing works, therewith intensifying the level of excitement to
% one of GREAT AMAZEMENT & DELIGHT in the PROFUSION OF PICTORIAL MARVELS!
%
%
% **NOTE:** The script of this Wonderwork-In-Progress was composed with **tabstop=8**
%   in Unix/vi (bafflingly Baroque in an era of emacs and mice, but quite
%   serviceable). Formatting in "paragraphed" blocks makes vi-navigation
%   through its dense but poorly annotated segments primarily a matter of
%   using vi's "{" & "}" keys to scroll up & down [an obvious ploy by the
%   programmer, who lazily substitutes form for explanatory substance and
%   is evidently untroubled by the manifest mort of percents & tabs].
%
% _____

FROM THE CAST OF CHARACTERS:

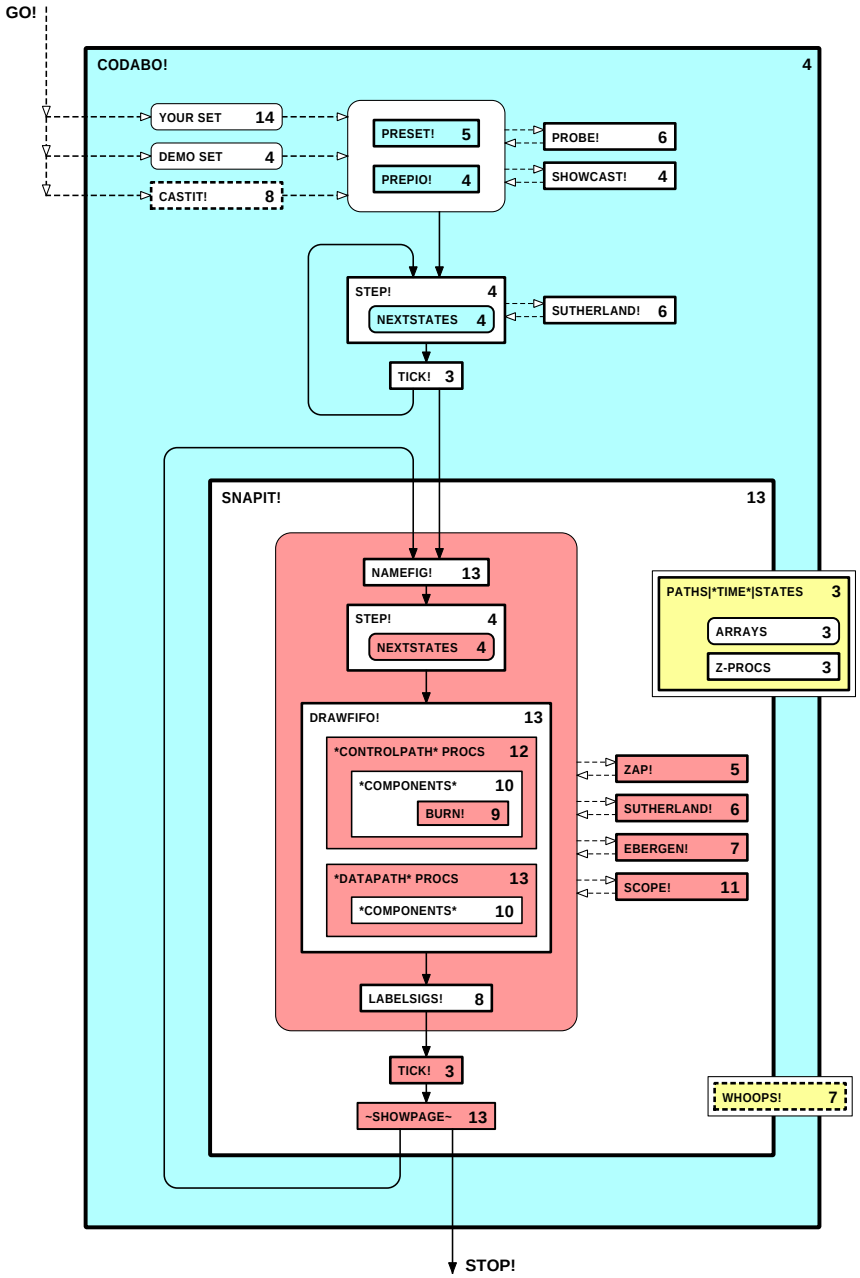




```
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%                                                                             %
%                                                                             %
%                          FOR PERUSAL BY THE BRAVE                           %
%                                     ~                                       %
%                   FifoSim PostScript Procedures Laid Bare                   %
%                                                                             %
%                                                                             %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

                                          /FIFOdict 500 dict def FIFOdict begin
/ldef {load def}def
         %
/clw’ /currentlinewidth ldef
/cp’  /closepath ldef
/cpt’ /currentpoint ldef
/gr’  /grestore ldef                      % % % % % % % % % % % % % % %
/gs’  /gsave ldef                         %                           %
/ifs’ /ifelse ldef                        %   POSTSCRIPT OPERATOR     %
/ix’  /index ldef                         %      ABBREVIATIONS        %
/lnt’ /lineto ldef                        %                           %
/mt’  /moveto ldef                        %  This "priming" has no    %
/np’  /newpath ldef                       %  special significance;    %
/rlt’ /rlineto ldef                       %  just thins things out    %
/rmt’ /rmoveto ldef                       %                           %
/rot’ /rotate ldef                        % % % % % % % % % % % % % % %
/rpt’ /repeat ldef
/sco’ /sethsbcolor ldef
/sd’  /setdash ldef
/sg’  /setgray ldef
/slc’ /setlinecap ldef
/slw’ /setlinewidth ldef
/stk’ /stroke ldef               %->>     If you’re serious about this,
/tr’  /translate ldef            %->>     do have at hand a copy of the
                                 %->>     PostScript Language Reference
                                 %->>     Manual – "The Red Book" – for
                                 %->>     clarification of details; and
/0?   {0 eq}def                  %->>     if you are really brave (as I
/1?   {1 eq}def                  %->>     am not), make it PS Level II!
/2?   {2 eq}def                  %->>     In either case, best of luck!
/adef {add def}def
/apop {aload pop}def
/b3r  {3 -1 roll}def
/cdef {1 ix’ def}def
/cpf  {cp’ gs’ sg’ fill gr’}def
/cpfs {cpf stnd stk’}def                  % % % % % % % % % % % % % % %
/cpts {cpt’ stk’}def                      %                           %
/cvxn {cvx cvn}def                        %    HAPPY COMBINATIONS     %
/edef {exch def}def                       %                           %
/edup {exch dup}def                       %   that make procedures    %
/eget {exch get}def                       %   easier to read (with    %
/eput {eget put}def                       %   occasional reference    %
/f3r  {3 +1 roll}def                      %   back to these forms)    %
/fdef {false def}def                      %                           %
/lnts {lnt’ stk’}def                      % % % % % % % % % % % % % % %
/npop {{pop}rpt’}def
/rlts {rlt’ stk’}def
/rput {b3r put}def
/stnd {[]0 sd’ 2 slc’}def
/swp  {stringwidth pop}def
/tdef {true def}def
/trot {tr’ rot’}def
/xor? {2 copy or f3r and not and}def
```

```
%  CS’s ugly word "default" always makes          % % % % % % % % % % % % % % %
%  me feel I’ve blundered by overlooking          %                           %
%  something really essential or useful;          %                           %
%  "fallback"’s faulty too, but at least          %        FALLBACKS          %
%  it taxes the programmer, not the user          %                           %
                                                  %                           %
/Jo?/Ivan?/scope?/ZaP?/cast?/demo?/inv?           %                           %
/dpath?/DPcut?/*** 10{fdef}rpt’/run? tdef         %            &              %
                                                  %                           %
/~/STACK/#snaps/yburn/qsx/wt.0/wt.1/q             %                           %
8{0 def}rpt’ /ZWS/ZNS 2{19 array def}rpt’         %                           %
/TXS 20 array def                                 %      INITIALIZATIONS      %
                                                  %                           %
100 460 tr’ 8 dup scale                           %                           %
2 setlinecap 2 setlinejoin 3 setmiterlimit        % % % % % % % % % % % % % % %


/cshow {dup swp -2 div 0 rmt’ show}def            % % % % % % % % % % % % % % %
/bshow {dup swp neg 0 rmt’ show}def               %                           %
                                                  %      SHOWS & FONTS        %
/nshow  {ncvs show}def                            %                           %
/nbshow {ncvs bshow}def                           % % % % % % % % % % % % % % %
/ncshow {ncvs cshow}def                                                    %
                                                                           %
      /ncvs{10 string cvs}def                                              %
                                                                           %
                                                                           %
/fHH {/Helvetica sft}def                                                   %
/fHB {/Helvetica-Bold sft}def                                              %
/fHO {/Helvetica-BoldOblique sft}def                                       %
/fTB {/Times-Bold sft}def                                                  %
/fNB {/NewCenturySchlbk-Bold sft}def                                       %
        %                                                  shows & fonts   %
      /sft{findfont exch scalefont setfont}def        % % % % % % % % % % % % %


/red    {  0  1  1 sco’}def                       % % % % % % % % % % % % % % %
/blue   {.56  1  1 sco’}def                       %                           %
/green  {.33  1  1 sco’}def                       %       COLOR CODING        %
/yellow {.17  1  1 sco’}def                       %                           %
/black  {0 sg’}def                                %  (hacky for readability)  %
/white  {1 sg’}def   % (used to "erase")          %                           %
                                                  %                           %
/ccpf {                                           %      HI: red or blue      %
cp’ gs’ CPF eget exec fill gr’                    %      with *wires* in      %
}def                                              %      dashed-color to      %
      /CPF [                                      %      show transition      %
         {sred}{sblue}{sgreen}{syellow}           %      to the HI STATE      %
        ]def                                      %                           %
                /sred    {0.0 exch 1 sco’}def     %      LO: color-free,      %
                /sblue   {.56 exch 1 sco’}def     %      with *wires* in      %
                /syellow {.17 exch 1 sco’}def     %      dashed-black to      %
                                                  %      show transition      %
/ccpfs {ccpf stnd black stk’}def                  %      to the LO STATE      %
                                                  %                           %
/ccomp {2?{0}{.5}ifs’}def                         %      [Green elements      %
                                                  %      are unsettling]      %
/cRelate {                          %  "z"-words  %                           %
dup zWas dup 1?                     %  are STATE  %                           %
  {pop zRef zcomp}{exch pop}ifs’    %  encodings  %  [see also:  Burn!\cNow]  %
inv?{ccomp}{0?{0}{.5}ifs’}ifs’      %  0, 1 or 2  %                           %
}def                                              % % % % % % % % % % % % % % %
```

```
%  Wherein quantized time is captured in          % % % % % % % % % % % % % % %
%  one PostScript array, while the STATE          %                           %
%  of each circuit-PATH is held in other          %    PATHS|*TIME*|STATES    %
%  arrays summarizing the *FIFO*'s STATE          %                           %
%  as of *NOW* (the current *time*-step)          %    Conveniently Arrayed   %
%  as well as at other important *times*          %                           %
%  (the preceding and "reference" steps)          % % % % % % % % % % % % % % %
                                                                              %
                                                                              %
   %            ---  ARRAY INDEXES (#~) ---                                  %
/#a  0 def /#b  1 def /#d  2 def /#e  3 def /#c  4 def     % Unfortunate     %
/#f  5 def /#g  6 def /#h  7 def /#i  8 def /#j  9 def     % cde --> dec     %
/#k 10 def /#l 11 def /#m 12 def /#n 13 def /#o 14 def     % in indicies     %
/#p 15 def /#q 16 def /#r 17 def /#s 18 def /#t 19 def     % & all PATHS     %
                                                                              %
                                                                              %
                                                                              %
   %             ---  PATH NAMES (note the misordering ) ---                  %
/PATHS [(A)(B)(D)(E)(C)(F)(G)(H)(I)(J)(K)(L)(M)(N)(O)(P)(Q)(R)(S)]def         %
                                                                              %
                                                                              %
                                                                              %
   %             ---  TIMEX ARRAY  ---                                       %
   %     A B D E C F G H I J K L M N O P Q R S (T)  <--- For these PATHS:     %
/TIMEX [ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~  0 ]def  %  delay-counters     %
                                                      %  [NULLPATH (T):     %
                                                      %  FifoSim*TIME*]      %
                                                                              %
                                                                              %
/tTime {TIMEX eget}def    % *Time* (or remaining delay-*time*) for #~          %
/Time  {#t tTime neg}def  % (*Time* runs backwards in the TIMEX array)        %
                                                                              %
/Xt?   {tTime 0?}def      % Xition-time?; has PATH #~'s delay run out?        %
/burn? {tTime 0 ge}def    % Is circuit-*COMPONENT* #~ *NOW* in Xition?         %
                                                                              %
                                                                              %
                                                                              %
   %             ---  STATE ARRAYS  ---                                      %
   %     A B D E C F G H I J K L M N O P Q R S   <--- For these PATHS:       %
/ZPRE  [ 0 0 0 2 0 2 2 0 2 2 2 0 2 2 2 0 0 0 0 ]def  %  the PRESET STATES    %
/ZNOW  [ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ]def  %  *time*  t  STATES    %
/ZWAS  [ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ]def  %  *time* t-1 STATES    %
/ZREF  [ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ]def  %  Xition-Ref STATES   %
                                                                              %
                                                                              %
/zcomp {0?{2}{0}ifs'}def       % (FifoSim's "logical complement")              %
                                                                              %
/zRef {ZREF eget}def                                                          %
                                                                              %
/zWas {ZWAS eget}def                    % Z-PROCS                            %
/zNow {dup Xt?{zXtn}{zNew}ifs'}def                                            %
/zNew {dup zWas 1?{zRef zcomp}{zWas}ifs'}def                                  %
/zXtn {ZREF edup zWas put 1}def                                               %
   %                                                  paths|*time*|states     %
/rising? {# zRef 0?}def   % (rising Xition?)          % % % % % % % % % % % % % %


/Tick! {        % <== CODABO! [dn             % % % % % % % % % % % % % % % %
/ZWAS ZNOW def  %   & Probe!  [dn             % I                           %
0 1 19          %   & Snapit! [dn             %              TICK!          %
 {TIMEX exch                                  %                             %
  2 copy get 1 sub put                        %  advances *time* by 1 step  %
 }for                                         %                             %
}def                                          % % % % % % % % % % % % % % % %
```



```
/CODABO! {      % <== YOU!!                    % % % % % % % % % % % % % % % %
[1] Preset!                                    % I II                          %
demo?{DemoSet}if  PrepIO!                      %            CODABO!            %
1 add /n2 edef /n1 cdef n2 add /nf edef        %                               %
scope?{Probe!}if  cast?{ShowCast!}if           %  COunt Down And Blast Off!  %
[0 2] Preset!                                  %  (Ancient TX-2 expression)  %
n1{Step! Ivan?{0 Sutherland!}if Tick!}rpt'     %                               %
n2{run?{Snapit!}{exit}ifs'}rpt'                % % % % % % % % % % % % % % % %
}def                                                                           %
      /run? {#snaps nf le}def                                                  %
                                                                               %
      /DemoSet {                                                               %
        /SrcL 0 def /SnkL 0 def /Lag 5 def /ZaP? fdef /Ivan? tdef              %
        /scope? tdef /Jo? fdef /cast? tdef pop pop 0 100                       %
        0 1 18{dup DEMDELS eget DELAYS f3r put}for                             %
        }def                                                                   %
              /DEMDELS [0 8 8 3 7 2 2 5 4 2 2 5 4 2 0 0 0 7 0]def              %
                                                                               %
                                                                               %
      /ShowCast! {                                                             %
        COC{exec /q edef Snapit!}forall /cast? fdef clear                      %
        }def                                                                   %
              /COC [    % (Cast of Characters)              Cast pic:        %
                 {[(The 2-Stage)(FIFO Circuit)] 1}             % #1          %
                 {[(Ebergen Analysis)(Delays)]Ebergen! 1}      % #2 <==      %
                 {scope?{cScope}if                             % #3          %
                  [(The Circuit in)(PRESET State)][2]Preset! 0                 %
                 }                                                             %
                ]def                                                           %
                     /cScope {                                                 %
                       9 dict begin                                            %
                        /Time 0 def /run? tdef                                 %
                        /T0/T1/T2/T3/T4 5{[54.25]def}rpt'                      %
                        /ds .25 def                                            %
                        0 Scope!                                               %
                       end                                                     %
                       }def                                                    %
                                                                               %
  %+   dn] Probe! + Preset! + Step! + Ebergen!     [dn              codabo!    %
  %+   dn] Snapit! + Tick! + Sutherland! + PrepIO! [dn   % % % % % % % % % % %
                                                                               %
                                                                               %
                                                                               %
/PrepIO! {          % <== CODABO! [up           % % % % % % % % % % % % % % % %
#b ?DEL /Dff edef                               %                               %
#d ?DEL /Drr edef                               %            PREPIO!            %
[#c #e #f][Drr Dff SrcL SnkL] Delsum            %                               %
/Cyc edef  /2Cyc Cyc 2 mul def                  % % % % % % % % % % % % % % % %
/dang -180 Cyc div def                                                         %
/req.1 Dff #c ?DEL adef                                                        %
/ack.1 req.1 Dff adef                                                          %
[][Cyc Dff neg SrcL] #a Prime   % (REQ.in)   <---  Jo: Seems to me these      %
[][Cyc Dff neg SnkL] #o Prime   % (out.ACK)  <---  *delay* specs require      %
[#a][Lag neg] #p Prime          % (DATA.in)  <---  Dutch re-formulation!      %
}def                                                                           %
      /?DEL {DELAYS eget}def                                                   %
                                                                               %
      /Prime {/#* edef Delsum DELAYS #* rput}def                               %
                                                                               %
      /Delsum {                                                                %
        0 exch {add}forall exch                                                %
        dup length 0 ne                                                        %
          {{?DEL add}forall}{pop}ifs'   %                          prepio!    %
        }def                                          % % % % % % % % % % % % % %
```

```
                % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
                %                                                             %
                %      Higgeldy-piggeldy Ivan E. Sutherland                   %
        %       %      — Principal Guru in Residence — said                   %       %
      % % %     %          Unhelpful excursions, these                        %     % % %
        %       %          Playful subversions! Please                        %       %
                %      Limit your effort to theory instead!                   %
                %                                                             %
                % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

/Step! {        % <== CODABO! [up               % % % % % % % % % % % % % % % %
[NEXTSTATES     %   & Probe!  [dn               % I                           %
 {exec}forall   %   & Snapit! [dn               %            STEP!            %
] /ZNOW edef    %   & Sutherland!\\Phmark [dn   %                             %
[#b #d]                                         %      generates the next     %
 {?MCD}forall  /#snaps #snaps 1 adef            %      set of PATH STATES     %
}def                                            %      from the *present*     %
      /NEXTSTATES [                             %      set of PATH STATES     %
        {#a zNow}                               %                             %
        {#b [#p #h #c #a] zBD}                  % % % % % % % % % % % % % % % %
        {#d [#e #l #o] zBD}                                                    %
        {#e #f zDEL}                                                           %
        {#c zNow}                                                              %
        {#f #g zDEL}          %  These procs                                   %
        {#g #i zDEL}          %  do the work                                   %
        {#h #i zDEL}          %  of PHASE I!                                   %
        {#i #j zDEL}          %                                               %
        {#j zNow}             %  Not much to                                  %
        {#k #m zDEL}          %  it, really,                                  %
        {#l #m zDEL}          %      --                                       %
        {#m #n zDEL}          %  ... and if you believe that,                 %
        {#n zNow}             %  then there's this Bridge ...                 %
        {#o #k zDEL}                                                           %
        {#p zNow /zP cdef}                                                     %
        {#q qXpass}                                                            %
        {#r zNow /zR cdef}                                                     %
        {#s sXpass}                                                            %
        ]def                                                                   %
              /setDel {TIMEX edup ?DEL put}def                                 %
                                                                               %
              /zBD  {1 ix' Xt?{{setDel}forall zXtn}{pop zNew}ifs'}def          %
              /zDEL {1 ix' Xt?{setDel zXtn}{pop zNew}ifs'}def                  %
                                                                               %
              /qXpass {                                                        %
                zNow /zQ edef #j zNow 2?                                       %
                 {#p Xt?{#r setDel}if zP}                                      %
                 {#j Xt? #j zWas 0? and zP zQ ne                               %
                   and{#r setDel TIMEX #q 0 put 1}{zQ}ifs'                     %
                 }ifs'                                                         %
                }def                                                           %
                                                                               %
              /sXpass {                                                        %
                pop #n zNow 2?                                                 %
                  {#r zNow}                                                    %
                  {#n Xt? #n zWas 0? and #r zNow #s zNow ne and                %
                    {TIMEX #s 0 put 1}{#s zNow}ifs'                            %
                  }ifs'                                                        %
                }def                                                           %
                                                                               %
    /?MCD {FireMC?{TIMEX DELAYS 2{#* Dxx put}rpt'}if}def                       %
  %                                                                    step!   %
  %+  FireMC? [dn                                  % % % % % % % % % % % % % % %
```

CODABO! **COC [Cast Of Characters]** **PREPIO!** **STEP!**

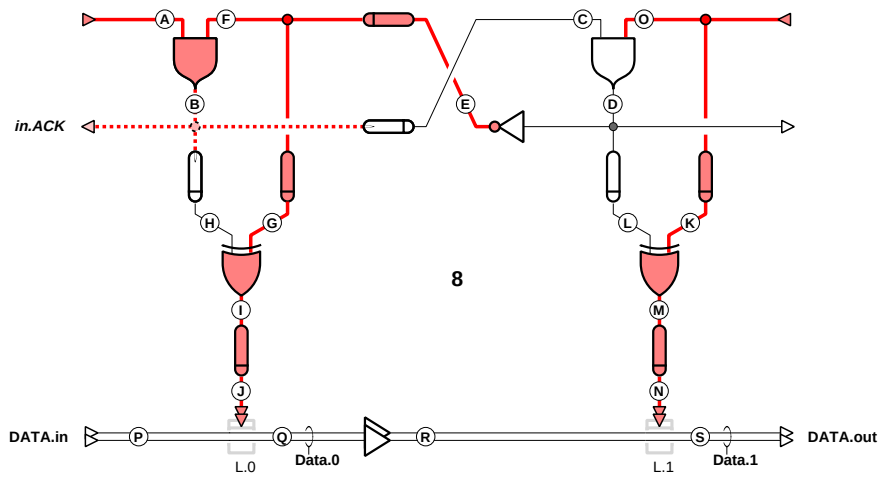
ZAP! — A Debugging Tool

```
/ZAP! {         % <== Snapit!\?Options [dn      % % % % % % % % % % % % % % % %
gs'                                             %                               %
scope?{-24}{-4}ifs' -4 exch tr'                 %             ZAP!              %
black 1.11 fHB 0 1 18{Ptag}for 1.2 fHB          %                               %
SAA {exec PZshow 0 -2 tr'}forall                %      (A debugging tool)       %
gr'                                             %                               %
}def                                            %    puts an ID tag on each     %
      /SAA [                                    %    PATH, and displays the     %
         {(PATHS) 1}                            %    STATE and TIMEX arrays     %
         {(ZREF)  0}{(ZWAS)  0}                 %                               %
         {(ZNOW)  0}{(TIMEX) 0}                 % % % % % % % % % % % % % % % % %
        ]def                                                                    %
                                                                                %
      /PZshow {                                                                 %
        gs' 5 0 tr' pzsetup ?bracket                                            %
        0 1 18{SA eget 0 0 mt' proc 2.6 0 tr'}for                               %
        gr'                                                                     %
        }def                                                                    %
              /pzsetup {                                                        %
                /pz cdef 1?                                                     %
                   {{}{cshow} 3.6}{{dup}{Zshow} 4}ifs'                          %
                -9.2 tr'/proc edef exec cvxn /SA edef                           %
                }def                                                            %
                      /Zshow {                                                  %
                        dup 0 lt                                                %
                          {pop -.75 0 rmt'(~)show .75 0 rmt'}                   %
                          {nbshow}ifs'                                          %
                        }def                                                    %
                                                                                %
              /?bracket {                                                       %
                pz 0?                                                           %
                  {-3.6 -.05 mt' bshow                                          %
                   -2.7 .1 mt'([)show 48.5 .1 mt'(])show                        %
                  }if                                                           %
                }def                                                            %
                                                                                %
      /Ptag {                                                                   %
        dup PXY eget apop                                                       %
        gs' scope?{21 add}if tr' .07 slw'                                       %
        .74 1.48 1.48 -.77 -.39 1 0 Box!                                        %
        -.04 -.06 mt' PATHS eget cshow                                          %
        gr'                                                                     %
        }def                                                                    %
              /PXY [                                                            %
                 [ 7.6  35.8] [10.0  29.0] [43.5  29.0]                         %
                 [31.7  29.0] [41.1  35.8] [12.5  35.8]                         %
                 [16.2  19.5] [11.2  19.5] [13.7  12.5]                         %
                 [13.7   6.0] [49.7  19.5] [44.7  19.5]                         %
                 [47.2  12.5] [47.2   6.0] [46.0  35.8]                         %
                 [ 5.5  2.35] [17.0  2.35]                                      %
                 [28.5  2.35] [50.5  2.35]      %                       zap!    %
                ]def                                    % % % % % % % % % % % % % %


                      % % % % % % % % % % % % % % % % %
                      %                               %
        %             %     Bill Coates Jon Lexau     %           %
      % % %           %     & Jon coates him back     %         % % %
      % % %           %     In issues complex how     %         % % %
        %             %     Does one zap an ACK?!     %           %
                      %                               %
                      % % % % % % % % % % % % % % % % %
```

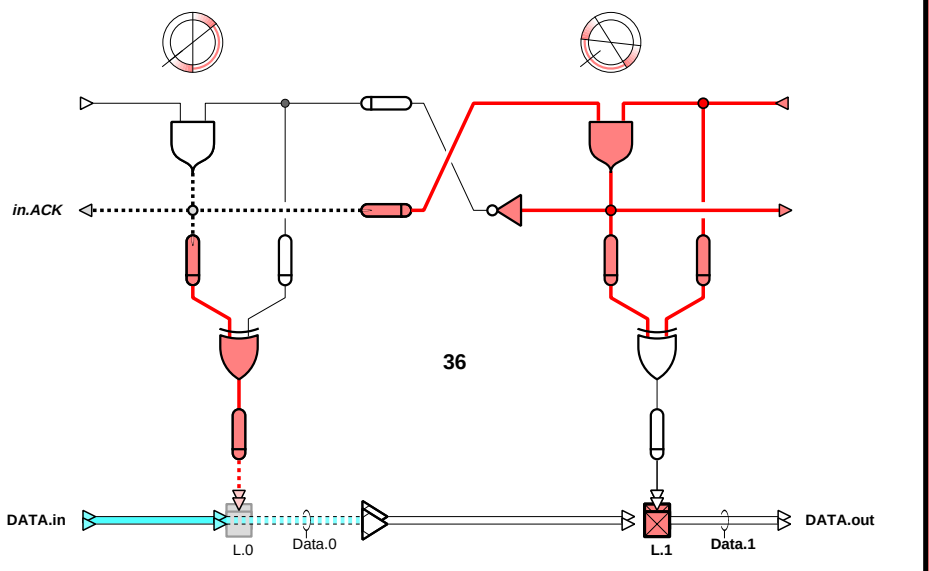
PRESET!

```
/Preset! {      % <== CODABO! [up                % % % % % % % % % % % % % % % %
 {PRST eget     %   & Probe! [dn                 %                               %
  exec                                          %            PRESET!            %
 }forall                                        %                               %
}def                                            %   primes TIMEX & Z-arrays     %
      /PRST [                                   %                               %
        {[0 {Dff}exec 18{-1}rpt']    % 0        % % % % % % % % % % % % % % % % %
         /TIMEX edef                                                            %
        }                                                                       %
        {[20{-1}rpt'] /TIMEX edef}   % 1                                        %
        {0 1 18{ZNOW preset}for      % 2                                        %
         0 1 18{ZWAS preset}for                                                 %
         0 1 18{ZREF preset}for                                                 %
         /#snaps 1 def                                                          %
        }                                                                       %
        ]def     %                                                   preset!    %
              /preset {edup ZPRE eput}def           % % % % % % % % % % % % % % %
```

THIS SNAP IS ZAPPED!



	A	B	D	E	C	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
ZREF [	0	0	0	2	0	2	2	0	2	2	2	0	2	2	2	0	0	0	0]
ZWAS [	2	0	0	2	0	2	2	0	2	2	2	0	2	2	2	0	0	0	0]
ZNOW [	2	1	0	2	0	2	2	0	2	2	2	0	2	2	2	0	0	0	0]
TIMEX [	20	0	~	~	7	~	~	5	~	~	~	~	~	~	~	20	~	~	~]



THE WHEELS OF REINVOCATION

```
/Probe! {       % <== CODABO! [up
SetupTs
tct{Step! XCAP{apop ?Xtime}forall Tick!}rpt'
/TIMEX TXS def /ZNOW ZNS def /ZWAS ZWS def
}def
      /SetupTs {
        Zsave /big? {nf 101 gt}def
        [0 2] Preset!  big?{.125}{.25}ifs'
        /ds edef /tct Cyc big?{10}{5}ifs'
        mul def /i0/i1/i2/i3/i4 5{0 def}rpt'
        /T0/T1/T2/T3/T4 5{[40{-1}rpt']def}rpt'
        }def
              /Zsave {
                0 1 18{ZWS edup ZWAS eput}for
                0 1 18{ZNS edup ZNOW eput}for
                0 1 19{TXS edup TIMEX eput}for
                }def

      /XCAP [
        [#j (i0) (T0)][#n (i1) (T1)][#p (i2) (T2)]
        [#q (i3) (T3)][#s (i4) (T4)]
        ]def

      /?Xtime {
        cvxn /L* edef dup cvn f3r cvxn /i* edef
        ZNOW eget 1?{L* i* Time 1 sub ds mul 2 mul put 1}{0}ifs' i* adef
        }def
   %
   %+   up] Preset! + Step! + Tick! [up
```

```
% % % % % % % % % % % % % % % % % %
%                                   %
%             PROBE!                %
%                                   %
%    builds *DataPath* and          %
%    *Latch* *ScopeTraces*          %
%                                   %
% % % % % % % % % % % % % % % % % %

%  This one's a real
%  challenge to your
%  grasp of basic PS

%  Pass muster? Feel
%  humble and proud!

                                probe!
% % % % % % % % % % % % % % % % % %
```

PROBE!

```
%  Ivan, with characteristic insight,
%  observed that somehow relating the
%  2 bits representing each MullerC's
%  pair of input-Req/Ack logic levels
%  to "vectors" crossed at 90 degrees
%  would, suitably displayed, enhance
%  the viewer's understanding of what
%  is going on during each FIFO-stage
%  "phase" (passing or latching); the
%  display of WHEELS adjusts his idea
```

```
% % % % % % % % % % % % % % % % % %
% I II                             %
%          SUTHERLAND!              %
%                                   %
%    WHEELS OF REINVOCATION         %
%      show the phases of           %
%      *Muller C-element*           %
%      STATE & TRANSITION           %
%                                   %
% % % % % % % % % % % % % % % % % %
```

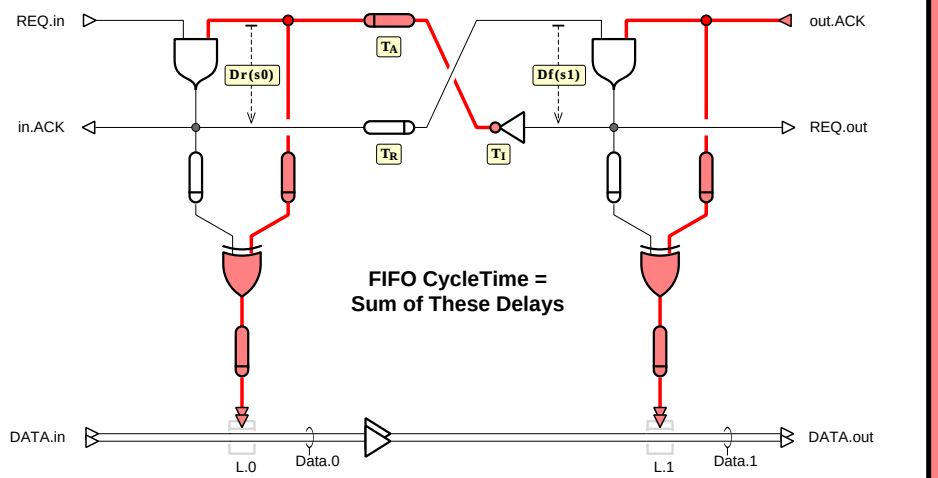
```
/Sutherland!  {  % <== CODABO! [up
0?{false}        %   & Snapit!\?Options [dn
  {true}ifs'
/showit? edef [#b #d]{Wheel}forall
}def
      /Wheel {
        gs' 8 23 tr'  /#~ cdef
        #b eq
          {/wt.0 wt.0 0 Dff -2}
          {/wt.1 wt.1 req.1 ack.1 31.5}ifs'
        14 tr' /ack edef /req edef
        showit?{wheelfig}if  Phmark
        gr'
        }def
              /wheelfig {
                phshade .07 slw'
                1.8 2.4 2{dup 2 mul dup 2 ix' neg dup 1 5 Box!}rpt'
                req ack 2{2.4 2 mul 2.4 neg b3r 1 add Dangit}rpt'
                }def
                      /phshade {
                        gs' 0 slc' .6 slw' req 0 Dangle
                        req ack sub 1 add dup dang mul Dff div
                        dup neg /alpha edef 360 exch sub /beta edef
                        1 sub cvi abs 0 1 b3r 3 copy
                         {burnphase}for
                        redphase 180 rot' 1 -1 scale
                         {burnphase}for
                        gr'
                        }def
                              /burnphase {
                                .5 mul Dff div sred 2.1 0 mt'
                                0 0 2.1 0 beta arcn stk' alpha rot'
                                }def

                              /redphase {
                                gs' .18 slw' 2.1 0 mt'
                                0 0 2.1 0 -180 Dff dang mul sub arcn stk'
                                gr'
                                }def

              /Phmark {
                dup req eq 1 ix' req Cyc add eq or
                  {10 dict begin Step! #~ FireMC?{1}{0}ifs' end}
                  {1}ifs' add 2Cyc mod dup f3r def
                showit?{2 1.8 .7 sub b3r Dangit}{pop}ifs'
                }def

      /Dangit {gs' 1 Dangle 0 mt' 0 rlts gr'}def
      /Dangle {sub dang mul 90 add rot'}def
   %                                          Sutherland!  %
   %+  Step! [up     %+ FireMC? [dn
```

SUTHERLAND! — The Wheels of Reinvocation

```
%  This proc attaches specified delay          % % % % % % % % % % % % % % % %
%  labels to the *FIFO* picture, with          % II                           %
%  or without "boxing" or not at all,          %          EBERGEN!           %
%  based on elements of the EAR array          %                             %
%  or its cousin, the CAR array (used          %    selectively labels the   %
%  only by CODABO!\ShowCast); without          %    Ebergen-analysis PATH-   %
%  the advantage of pre-edited "data"          %    delays [see CAR & EAR]   %
%  to work with, EbShow! really has a          %                             %
%  job-and-a-half to do in PostScript          % % % % % % % % % % % % % % % %
                                                                             %
                                                                             %
/Ebergen! {     % <== CODABO!\ShowCast! [up   & Snapit!\?Options [dn        %
0 1 9                                                                        %
 {dup cast?{CAR}{EAR}ifs' eget                                               %
  dup 9 eq{pop pop}{exch EPROCS eget exec}ifs'                                %
 }for                                                                        %
}def    %    a b c d e f g h i j   <----- UNrelated to PATH names;           %
      /CAR [ 0 0 0 0 0 0 0 0 0 0 ]def   % merely used to match EAR           %
                                                                             %
      /EAR [           % %                                                 %
         9   %a-Dfs0     %     Re LABELS:       The thought here is that    %
         9   %b-Drs0     %                      "dynamic" labeling might    %
         9   %c-Dfs1     %   Changing a "9"     be useful in some future    %
         9   %d-Drs1     %    --  nein! --      variant of FifoSim procs    %
         9   %e-TsubR    %   to "0" creates                                 %
         9   %f-TsubI    %   an open label;             Jo, NOTE:           %
         9   %g-TsubA    %   a "1" displays     What works for EAR works    %
         9   %h-TsubO    %   the label in a     also for CAR; try it out    %
         9   %i-TsubT    %   "boxed" format     on ShowCast!(picture #2)    %
         9   %j-TsubD    %   (for emphasis)     [see (below): CASTIMAGE]    %
        ]def          % %                                                    %
                                                                             %
      /EPROCS [                                                              %
        {Dfs0} {Drs0} {Dfs1} {Drs1} {TsubR}                                  %
        {TsubI}{TsubA}{TsubO}{TsubT}{TsubD}                                  %
        ]def                                                                 %
              /Dfs0  {[(Df)()]  0  1.55  1 EbShow!}def                       %
              /Drs0  {[(Dr)()]  0  10.45 1 EbShow!}def                       %
              /Dfs1  {[(Df)()]  1  35.05 1 EbShow!}def                       %
              /Drs1  {[(Dr)()]  1  43.95 1 EbShow!}def                       %
              /TsubA {[(T)(A)] 21.5 29.3 2 EbShow!}def                       %
              /TsubI {[(T)(I)] 30.3 20.7 2 EbShow!}def                       %
              /TsubR {[(T)(R)] 21.5 20.7 2 EbShow!}def                       %
                                                                             %
              /TsubD {                                                       %
                gs' 0 -1 tr' .04 slw' 42.2 -5.3 mt' 0 2.2 rlts dup           %
                 gs' 0 -4.5 tr' [] 3 42.2 0 33.5 0 0 EbShow! gr'             %
                [(T)(D)] 25.8 -5.2 2 EbShow!                                 %
                gr'                                                          %
                }def                                                         %
                                                                             %
              /TsubO {                                                       %
                [(T)(O)] 2 3 0 23.12 0 0 EbShow!                             %
                .04 slw' 2 0 mt' 5.8 0 rlts                                  %
                }def                                                         %
                                                                             %
              /TsubT {                                                       %
                [(T)(T)] 2 16.4 0 31.7 1 0 EbShow!                           %
                .04 slw' 17.3 0 mt' -5.8 0 rlts                              %
                }def                                                         %
   %                                                      Ebergen!  %
   %+   EbShow! [dn                                % % % % % % % % % % % % % %
                                                                        %
                                                                        %
```

```
                                                                        %
                                                                        %
/EbShow! {      % <== Ebergen!\EPROCS\ [up      % % % % % % % % % % % % % % %
 [                                              % II                    %
  {Ebdisp}                % 0                  %          EBSHOW!      %
  {23.9 31.7 0 Ebdisp}    % 1                  %                       %
  {gs' tr' /ym 0 def      % 2                  %  displays labels fed to it %
   /ebmode 0 def                                %  by the Ebergen! procedure %
   EbLabel                                      %                       %
   gr'                                          %    [Such a hack this is!] %
  }                                             %                       %
 ] eget exec                                    % % % % % % % % % % % % % % %
}def                                                                    %
                                                                        %
      /Ebdisp {    %: bx [(m)(s)] 0|1|2 x yb yt cutln      % [Egad!]    %
        gs'                                                             %
        /cutln edef /yt edef /yb edef 0 tr'                             %
        [[0 1 0][1 1 0][0 0 0][0 0 90]] eget apop                       %
        /ang edef /ebmode edef /stg# edef ebline                        %
        yt yb sub .49 mul yb add .7 sub /ym edef                        %
        ang 0?{0 0 0}{-90 -14.3 14.3}ifs' trot                          %
        EbLabel                                                         %
        gr'                                                             %
        }def                                                            %
              /ebline {                                                 %
                gs' ang rotate .07 slw' black                           %
                .15 slw' -.4 yt mt' .8 0 rlts              %   _        %
                .07 slw' [.295]0 sd' 0 yt mt'              %   .        %
                cutln 1?{0 23.9 lnts 0 23.12}{0 yt}ifs'    %   .        %
                mt' 0 yb lnts []0 sd'                      %   .        %
                -.3 yb .8 add mt' .3 -.8 rlt' .3 .8 rlts   %   v        %
                gr'                                                     %
                }def                                                    %
                                                                        %
      /EbLabel {        %: bx [(m)(s)]                                  %
        exch /bx edef black                                             %
        dup length 0 ne                                                 %
          {apop /sb edef /mm edef 0 1000 ebdelshow                      %
           gs' stnd .2 x -2.2 mul .4 add 1.6 x .3 sub y .5 sub          %
           bx 1?{.2 4}{white 1 0}ifs' Box!                              %
           gr' x y ebdelshow                                            %
          }{pop}ifs'                                                    %
        }def                                                            %
             /ebdelshow {       %: x y                                  %
                /y edef /x edef x y mt' 1.1 fNB mm show .9 fNB           %
                ebmode 0?{sb (I) ne{0 -.1}{.1 0}ifs'}{0 0}ifs'          %
                -.2 rmt' sb show 0 rmt'                                 %
                ebmode 1?                                               %
                  {1.1 fNB .08 .2 rmt'                                  %
                   (\(s) show stg# 1 string cvs show (\)) show          %
                  }if                                                   %
                cpt' pop -2 div /x edef                                 %
                /y ym .4 adef       %                          ebshow!  %
                }def                                % % % % % % % % % % % % % %
```

```
/Whoops! {              %  A place-holder     % % % % % % % % % % % % % % %
%                       %  humbles one to     %                         %
%                       %  remember those     %           WHOOPS!       %
%  WATCH THIS SPACE FOR %  inevitable but     %                         %
%  FURTHER DEVELOPMENTS %  fixable errors     %       To err is human,  %
%                       %  of concept and     %       to patch divine.  %
%                       %  the goofs that     %                         %
}def                    %  plague us all!     % % % % % % % % % % % % % % %
```



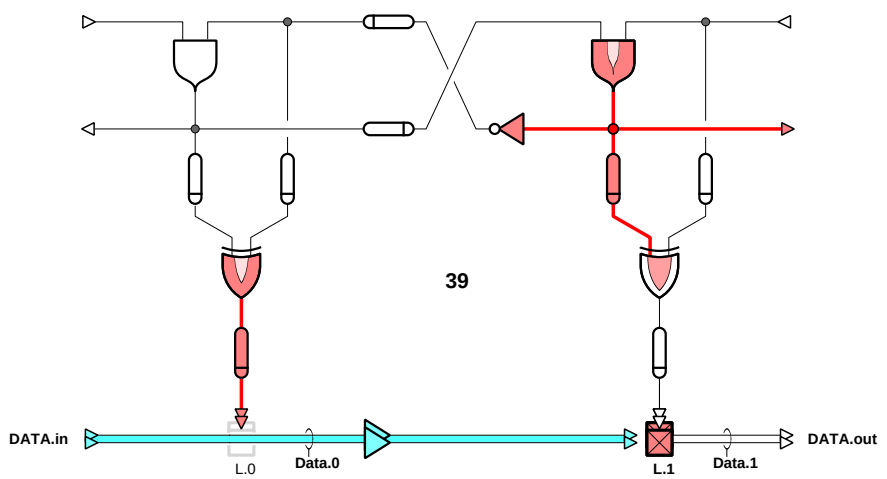
A "CASTIT" EXAMPLE FOR JO

```
%  Initially modifying definitions to          % % % % % % % % % % % % % % % %
%  force FifoSim to "cast" images for          %                               %
%  documents is shown by example: the          %          CASTIMAGE            %
%  procedure below tailors FifoSim to          %                               %
%  the task of generating exactly one          %    For specialized prints     %
%  snapshot specialized to illustrate          %                               %
%  the "Jo?"-option's use of Ebergen!          % % % % % % % % % % % % % % % %
                                                                               %
/CastIt! {                                                                     %
  .58 dup scale       % % % % % % % %                                           %
    %                 %             %                                          %
  /cast? tdef         %  ~EXAMPLE~   %        Required modifications            %
    %                 %             %        are established within            %
  /COC [              % % % % % % % %        the CastIt! proc shown            %
   {[                               %        (and the Modifier then            %
     (FIFO CycleTime =)             %        assumes responsibility            %
     (Sum of These Delays)          %        for the consequences!)            %
   ] Ebergen! 1                     %                                          %
  }                                 %                                          %
  ]def                              %        To set up whatever you            %
    %                               %        are re-defining (procs            %
    %   a b c d e f g h i j         %        arrays, params, etc.),            %
  /CAR [9 1 1 9 1 1 1 9 9 9]def     %        simply kill the "%" in            %
    %                               %        the "%CastIt!" command            %
  /CODABO! {                        %        [see FifoSim.ps's END]            %
  [1]Preset! PrepIO!                %                                          %
  pop pop ShowCast! %     example % %                                          %
  }def                  % % % % % % %                                          %
    %                                                                          %
    %   CastIt! holds the fort until FifoSim is adjusted                       %
    %   to simplify the production of specialized images                       %
    %                                                                castimage %
}def    %+  up] Ebergen! + CODABO!\COC [up          % % % % % % % % % % % % % %
```

```
/LabelSIGS! {       % <== Snapit! [dn          % % % % % % % % % % % % % % % %
/F1 {1.2 fHH}def                               % II                           %
/F2 {1.2 fHB}def black                         %          LABELSIGS!         %
ReqAck                                         %                             %
DPLAB {exec mt' exec show}forall               %    adds STATE-conditioned   %
DataWrap                                       %    REQ, ACK, DATA & LATCH   %
}def                                           %    labels to the pictures   %
      /ReqAck {                                %                             %
        black q 0?{1.2 fHO}{F1}ifs'            % % % % % % % % % % % % % % % %
        CLAB                                                                   %
         {exec Xt? q 1? or                                                     %
           {mt' 0?{show}{bshow}ifs'}                                           %
           {4 npop}ifs'                                                        %
         }forall                                                               %
        }def                                                                   %
              /CLAB [                                                          %
                 {(REQ.in) 1  -4.5  31.7 #a}                                   %
                 {(in.ACK) 1  -4.5  23.2 #b}                                   %
                 {(REQ.out) 0 55.2  23.2 #d}                                   %
                 {(out.ACK) 0 55.2  31.7 #o}                                   %
                ]def                                                           %
                                                                               %
      /DPLAB [                                                                 %
         {(L.0){#j latchlab} 9.2 -4.1}                                         %
         {(L.1){#n latchlab} 42.7 -4.1}                                        %
         {(Data.0){#j [#p #q] datalab} 14 -3.6}                                %
         {(Data.1){#n [#r #s] datalab} 47.5 -3.6}                              %
         {(DATA.in){#j PipeCut? q 1? or{F1}{F2}ifs'} -8.9 -1.7}                %
         {(DATA.out){#n [#r #s] datalab} 55.1 -1.7}                            %
        ]def                                                                   %
              /latchlab {zNow 0? cast? not and{F2}{F1}ifs'}def                 %
                                                                               %
              /datalab {                                                       %
                 {zNow}forall ne                                               %
                exch Xt? and q 1? or{F1}{F2}ifs'                               %
                }def                                                           %
                                                                               %
      /DataWrap {                                                              %
        gs' 15.1 -1.3 tr' .07 slw'                                             %
        2{dwrap 0 -.8 mt' 0 -.6 rlts 33.5 0 tr'}rpt'                           %
        gr'                                                                    %
        }def                                                                   %
              /dwrap {                                                         %
                gs' .4 1 scale                                                 %
                2{.8 0 mt' 0 0 .8 0 140 arc stk'                               %
                  1 -1 scale                                                   %
                 }rpt'                                                         %
                gr'                                                            %
                }def                                                           %
  %                                                          labelsigs!  %
  %+   Pipe\PipeCut? [dn                          % % % % % % % % % % % % % % %
```

AHD! DOT! BOX! BURN!

```
/Ahd! {            % <== dn] A & B\ & D\ [dn     % % % % % % % % % % % % % % %
gs' trot .12 slw'  %   & dn] F\ & O\ & P [dn     % II                         %
stnd 0 0 mt'       %   & dn] S & Pipe.b  [dn     %             AHD!           %
1 .5 rlt'          %   & dn] Latch\ltch  [dn     %                            %
0 -1 rlt' cp'                                    %   One of the two simplest  %
# ZNOW eget rising? not # Xt? and                %   geometrical shapes that  %
 {pop black .86 cpfs}                            %   have to be constructed;  %
 {[0 .2 .5] eget                                 %                            %
  dpath?{1}{0}ifs' ccpfs                         %                            %
 }ifs'                                           %             &              %
gr'                                              %                            %
}def                                             %                            %
                                                 %                            %
/Dot! {            % <== dn] B\ & D\ [dn         %             DOT!           %
gs' tr'            %   & dn] F\ & O\ [dn         %                            %
zz [.32 .37 .4]                                  %     ... the other one.     %
eget dup dup 2 mul dup b3r neg dup               %                            %
zz 1?                                            %                            %
 {rising?                                        %                            %
    {.3 1}{.18 slw' .86 0}ifs'                   %             Note:          %
 }                                               %  It's animation that makes  %
 {zz 0?                                          %  even the simplest figures  %
    {.4 0}{1 1}ifs'                              %  require such complication  %
 }ifs' Box!                                      %  in the drawing procedures  %
gr'                                              %                            %
}def                                             % % % % % % % % % % % % % % %


/Box! {            % <== Dot! [up               % % % % % % % % % % % % % % %
7 1 roll /gf edef  %   & Sutherland!\           % II                         %
gs' tr' /h edef    %           \wheelfig [up     %             BOX!           %
/w edef /r edef    %   & ZAP!\Ptag       [up     %                            %
np' rbx            %   & EbShow!\EbLabel [up     %          Enter with:       %
CFL eget exec      %   & Scope!\Cursor   [dn     %     r w h x y gray mode#   %
gr'                %   & Latch\ltch      [dn     %                            %
}def               %   % Delay [dn  & E [dn     %    Box corners have radius %
      /rbx {                                     %     r (r > 0); a Box with  %
        0 r mt'                                  %      r = w/2 = h/2  is a   %
        0 h w h r arcp                           %    circle [e.g., see Dot!] %
        w h w 0 r arcp                           %                            %
        w 0 0 0 r arcp                           % % % % % % % % % % % % % % %
        0 0 0 h r arcp cp'                                                    %
        }def                                                                  %
            /arcp {arcto 4 npop}def                                           %
                                                                              %
      /CFL [                                                                  %
         {gf cpfs}                    %  Some of this complexity             %
         {0 0 cff}                    %  owes to a hedge against               %
         {1 0 cff}                    %  future variants of Box;               %
         {2 0 cff}                    %  dashed borders? ghosts?               %
         {3 3 cff}                    %  colored...green? (ugh.)               %
         {stk'}                                                               %
        ]def                                                                  %
            /cff {                                                            %
              /cc edef CFF eget exec                                          %
              gf 0?{cpf}{pop gf cc ccpf}ifs' slw'                             %
              exec stk'                                                       %
              }def                                                            %
                  /CFF [                                                      %
                    {{black} .14 .6}                                          %
                    {{black} .18 1}                                           %
                    {{} .25 1}                                                %
                    {{} .07 1}        %                              box!     %
                    ]def                          % % % % % % % % % % % % % % %
```



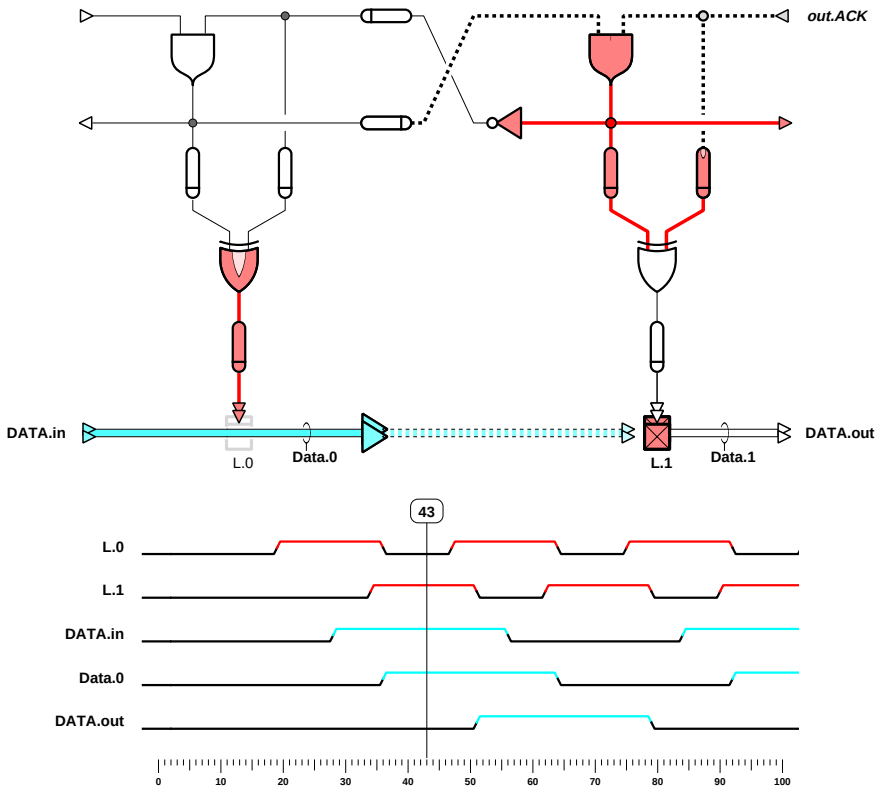
THE *COMPONENTS* ARE BURNING!!

```
/Burn! {     % <== dn] Delay &  InvAmp  [dn     % % % % % % % % % % % % % % %
/#* edef     %   & dn] ITog  &  MullerC [dn     % II                         %
/ybn edef                                        %             BURN!          %
/xbn edef dup f3r                                %                            %
gs' exec clip np' 0 ybn #* flamesize             %     depicts a "fire" in    %
 4 ix' 1?{exch tr'}{tr' scf}ifs'                 %     DELAY, INVAMP, ITOG    %
 exec #* cNow #* #r eq{1}{0}ifs' ccpfs           %     & MULLERC (LATCH is    %
gr' exec cp' stk' pop                            %     not flammable) when    %
}def                                             %     in TRANSITION STATE    %
      /scf {                                     %                            %
        xbn dup 0 tr' 0 slw'                     %     A novel feature and    %
        1 #* tTime .6 add div                    %     gratifyingly tricky    %
        1 scale neg 0 tr'                        %                            %
        }def                                     % % % % % % % % % % % % % % %
                                                                              %
      /flamesize {                                                            %
        dup tTime dup 0 lt{3 npop 0}{exch ?DEL div mul}ifs'                   %
        }def                                                                  %
                                                                              %
      /cNow {                       % <--- This  coloring proc is used      %
        dup tTime /tt cdef 0?             %   only by Burn!; definition       %
          {zWas inv?{zcomp}if ccomp}      %   might have been put under       %
          {cburn}ifs'                     %   COLOR CODING but the proc       %
        }def                              %   is shown here in situ for       %
              /cburn {                    %   the sake of inconsistency       %
                dup ?DEL tt 2 ix'                                             %
                zWas 2?{1 sub}if                                              %
                exch div .5 mul exch zWas inv?{zcomp}if 0?{neg .5 add}if      %
                }def                                                          %
  %                                                                    burn!  %
  %+   PrepIO\?DEL [up                          % % % % % % % % % % % % % % %
```

```
%  These geometric figures (except for          % % % % % % % % % % % % % % % %
%  *Latch*) are displayed as "burning"          % II                          %
%  *whenever* the displayed element is          %     CIRCUIT *COMPONENTS*    %
%  in its TRANSITION (output-stalling)          %                             %
%  STATE; burning enlivens the picture          % % % % % % % % % % % % % % % %

/Delay {        % <== dn] B\ & D\ & E       [dn         % %
cpts            %   & dn] F\ & O\ & I\ & M\ [dn           %
gs' trot                                                  %
-.5 0 tr' stnd .18 slw' /#* edef                          %
black .5 1 4 0 0 #* cRelate 2 Box!                        %     DELAY
#* burn?{0 {rbx} dup .5 -3.2 #* Burn!}if                  %
.18 slw' 0 3.2 mt' 1 0 rlts                               %
gr'                                                       %
}def                                                    % %

/InvAmp {       % <== dn] E & R [dn                     % %
/inv? edef black                                          %
gs' inv?                                                  %
 {0 30.3 23.54}                                           %
 {180 21.6 -1}ifs'                                        %
trot .18 slw' invamp                                      %
cast?                                                     %
 {1 cpfs}                                                 %
 {# cRelate                                               %
  inv?{0}{1}ifs' ccpfs                                    %
 }ifs'                                                    %
# burn?                                                   %
 {1 {iamask}{invamp} 0 2.4 # Burn!}if                     %     INVAMP
/inv? fdef                                                %
gr'                                                       %
}def                                                      %
      /iamask {                                           %
        .07 slw' .4 0 mt' -1.9 1.25 rlt'                  %
        3.5 0 rlt' 0 -2.5 rlt'                            %
        -3.5 0 rlt' cp'                                   %
        }def                                              %
          %                                               %
      /invamp {                                           %
        0 0 mt' 2 -1.25 rlt' 0 2.5 rlt'                   %
        }def                                            % %

/ITog {         % <== dn] I\ & M\ [dn                   % %
gs' -1.5 0 tr'                                            %
.18 slw' stnd itog # cRelate 0 ccpfs                      %
# burn?                                                   %
 {0 {itog} dup 1.5 2.4 # Burn!}if                         %
0 .5 tr' 0 3.54 mt' itop stk'                             %
gr'                                                       %
}def                                                      %     ITOG
      /itog {                                             %
        3 3.54 mt' 0 -.94 rlt'                            %
        0 2.6 3 0 -60 arcn                                %
        3 2.6 3 -120 180 arcn                             %
        0 .94 rlt' itop                                   %
        }def                                              %
                                                          %
      /itop {1.5 7.54 4.27 249.44 290.55 arc}def        % %
```

```
/Latch {        % <== J & N [dn                         % %
gs' -1 -2 tr'                                             %
10 dict begin                                             %
 stg# 0?{#p}{#r}ifs' zNow Pipe                            %
 [0 .6 .84] zl get sg'                                    %
 2{ltch twin}rpt'                                         %
 zl 2?                                                    %
   {gs' white 1 slw' 0 1.9 mt' 2 0 rlts gr'}if            %
end                                                       %
gr'                                                       %
}def                                                      %
      /ltch {                                             %
        .18 slw' stnd .01 2 2 0 0                         %
        cast?                                             %     LATCH
          {1 0}                                           %
          {zl rising? # Xt? and                           %
            {pop .86 0}                                   %
            {[.5 .2 0] eget 3}ifs'                        %
          }ifs' Box!                                      %
        1 90 1 2.1 Ahd!                                   %
        zl 0?                                             %
          {.07 slw'                                       %
           0 0 mt' 2  2 rlts                              %
           0 2 mt' 2 -2 rlts                              %
          }if                                             %
        }def                                            % %

/MullerC {      % <== B & D [dn                         % %
gs' 4.3 3 tr'                                             %
.18 slw' stnd black mcel                                  %
# cRelate 0 ccpfs                                         %
# burn?                                                   %
  {0 {mcel} dup 1.7 3.6 # Burn!}if                        %
gr'                                                       %     MULLERC
}def                                                      %
      /mcel {                                             %
        0 4.08 mt' 0 -1.632 rlt'                          %
        0 .612 1.7 1.224 1.7 0 curveto                    %
        1.7 1.224 3.4 .612 3.4 2.448 curveto              %
        0 1.632 rlt' cp'                                  %
        }def                                            % %

%+   up] Ahd! + cRelate + Box! + Burn! [up            circuit *components*  %
%+   dn] twin + Pipe [dn                            % % % % % % % % % % % % % %

/FireMC? {    % <== Sutherland!\Phmark [up    % % % % % % % % % % % % % % % %
/#* cdef      %   & Step!\?MCD [up            % I II                        %
#b eq                                         %           FIREMC?           %
  {#a #f}{#c #o}ifs' 2 copy                   %                             %
zNxt exch zNxt eq f3r                         %       Have REQ and ACK      %
  Xt? exch Xt? dup{Dff}{Drr}ifs'              %       both checked in?      %
/Dxx edef or and                              %              -              %
}def                                          %    Have their STATES just   %
      /zNxt {                                 %    *now* become the same?   %
        dup Xt?                               %    If so, *time* to fire!   %
          {zRef zcomp}{ZNOW eget}ifs'         %                             %
        }def                                  % % % % % % % % % % % % % % % %
```

SCOPE TRACES



DATAPATH STATES *SCOPE-TRACED*

```
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
%                                                                       %
%                                                                       %
%     There once was a young man named Ian                              %
%     (of Jones) and he just wasn't seein'                              %
%           How ACK followed REQ ...                                    %
%           But then, what the heck,                                    %
%     It was surely the sequence to be in!                              %
%                                                                       %
%     Now, it's not just a matter of fonts                              %
%     And of course every ACK's a response                              %
%           Though never, Ian holds,                                    %
%           Can Italics – or Bolds –                                    %
%     Ever transform an ACKS into GRAHNTS!                              %
%                                                                       %
%                                                                       %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
```

SCOPE!

```
/Scope! {       % <== CODABO!\\\cScope [up
gs' 3 -11 tr'   %   & Snapit! [dn
/gln edef
.18 slw' 1 setlinejoin
 gs' Clip
   LL {Trace 0 -3.5 tr'}forall
 gr' Cursor  tLabel
gr'
}def
      /Clip {
        0 slw' dup ds add -16.3 mt' 51.28 exch sub dup 0 rlt'
        0 21 rlt' neg 0 rlt' cp' clip np' .18 slw'
        }def

      /Trace {
        gs' apop 0?{{red}}{{blue}}ifs'
        /color edef /y 0 def 0 .5 mt' pop exec
          {/Xtime cdef 0 gt{tline}{exit}ifs'}forall
        gr'
        }def
              /tline {
                y 0?{1 black .5 -.5}{0 color -.5 .5}ifs'
                ds exch rlt' Xtime y lnt' ds exch rlt'
                cpt' stk' mt' /y edef
                }def

      /Cursor {
        gs' -2 Clip Time ds mul 2 mul ds sub 2.5 tr'
        run?{.07}{.12}ifs' slw' 0 0 mt' 0 -18.8 rlts
        .6 2.6 1.9 -1.3 0 run?{1}{.96}ifs' 0 Box!
        black 1.18 fHB 0 .5 mt' Time 1 sub ncshow
        gr'
        }def

      /tLabel {
        black 1.2 fHB
        LL {1 get -2.5 .2 mt' bshow
            -1 0 mt' 2.2 0 rlts 0 -3.5 tr'
           }forall
        gln 0?{gs' ghostline gr'}if
        .07 slw' .8 fHB 0 1 big?{200}{100}ifs' {tscale}for
        }def
              /ghostline {
                .07 slw' [.53]0 sd'
                5{-1.1 4.5 mt' 52.5 0 rlts 0 3.5 tr'}rpt'
                }def

              /tscale {
                /i edef i ds mul 2 mul ds add 1 mt'
                i 10 mod 0?
                  {cpt' 2 copy 2 sub mt' i ncshow mt' 0 -1}
                  {0 i 5 mod 0?{-.8}{-.4}ifs'}ifs' rlts
                }def

      /LL [
         [{T0} (L.0)      0]
         [{T1} (L.1)      0]
         [{T2} (DATA.in)  1]
         [{T3} (Data.0)   1]
         [{T4} (DATA.out) 1]
        ]def
  %                                                          scope!  %
  %+   CODABO\run? [up   (see also: PROBE! [up)    % % % % % % % % % % % % % %
```

```
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% II                                                                    %
%                 SCOPE!                                              %
%                                                                       %
%   displays Traces, Cursor                                             %
%   and a tiny *Time*-scale                                             %
%                                                                       %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
```

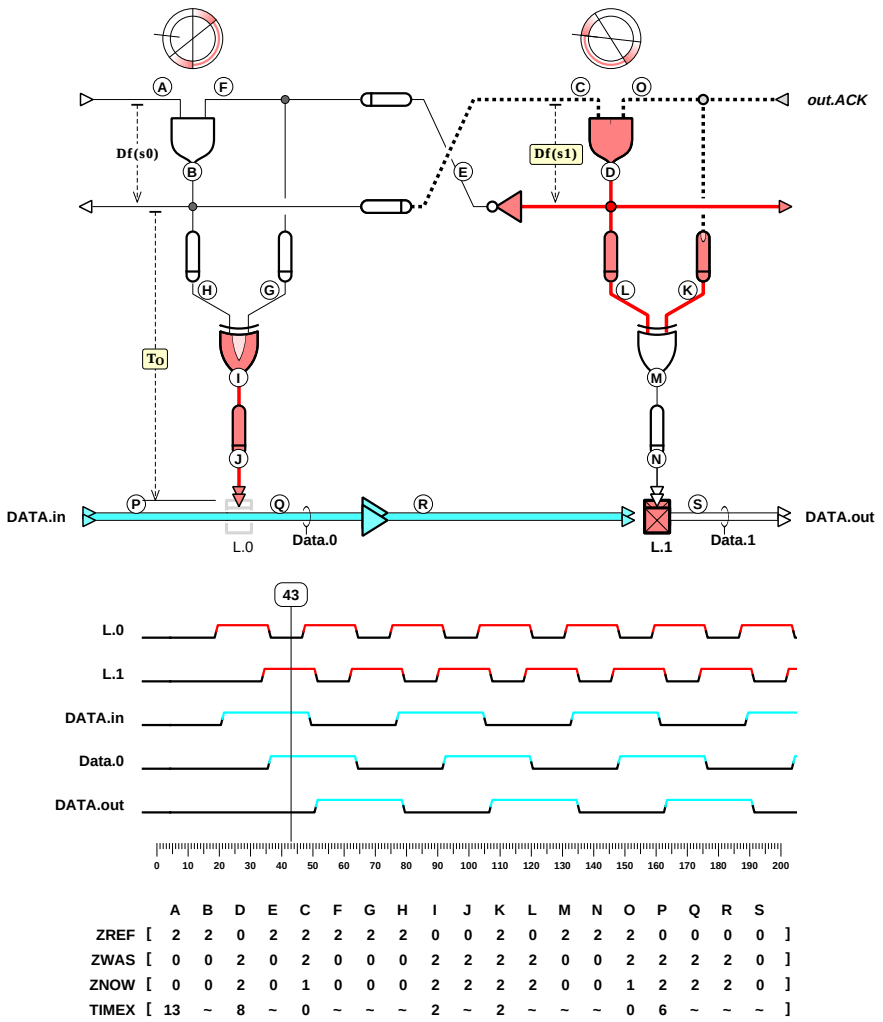
```
                % % % % % % % % % % % % % % % % % % % % % % % % % % % %
        %       %                                                     %       %
      % % %     %       SCOTT FAIRBANKS, CALL YOUR OFFICE!            %     % % %
        %       %                                                     %       %
                % % % % % % % % % % % % % % % % % % % % % % % % % % % %



%  Except for paths A, E and C, these          % % % % % % % % % % % % % % % % % %
%  geometric *FIFO-ControlPath* procs          % II                               %
%  are stg# dependent: "x"-located by          %     *CONTROLPATH*  PROCS        %
                                 %             %                                 %
/?x {/stg# cdef 33.5 mul}def   % <---          % % % % % % % % % % % % % % % % % %
                                                                                  %
                                                                                  %
     %% (TYPICAL)--->   /A {  % <== Snapit!\DrawFIFO! [dn                        %
                                                                                  %
                                                                                  %
/A {                                                                  % %         %
[.305]CPdash -2 32.12 mt' 7 0 rlt'                                      %    A    %
0 -1.4 rlts stnd 1 180 -2 32.12 Ahd!                                    %         %
}def                                                                  % %         %
                                                                                  %
                                                                                  %
/B {0 BDpath}def                                                      % %         %
/D {1 BDpath}def                                                        %         %
                                                                        %         %
      /BDpath {                                                         %    B    %
        gs' ?x 26.54 tr' [.287]CPdash 6 0 mt' 0 -5 rlt'                 %         %
        stg# 0?{#h}{#l}ifs' 180 Delay 0 -3 tr' -2 0 mt'                 %         %
        stg# 0?{19.5}{20}ifs' 0 lnt'                                    %    D    %
        stg# 0?{#c -90 Delay 1 0 -3.1}{stk' 1 180 20.5}ifs'             %         %
        0 Ahd!   MullerC   6 0 Dot!                                     %         %
        gr' stnd                                                        %         %
        }def                                                            % %       %
                                                                                  %
                                                                                  %
/E {                                                                  % %         %
[.3]CPdash 29.9 23.54 mt' -1.4 0 rlt' 24.5 32.12 lnt'                   %         %
-1 0 rlt' #f 90 Delay gs' white .8 slw' cutC gr' true InvAmp            %    E    %
.4 .8 .8 29.6 23.14 zz 1?{.2}{# cRelate}ifs' 2 Box!                     %         %
}def                                                                    %         %
      /cutC {26 26.7 mt' 66 rot' 2 0 rlts}def                       % %         %
                                                                                  %
                                                                                  %
/C {                                                                  % %         %
gs' 1 0 tr' [.3]CPdash 0 slc' 37.5 30.72 mt'                            %         %
0 1.4 rlt' -10 0 rlt' 23.5 23.54 lnt' -1 0 rlts                         %    C    %
gr'                                                                     %         %
}def                                                                  % %         %
                                                                                  %
                                                                                  %
/F {0 FOpath}def                                                      % %         %
/O {1 FOpath 1 0 52.7 32.12 Ahd!}def                                    %         %
                                                                        %         %
      /FOpath {                                                         %    F    %
        gs' ?x 32.12 tr' [.3]CPdash 0 slc' 7 -1.4 mt'                   %         %
        0 1.4 rlt' stg# 0?{19.4}{20}ifs' 0 lnts                         %         %
        13.4 0 mt' 0 -7.92 rlts 13.4 -9.22 mt' 0 -1.36 rlt'             %    O    %
        stg# 0?{#g}{#k}ifs' 180 Delay  13.4 0 Dot!                      %         %
        gr'                                                             %         %
        }def                                                            % %       %
                                                                                  %
                                                                                  %
                                                                                  %
```

```
                                                                                  %
                                                                                  %
/G {0 GKpath}def                                                      % %         %
/K {1 GKpath}def                                                        %         %
                                                                        %    G    %
      /GKpath {                                                         %         %
        gs' ?x 1 add 0 tr' [.3]CPdash 0 slc'                            %         %
        9.45 13.44 mt' 0 1.4 rlt' 12.4 16.54 lnt' 0 1 rlts              %    K    %
        gr'                                                             %         %
        }def                                                          % %         %
                                                                                  %
                                                                                  %
/H {0 HLpath}def                                                      % %         %
/L {1 HLpath}def                                                        %         %
                                                                        %    H    %
      /HLpath {                                                         %         %
        gs' ?x 0 tr' [.3]CPdash 8.95 13.34 mt'                          %         %
        0 1.5 rlt' 6 16.54 lnt' 0 .8 rlts                               %    L    %
        gr' stnd                                                        %         %
        }def                                                          % %         %
                                                                                  %
                                                                                  %
/I {0 IMpath}def                                                      % %         %
/M {1 IMpath}def                                                        %         %
                                                                        %    I    %
      /IMpath {                                                         %         %
        gs' ?x 9.7 add 10 tr' [.27]CPdash 0 .4 mt'                      %         %
        0 -2.8 rlt' stg# 0?{#j}{#n}ifs' 180 Delay ITog                  %    M    %
        gr'                                                             %         %
        }def                                                          % %         %
                                                                                  %
                                                                                  %
/J {0 JNpath}def                                                      % %         %
/N {1 JNpath}def                                                        %         %
                                                                        %    J    %
      /JNpath{                                                          %         %
        gs' ?x 9.7 add 0 tr' /zl cdef                                   %         %
        [.32]CPdash 0 slc' 0 0 mt' 0 3.5 rlts Latch                     %    N    %
        gr'                                                             %         %
        }def                                                          % %         %
                                                                                  %
                                                                                  %
%+   up] Ahd! + Dot! + Box! + cRelate [up                                         %
%+   up] (CIRCUIT *COMPONENTS*)                    *controlpath* procs            %
%+   dn] CPdash                                % % % % % % % % % % % % % % % % % %
                                                                        %         %
                                                                        %         %
                                                                        %         %
/CPdash {  % <== (*CONTROLPATH* PROCS) [up      % % % % % % % % % % % % % % % % % %
/dash edef                                      % II                              %
/zz cdef                                        %             CPDASH              %
[{black 2 [] .07}                               %                                 %
 {rising?{red}{black}ifs' 0 dash .28}           %     shows *wire nets* in        %
 {red 2 [] .28}                                 %     the TRANSITION STATE        %
] eget exec slw' 0 sd' slc'                     %                                 %
}def                                            % % % % % % % % % % % % % % % % % %



                % % % % % % % % % % % % % % % % % % % % % % % % % % % %
        %       %                                                     %       %
      % % %     %       Ann Coulthard smooths all paths!              %     % % %
        %       %                                                     %       %
                % % % % % % % % % % % % % % % % % % % % % % % % % % % %
```

DATAPATH PROCS P Q R S PIPE PROCS ... PIPE PROCS SNAPIT!

```
%  The *DataPath* procedures generate
%  identical twins, together creating
%  a thick pipe-like path; all invoke
                                    %
/twin {0 -.6 tr'}def            % <---


     %% (TYPICAL)--->   /P {  % <== Snapit!\DrawFIFO! [dn


/P {                                                  % %
/dpath? tdef /zP cdef                                   %
gs' Pipe -2.2 #p 11 #j Pipe.a                           %     P
2{0 180 -1.8 -1 Ahd! 8.7 #j Pipe.b twin}rpt'             %
gr'                                                     %
}def                                                  % %


/Q {                                                  % %
dup 1? zP 0? and{1}{0}ifs' /qsx edef                    %
Pipe #j zNow 2?{#p}{#q}ifs' 10.9 8.7 -1 Pipe.c          %     Q
#j zNow 0?{/# #j def J}if                               %
}def                                                  % %


/R {                                                  % %
gs' /zR cdef Pipe                                       %
20 #r 22.1 #n Pipe.a                                    %     R
2{42.2 #n Pipe.b false InvAmp twin}rpt'                 %
gr'                                                     %
}def                                                  % %


/S {                                                  % %
dup 1? zR 0? and{1}{0}ifs' /qsx edef                    %
Pipe #n zNow 2?{#r}{#s}ifs' 11.1 42.1 -1 Pipe.c         %
gs' 2{0 180 53.9 -1 Ahd! twin}rpt' gr'                  %     S
#n zNow 0?{/# #n def N}if                               %
/dpath? fdef                                            %
}def                                                  % %


%+    up] Ahd! + InvAmp [up                                *datapath* procs
%+    PIPE PROCS [dn                          % % % % % % % % % % % % % %
                                                                      %
                                                                      %
                                                                      %
%  Crafty animation of the *DataPath*
%  needs the following sub-procedures
%  plus STATE-dependent labeling [see
%  LabelSIGS!\DPLAB's subprocedures];
%  each of the PATHS P, Q, R & S uses
                                    %
/Pipe {                         % <---
[{1 sg' .07 slw'}               % 0
 {rising?                       % 1
   {qsx 0?
      {.6 sblue}{.85 sg'}ifs'
   }{.85 sg'}ifs' .1 slw'
 }
 {.7 sblue .18 slw'}            % 2
] eget exec
}def
```

```
                                                                        %
/PipeCut? {dup zNow 0? exch tTime -1 ne and Time 0 ge and}def           %
                                                                        %
/Pipe.a {PipeCut?{1.3 sub}if b3r -1 Pipe.c}def                          %
                                                                        %
/Pipe.b {                                                               %
dup zNow 2? not                                                         %
 {0 180 4 2 roll PipeCut?{.8 sub}if -1 Ahd!}                            %
 {pop pop}ifs'                                                          %
}def            %+  Ahd [up                                             %
                                                                        %
/Pipe.c {                                                               %
gs' tr' /pln edef Xt?{[.305]0 sd'}if 0 slc'                             %
gs' .53 slw' 0 -.3 mt' pln 0 rlts gr'                                   %
0 sg' .07 slw' 2{0 0 mt' pln 0 rlts twin}rpt'                           %
gr'             %                                          pipe procs   %
}def                                                 % % % % % % % % % % % % %



%  HERE'S THE PAYLOAD:  Snapit! calls          % % % % % % % % % % % % % % %
%  on the simulation (PHASE I) procs,          % I II                     %
%  & then *FIFO* depiction (Phase II)          %           SNAPIT!       %
%  procs that, together with whatever          %                         %
%  optional procs have been specified          %   Compose & Display Image %
%  by the user, compose the image for          %                         %
%  this *time*-step and this snapshot          % % % % % % % % % % % % % % %
                                                                        %
                                                                        %
/Snapit! {      % <== up] CODABO! & CODABO!\ShowCast! [up               %
gs'                                                                     %
NameFig!            % <--- Name it!                                     %
Step!               % <--- Launch PHASE I!                              %
DrawFIFO!           % <--- Launch PHASE II!                             %
LabelSIGS!          % <--- Label the signals!                           %
cast? not                       %                                       %
 {?Options          % <--- Add optional features if any!                %
  Tick!             % <--- Advance *time* by one *time*-step!           %
 }if                            %                                       %
gr' save showpage restore       % <--- Print it!                         %
}def                            %                                       %
      /?Options {               %            OPTIONS:                   %
        ZaP?{ZAP!}if            % <--- Z-Arrays & PATH tags              %
        Ivan?{1 Sutherland!}if  % <--- Wheels of Reinvocation            %
        Jo?{Ebergen!}if         % <--- Ebergen analysis delays          %
        scope?{1 Scope!}if      % <--- *Scope-traces*                    %
        }def                    %   (see CODABO! for cast?-Option)      %
                                                                        %
      /NameFig! {                                                       %
        gs' 27 10.8 tr' 1.7 fHB black 0 0 mt'                           %
        cast?{{0 0 mt' cshow 0 -2 tr'}forall}{tStamp}ifs'               %
        gr'                                                             %
        }def                                                            %
              /tStamp {scope? not{Time 1 sub ncshow}if}def              %
                                                                        %
      /DrawFIFO! {   % (Deceptively simple, this PATH MASTER!)          %
        0 1 18                                                          %
          {/# cdef dup ZNOW eget exch PATHS eget cvx exec}for           %
        }def                                                            %
                                                                        %
  %+  up] Step! + LabelSIGS! + Ebergen! [up                    snapit!  %
  %+  up] ZAP! + Scope! + Sutherland! + Tick! [up   % % % % % % % % % % % % %

  %%                                                                %%
```



	A	B	D	E	C	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
ZREF [	2	2	0	2	2	2	2	2	0	0	2	0	2	2	2	0	0	0	0]
ZWAS [	0	0	2	0	2	0	0	0	2	2	2	2	0	0	2	2	2	2	0]
ZNOW [	0	0	2	0	1	0	0	0	2	2	2	2	0	0	1	2	2	2	0]
TIMEX [	13	~	8	~	0	~	~	~	2	~	2	~	~	~	0	6	~	~	~]

SCOPEUS MAXIMUS

```
%  SO YOU ALMOST HAVE IT!  You got your TIMEX gadget, you got your TRANSITIONS,
%  you got your STATES, and there's this CODABO thing and it starts CASTING and
%  PROBING and ZAPPING the CHARACTERS and then the FIFO gadget is STIMULATED to
%  BOX the OPTION, and EBERGEN's always being DELAYED on some PATH or other and
%  SUTHERLAND's a WHEEL and the DATA PIPES while the COMPONENTS are BURNING and
%  if that MULLERC thing's on FIRE you have to PROC your STEP, etc. etc. ...
```

WRAPPING IT ALL UP and ...

```
      %>>>>        NOW, PAY ATTENTION!  THE NUB OF THIS WHOLE        <<<<%
      %>>>>           THING IS COMING ALONG AT ANY MOMENT!           <<<<%


% ==>>  FIRST, YOU SPECIFY THESE VALUES (Non-negative Integers, Please)  <<==
                                        %
/SrcL 0 def     %  # of *time*-steps by which the *Source* limits *FIFO input*
/SnkL 0 def     %  # of *time*-steps by which the *Sink* limits *FIFO output*
/Lag  0 def     %  # of *time*-steps by which *input DATA* lags *SourceReq*
                                        %
/DELAYS [                                                  % % %
~       % A  SourceReq (calc'd by PrepIO!)                   %
8       % b  Dff: MullerC "ff"-delay (for both stages)       %
8       % d  Drr: MullerC "rr"-delay (for both stages)       %          NOTES:
3       % E  TsubI -- *Inverter*                             %
7       % C  TsubR                                           %     1. *Rise-time*
2       % F  TsubA                                           %        = *Fall-time*
2       % G  ITOG.0 (transparency input)                     %        everywhere
5       % H  ITOG.0 (opacity input)                          %
4       % I  ITOG.0                                         %     2. The transition
2       % J  ITOG.0-output                                   %        *time* of a
2       % K  ITOG.1 (transparency input)                     %        *wire net* = 1
5       % L  ITOG.1 (opacity input)                          %
4       % M  ITOG.1                                         %     3. *MullerC*
2       % N  ITOG.1-output                                   %        delays are not
~       % O  SinkAck (calc'd by PrepIO!)                     %        CharliePlot-
~       % P  DsubIn  (calc'd by PrepIO!)                     %        corrected
~       % Q  LATCH.0 (unused for now)                        %
7       % R  DATAPATH *Amplifier*                           %
~       % S  LATCH.1 (unused for now)                        %
]def                                                   % % %


%  ===>>  THEN YOU ACTIVATE THESE OPTIONAL MODE-ASSERTIONS AS DESIRED  <<===
                                        %
%/ZaP? tdef     % Use to expose the Z(STATE)-Arrays and identify PATHS
                % [Primarily a debugging tool, but interesting anyway]
                                        %
%/cast? tdef    % Use to start with an expository pre-sequence of 3 pictures:
                %       1) The 2-stage FIFO Circuit Diagram;
                %       2) The same diagram with identification [see CAR] of
                %             the PATH-delays tracked in the Ebergen analysis;
                %       3) The *FIFO* STATE produced by the PRESET event.
                                        %
                                        %
%/Jo? tdef      % Use to display specified Ebergen PATH-delay labels [see EAR]
                                        %
%/Ivan? tdef    % Use to display the Sutherland WHEELS OF REINVOCATION
                                        %
%/scope? tdef   % Use to display *Scope traces* from *Latch & Datapath probes*


%%%  FINALLY,   % <<===    (and the final two parameters are yours too),
%%%    YOU
%%%  EITHER: advance
       %       to
37     % <--- this  *time* and Snapshot & Display the *FIFO* STATE plus
123    % <--- this   many more *FIFO* STATES in succession [< ~200 snaps, say]
       %
%%%    OR: ==>  %   You ignore all your settings and use the   %  For producing
                %     Demo Set instead -- by activating        %  variant snaps
%/demo? tdef    % <--- this  DEMO-mode assertion               %  see CASTIMAGE
                                                                        %CastIt!
                                                                    CODABO! end
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  SO GO ALREADY!  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

HOW YOU RUN THE DARN THING!

AFTERWORD

I realize that a program listing isn't the proper place for acknowledgements and general comments, but a few words are called for nevertheless and here they are. Let me point out first that the FIFO circuit simulated and described hereinabove is the subject of a pending Sun Microsystems Laboratories patent and represents the work of Ivan E. Sutherland, William S. Coates, the late Charles E. Molnar, and Robert F. Sproull. This circuit is one of many rich contributions to computerology being made by SML's "Async" group under Ivan's energetic leadership and with the occasional and very productive participation of Bob Sproull from SunLabs East.

It is quite gratifying to me that my playful foray into the simulation of this very intriguing FIFO device has indeed turned out to be useful in demonstrating some of the gadget's behavioral subtleties. Over the first few months of my assignment as an SML consultant (during which period I also began to put FifoSim.ps together, but on home turf), I found myself composing bits of doggerel around the names of various members of the dedicated Async group. Later, when I tried to print FifoSim's PostScript procedures in tidy sections that mapped cleanly onto the pages of a program listing in logical order, I realized that these diversionary snippets would serve perfectly as adjustive space fillers here and there, so I installed them in the simulator as further "comments." Surely, Jo Ebergen, Bill Coates, Ian Jones, Jon Lexau, Scott Fairbanks and, of course, Ivan — all of whom have been enthusiastic FifoSim demonstratees and demonstrators if not yet intrepid readers of PostScript code — will take my commemorative inclusions in the intended spirit of friendly good humor, as will Ann Coulthard, who continues to keep us all in good spirits administratively.

But why does one read a program listing anyway? To modify, to extend, perhaps even to learn — these are the usual suspects. Certainly I've had my share of each of these motivations while developing the program, in the course of which my Gray Belt in PostScript Programming has indeed become considerably darker, though it is still far from black. Driven onward (much too pridefully, I regret to confess) by an energizing comment of Lawrence Butcher's that he'd never seen such readable PostScript code, I proceeded to elaborate my little listing program to an excessive degree not at all warranted by the FIFO-simulation program it documents. Call it PostScriptian bravado, or more charitably, an adventure in creating a document that one might actually enjoy scanning through. It has been strongly influenced by Marc Donner's proposition that if reading program listings were more fun then maybe more programmers would do it...and it does seem that even in serious computerology the perusal of programs has turned into an increasingly rare exercise, the norm becoming instead one of docile dependence on "applications" [ugh] that manage to do only more or less what we all ought to be demanding of them. And wherever feasible, I have also tried to follow Marc's excellent practice of identifying in place each program procedure's invoking sources; and in these and further aids to navigation, the appendages 'up]' or '[up' and 'dn]' or '[dn' are scrolling directions for the reader who is content with a previewer's "linear" presentation.

Over the years, we all cheerfully build into our modes of expression many words and phrases that just happen to tickle our funny bones [for example]. Here are a few I seem to have appropriated, together with what I believe to be the correct attributions: The delightful turn-of-the-(last)-century archaism 'and nicely too' [on page 1] was apparently concocted by one J. W. Held Jr.; I modestly attach this to whatever I can in the way of personal achievement. 'For Perusal by the Brave' [page 2 – (note: lowest-level paginating from here on)] is how Robert R. Everett introduced the old Whirlwind computer's block diagrams, at that time thought to be unusually complex. We all know about "selling the Brooklyn Bridge" to the naive [page 4], a notion that first surfaced during the bridge's opening ceremonies in the late 1800's (no, I wasn't there). 'The Wheels of Reinvocation' [page 6]: a play on the *Wheel of Reincarnation* (found in an amusing 1968 paper by T. H. Myer and Ivan the Sutherland, "On the Design of Display Processors"). To 'feel humble and proud' [page 6] is one of the many thought-disrupters given voice by H. Phillip Peterson at Lincoln Laboratory. 'A job-and-a-half' [page 7] is a phrase I was taught by a cabinetmaker named, as I recall, Olafsson (though he pronounced it "a yob-and-a-half"). *To err is human* ...' [page 7] is the familiar old saw that has so often been corrupted; here I mistreat not only the words of Pope, who first wrote them, but also those of Cervantes *(Who errs and mends, to God himself commends,* as translated in my copy of Roget's Thesaurus). And *Now, Pay Attention!* ...' [page 14] was intoned in mock seriousness by Robert Benchley in a short 1944 training film on hygiene produced for the U.S. Navy (I was there).

Finally, a bit of traditional colophony: Apart from this page, only Courier and Helvetica fonts in various forms and sizes have been used. Wow. Only somewhat more fascinating, I suppose, is that the accents and emphases appearing throughout were all introduced manually by modifying font selection calls in the listing program's PostScript file. What is certainly notable, though, is the manner in which the several figures illustrating FifoSim's output displays were produced: to make them I simply embedded a slightly reduced version of FifoSim.ps itself in my listing program and then called on it to generate each required image when given its "snapshot" parameters. Dynamic listing! Totally super! Of course, except for the figures and various accents, the listing printed above accurately documents FifoSim.ps's contents. Pay no attention to my "natural" justification of fixed-pitch, Courier-font paragraphs, which is very little more than a harmless parlor trick made feasible by the inherent and quite marvelous flexibility of the English language, with its permissible abbreviations and punctuations plus a few excusable liberties...such as the dropping of any and all final 'periods' that disturbed my quiet enjoyment of rectangularity.

Wes Clark
November, 1997
Brooklyn, New York

Post Script: Ivan, I am happy to report, has forgiven my insubordination.