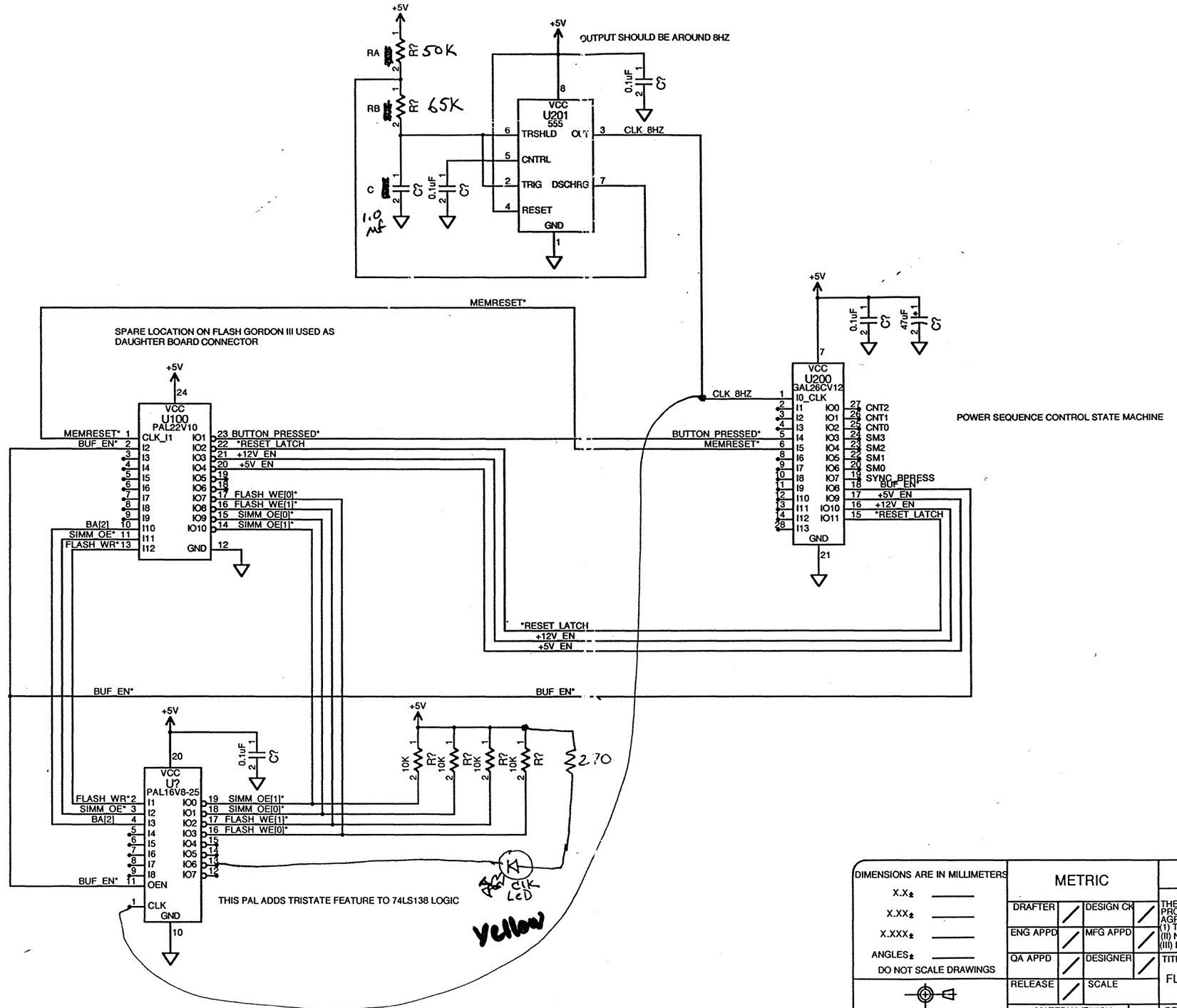


DIMENSIONS ARE IN MILLIMETERS
X.X±
X.XX±
X.XXX±
ANGLES±
DO NOT SCALE DRAWINGS
THIRD ANGLE PROJECTION

METRIC

DRAFTER		DESIGN CK	
ENG APPD		MFG APPD	
QA APPD		DESIGNER	
RELEASE		SCALE	

MATERIAL/FINISH NOTED AS APPLICABLE | SIZE D

Apple Computer, Inc

TITLE
FLASH_III+JW

DRAWING NUMBER | REV.

SHT 6 OF 6

≈ 40 µS for +12V switched to go to 0V

Remove switch use locations for power transistors

SIMM_+12V {4}
SIMM_+5V {4}
SIMM +12V
SIMM +5V

C2 0.01UF
C21 0.01UF
C20 0.01UF

S1
+5VEn
+12VEn
SIMM_+12V {4}
SIMM_+5V {4}

R5 220
GREEN DS1

+12V
IN4933 or 5817
2200 µf
+5V
2200 µf

Input to pal

Slow Turn on ckt for 28020 Vpp Rise min spec ≈ 24µS min

7406
2.2K
1K
47Ω
0.1µf used 2
1/2W 75
1/2W 75
620Ω
Red/
1K
7406
1K
75
270
1watt 2
Green

R2 4.7K
R3 4.7K
C11 0.01UF

*Pushed

U11 74LS132 (1/4)
*Button Pushed
BUF_EN* {1,2}

of Top BD

AUX_CPUCLK {1}
BA[31] {2}
BA[30] {2}
BA[29] {2}
BA[28] {2}
BA[27] {2}
BA[26] {2}
BA[25] {2}
BA[24] {2}
RW {1}
MEMRESET* {1}

U2 PAL22V10
CLK_I1
I2 I3 I4 I5 I6 I7 I8 I9 I10 I11 I12
IO1 IO2 IO3 IO4 IO5 IO6 IO7 IO8 IO9 IO10
GND

TA* {1}
TBI* {1}
STATE[0]
STATE[1]
STATE[2]
STATE[3]
BUF_EN*
BUF_EN* {1,2}
BR/W {2}
FLASH_WE*
SIMM_OE*

U1 74F74
TS* {1}
POST_TS*
C1 0.01UF

BUF_EN*

Buf_En
memReset*
*Reset_Latch

SPARES
U11 74LS132

SPARE
U100 PAL22V10
*Button pushed
*Reset_Latch
+12-En
+5V-En
Flash_WE[0]*
Flash_WE[1]*
Simm-OE[0]*
Simm-OE[1]*
C100 0.01UF

1K

No Tristate!

BA[2]
Simm-OE*
Flash-WE*

74F138
BA[2] {2}
SIMM_OE*
FLASH_WE*
FLASH_WE[0]* {4}
FLASH_WE[1]* {4}
SIMM_OE[0]* {4}
SIMM_OE[1]* {4}
C30 0.01UF

pullups

Remove U42

Flash-WE*
Simm-OE*
BA2
Flash WE Ø
Simm OE Ø
This Decoder Added for Flash Gordon III

+5V-En
+12Vout
+5VIn
+5Vout
Gnd
+12V-En
+12VIn

12P08 moto p-chan
G D S

(ROM SIMM Connectors - next page)
SIMM Isolation Switch Circuitry
Control PAL

Item 5538976 21-Oct-94 17:25PDT

Sub: Flash SIM Fallout

Hi Everyone,

This link is a follow up to the link I sent a while back in regards to the AMD Flash SIM fallout.

AMD has done an analysis on 16 devices from 1 bad sim and found one failing device. The device exhibited severe leakage current on the Vpp pin (Ipp was approx. 1.3ma vs spec limit of <1ua). This failure mode is indicative of exposure to electrical over-stress (EOS) on the Vpp pin, which usually implies an excessive voltage during programming.

We would like to meet with the people or person that does the programming and measure the voltage during programming to determine if possibly a voltage spike is "blowing up" these devices. Would you let me know who we should talk with.

RESULTS OF APPLE FLASH SIMM EVALUATION

Problem: Apple computer reports 30 to 40 % failure rate of flash simms.

Device: Device consists of 16 pcs. AM28F020-95JC, 1 pc. SN74F86D, 16 pcs. VJ0805U103MXAMT, and 1 pc. AP2216-02 (PCB).

Testing performed:

- During our visit to Apple on 5/2/95, we attempted to flash 10 pcs. of the simms with Apple's custom programmer. The programming was done by Sy Van Hoang.

- 10 pcs. of the simms were disassembled for further analysis by Wyle. (8 pcs. "bad", 1 pc. "good" , 1 pc. "unknown")*

- AM28F020-95JC chips were tested by attempting to program them individually. (10 from "bad", 16 from "good", 16 from "unknown" and 10 from Wyle stock)**

- "Stripped" boards were sent back to Winonics for testing.

- Original test data was obtained from Winonics for 400 pc. build (Catalyst # EM45674)

Test results: ***

- Of the 10 simms we attempted to flash at Apple: 4 were unrecognized by the programmer, 1 reported 2 bad chips, and 5 were programmed successfully. (50% fallout).

- Of the Individual chips we attempted to program:

	"Bad"	"Good"	"Unk."	Wyle
Passed:	1	14	12	9
Failed:	9	2	4	1
Total:	10	16	16	10
Failure %:	90	12.5	25	10

- Analysis of stripped boards (PCB) not available at this time.

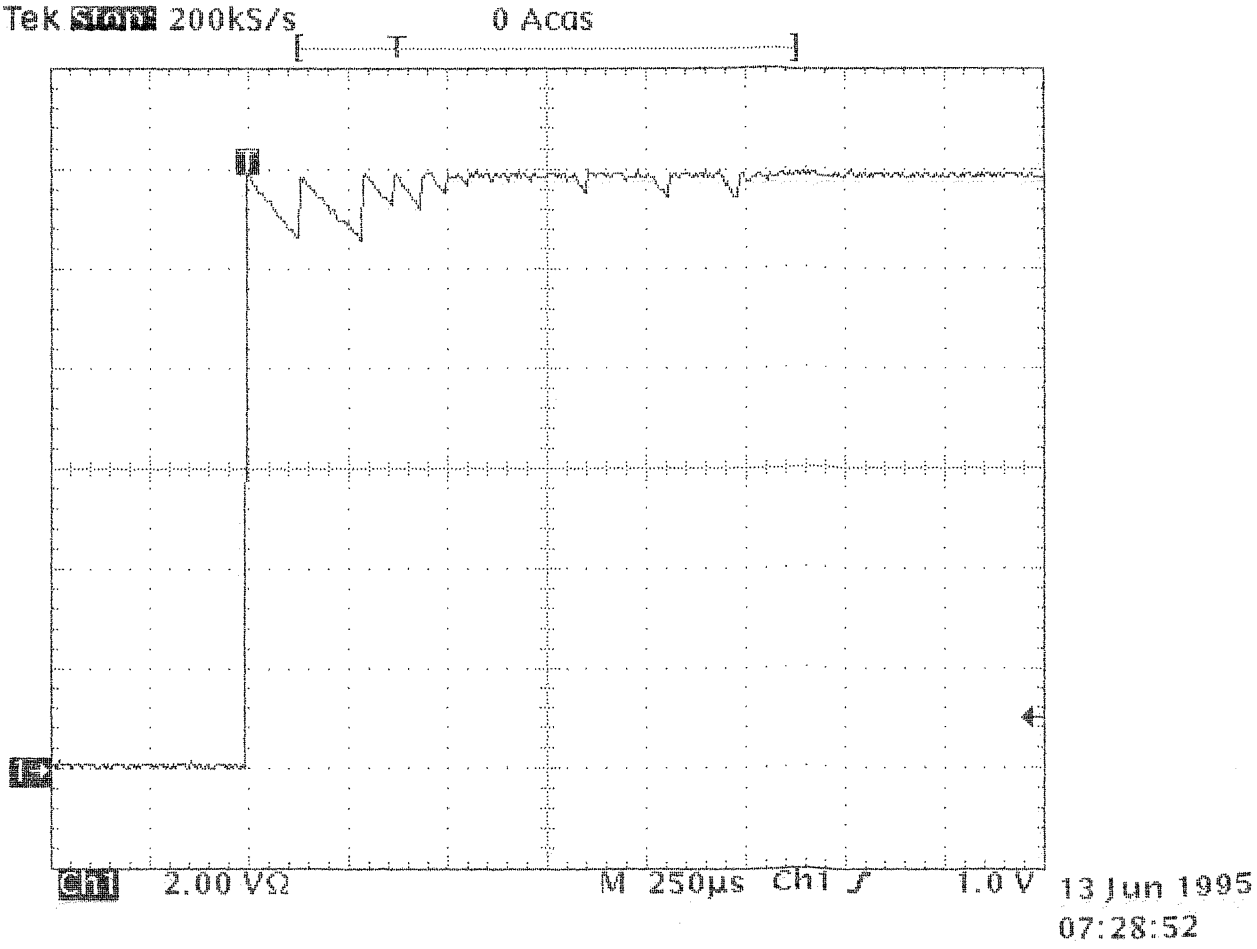
- 100% testing was performed on boards prior to assembly: (see attached document for 400 pc. Catalyst build.)

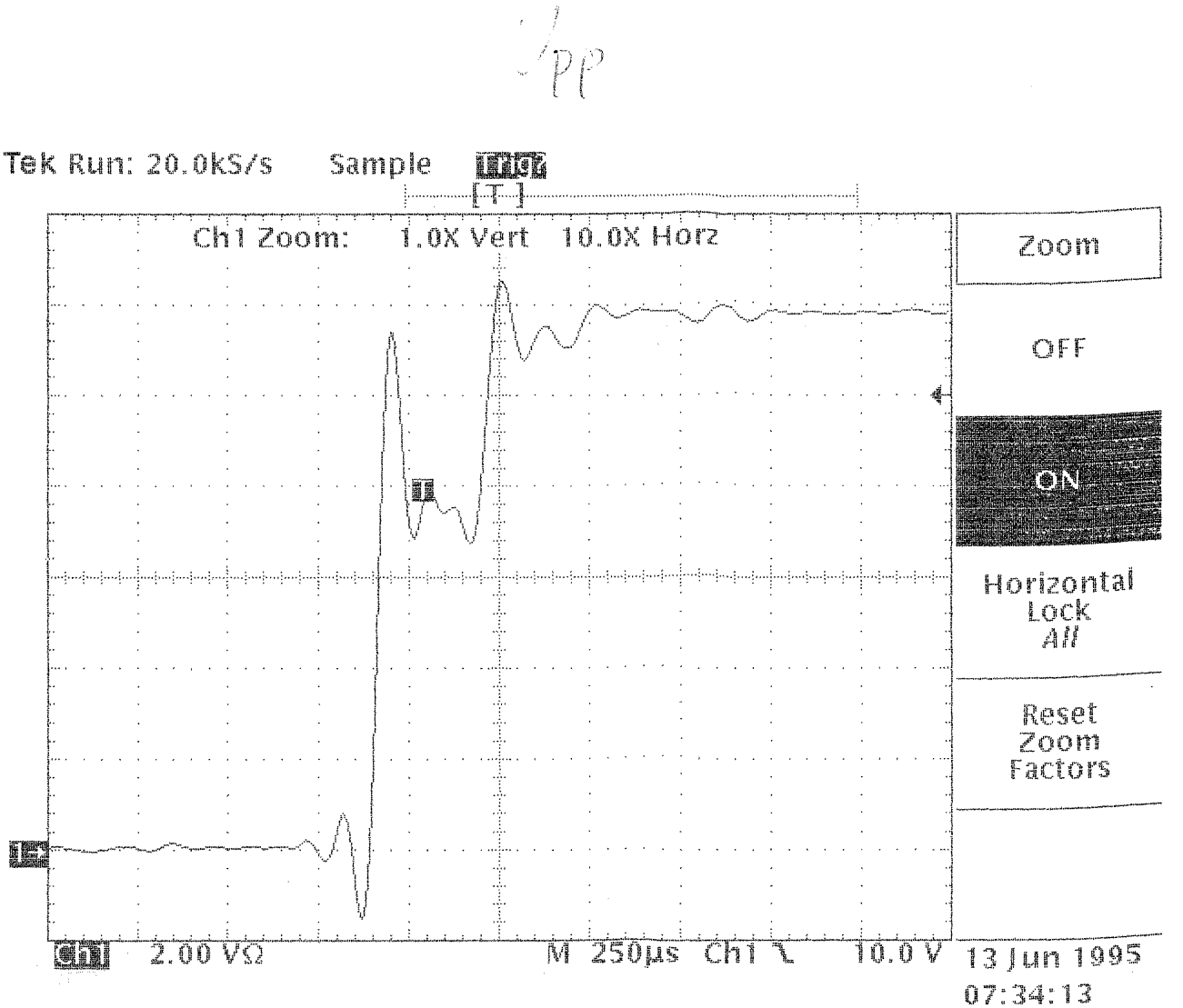
* Disassembled simms were from known good & bad modules tested at Apple on 5/2/95. Unknown was from untested batch.

** Good, bad and Unk. chips from the disassembled modules. Wyle chips were new chips from stock.

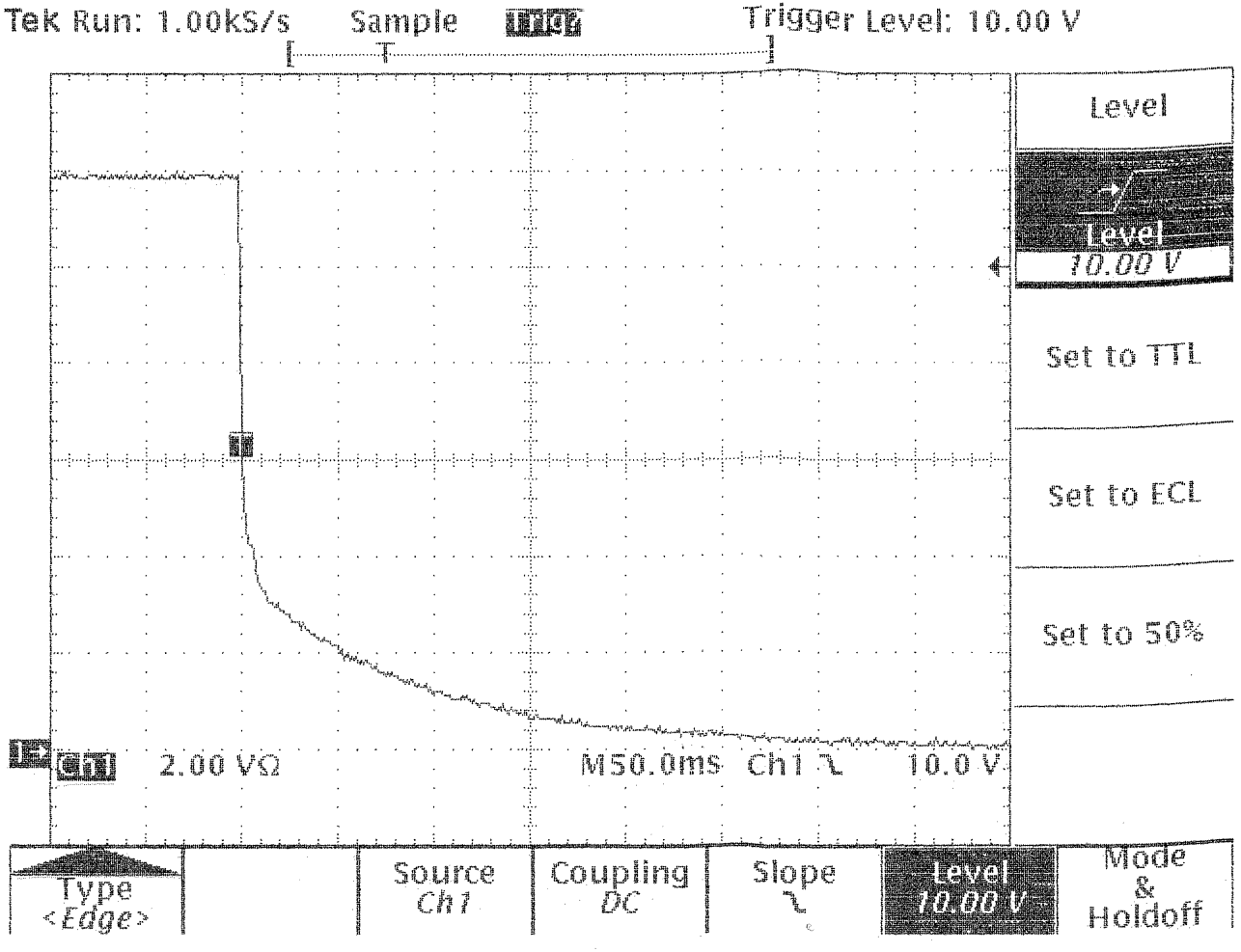
*** Results from testing may or may not indicate device failure since comprehensive analysis tools were not available.

1pp

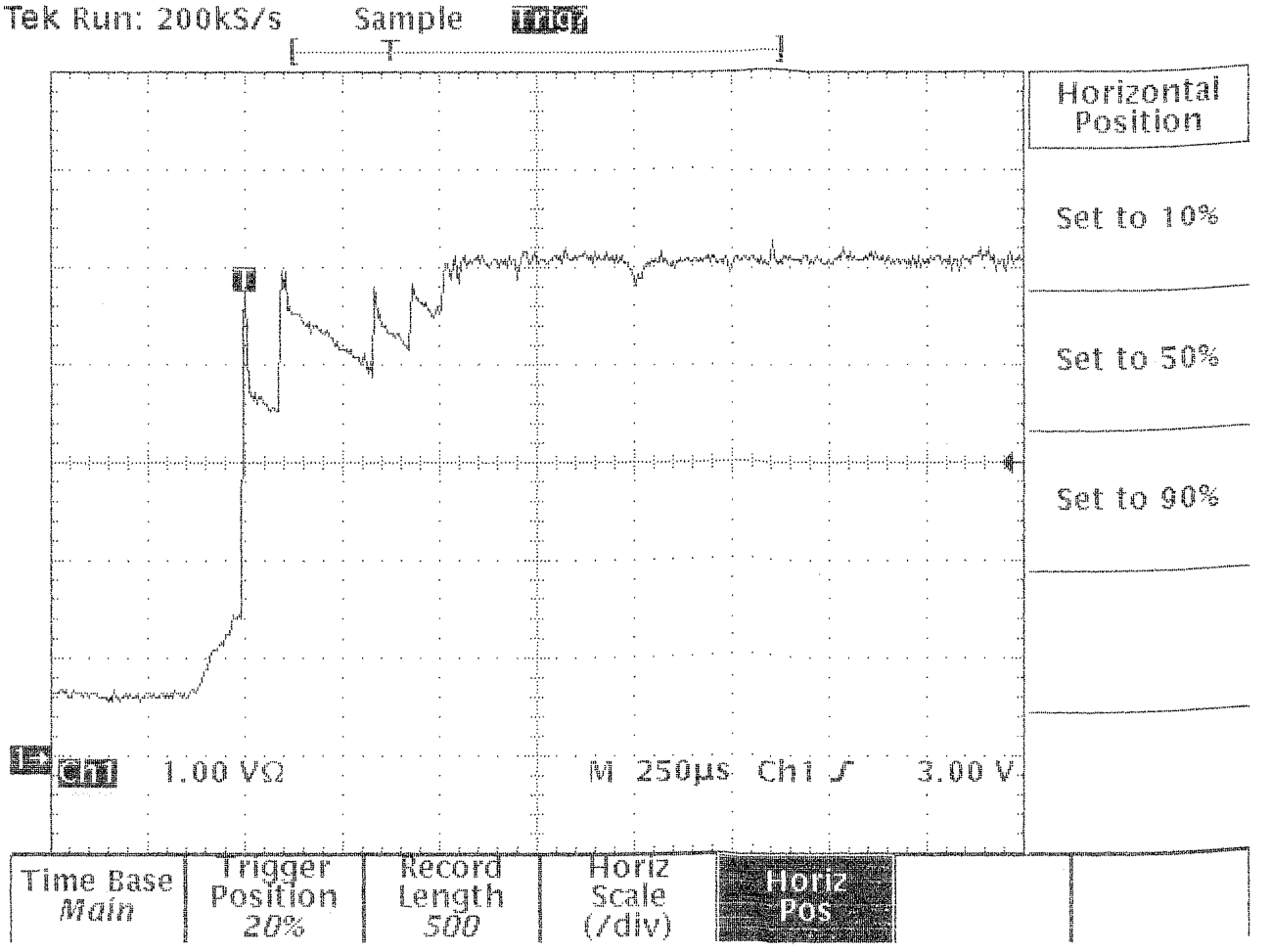

Tek Stop 200kS/s
0 Acqs
Ch1 2.00 VΩ
M 250µs Ch1 ƒ 1.0 V
13 Jun 1995
07:28:52

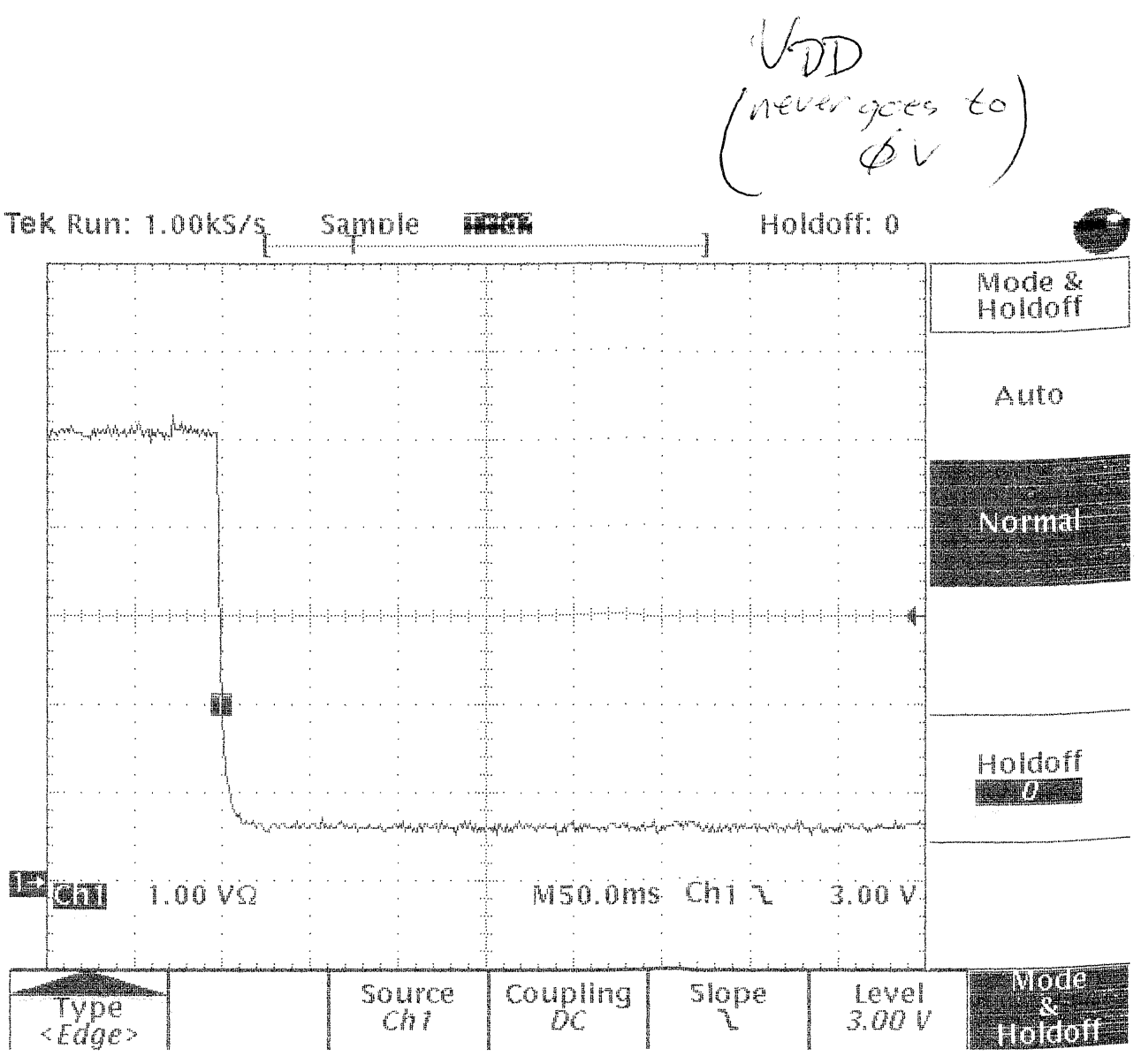

Tek Run: 20.0kS/s Sample
Ch1 Zoom: 1.0X Vert 10.0X Horz
Zoom
OFF
ON
Horizontal Lock All
Reset Zoom Factors
Ch1 2.00 VΩ
M 250µs Ch1 ∖ 10.0 V
13 Jun 1995
07:34:13

1/pp


Tek Run: 1.00kS/s Sample
Trigger Level: 10.00 V
Level
Level
10.00 V
Set to TTL
Set to ECL
Set to 50%
2.00 V Ω
M50.0ms Ch1 ∖ 10.0 V
Type
<Edge>
Source
Ch1
Coupling
DC
Slope
Level
10.00 V
Mode
&
Holdoff

V_{DD}


Tek Run: 200kS/s Sample
Horizontal Position
Set to 10%
Set to 50%
Set to 90%
1.00 VΩ
M 250µs Ch1 ƒ 3.00 V
Time Base Main
Trigger Position 20%
Record Length 500
Horiz Scale (/div)
Horiz POS


VDD
(never goes to
0 V
Tek Run: 1.00kS/s Sample
Holdoff: 0
Mode & Holdoff
Auto
Normal
Holdoff
0
Ch1 1.00 V Ω
M50.0ms Ch1 ↘ 3.00 V
Type
<Edge>
Source
Ch1
Coupling
DC
Slope
↘
Level
3.00 V
Mode & Holdoff

```
module FLASH_KEL_PAL
title 'Eclipse/Spike FLASH ROM PAL for PDS
Jason Stewart/Jano Banks
Apple Computer
21 March 1991'


     flash_kel device 'P22V10';

"--------------------------------
"INPUTS     (12)
"--------------------------------
    Clk          pin     1;
    A31          pin     2;
    A30          pin     3;
    A29          pin     4;
    A28          pin     5;
    A27          pin     6;
    A26          pin     7;
    A25          pin     8;
    A24          pin     9;
    RW           pin     10;
    !Reset       pin     11;
    !TS          pin     13;
    !Buf_En      pin     17;


"--------------------------------
"OUTPUTS    (6)
"--------------------------------
    !TA          pin     23;
    !TBI         pin     22;
    !WE_St0      pin     21;
    !OE_St1      pin     20;
    !Cnt0_St2    pin     19;
    !Cnt1_St3    pin     18;
    BRW          pin     16;
    !FlashWE     pin     15;
    !FlashOE     pin     14;

"--------------------------------
"ALIASES"
"--------------------------------
    C,H,L,X,Z    =    .C.,1,0,.X.,.Z.;
    X4           =    [X,X,X,X];
    X6           =    [X,X,X,X,X,X];
    X8           =    [X,X,X,X,X,X,X,X];

"--------------------------------
"Set assignments
"--------------------------------
    Addr         =    [A31,A30,A29,A28,A27,A26,A25,A24];
    DataSpace    =    ^hE0;
    CmdSpace     =    ^hEC;

    HiAddr       =    [A31,A30,A29,A28];
    LoAddr       =    [A27,A26,A25,A24];
    PDSSpace32   =    ^hE;
    PDSSpace24   =    ^hFE;

"--------------------------------
"STATES"
```

```
"--------------------------------
    CState  =   [Cnt1_St3,Cnt0_St2,OE_St1,WE_St0,TBI,TA];
    IDLE    =   ^b000000;
    S0      =   ^b001000;
    S1      =   ^b001100;
    S2      =   ^b011100;
    S3      =   ^b101100;
    S4      =   ^b011000;
    S5      =   ^b101000;
    S6      =   ^b001011;

    CStateX =   [Cnt1_St3,Cnt0_St2,OE_St1,WE_St0];
    IDLEX   =   ^b0000;
    S0X     =   ^b0010;
    S1X     =   ^b0011;
    S2X     =   ^b0111;
    S3X     =   ^b1011;
    S4X     =   ^b0110;
    S5X     =   ^b1010;
    S6X     =   ^b0010;


"--------------------------------
equations
"--------------------------------
"The state machine will start with a TS and HiAddr = $E"
"It will be for 7 clocks and TA and TBI will be generated"
"A FlashWE will be generated for a write to CmdSpace"
"A FlashOE will be generated for any read"
"NOTE:  the write recovery time must be 6usec, which must be done by software"
"NOTE:  there must be 10usec between the program data and the verify"

"State Machine is IDLE -> S0 -> S1 -> S2 -> S3 -> S4 -> S5 -> S6 -> IDLE
"This allows for a worst case of 33.33MHz 040 clock speed.
"We do take a slight performance hit at 25, but who cares for burning Flash ROMs!

"--------------------------------
"State Machine Eqns
"--------------------------------
Cnt1_St3    :=  (CState == S2) & !Reset
            #   (CState == S4) & !Reset;

Cnt0_St2    :=  (CState == S1) & !Reset
            #   (CState == S3) & !Reset;

OE_St1      :=  (CState == IDLE) & (HiAddr == PDSSpace32) & TS & !Reset
            #   (CState == IDLE) & (Addr == PDSSpace24) & TS & !Reset
            #   (CState == S0) & !Reset
            #   (CState == S1) & !Reset
            #   (CState == S2) & !Reset
            #   (CState == S3) & !Reset
            #   (CState == S4) & !Reset
            #   (CState == S5) & !Reset;

WE_St0      :=  (CState == S0) & !Reset
            #   (CState == S1) & !Reset
            #   (CState == S2) & !Reset;

TBI         :=  (CState == S5) & !Reset;

TA          :=  (CState == S5) & !Reset;
```

```
"--------------------------------
"Output Eqns
"--------------------------------
FlashOE      =   RW & OE_St1 & Buf_En & !Reset & (HiAddr == PDSSpace32);

FlashWE      =   !RW & WE_St0 & (Addr == CmdSpace) & Buf_En & !Reset & (HiAddr == PDSSpace32)

Enable TA    =   OE_St1 & Buf_En & !Reset;

Enable TBI   =   OE_St1 & Buf_En & !Reset;

BRW          =   RW & OE_St1 & Buf_En & !Reset & (HiAddr == PDSSpace32);

"--------------------------------
"   test vectors
"--------------------------------
"******************          enables          ****************
test_vectors 'Tristate Test'
    ([Clk,TS,HiAddr,LoAddr,RW,Buf_En,Reset]
                                  ->  [CStateX,TA,TBI,FlashOE,FlashWE,BRW])
    [X, X, X4, X4, X, 0, 0]       ->  [X4,Z,Z,0,0,0];
    [X, X, X4, X4, X, 1, 1]       ->  [X4,Z,Z,0,0,0];
    [C, 0, X4, X4, X, 1, 1]       ->  [IDLEX,Z,Z,0,0,0];




test_vectors 'Check Reads'
    ([Clk,TS,HiAddr,LoAddr,RW,Buf_En,Reset]
                                  ->  [CStateX,TA,TBI,FlashOE,FlashWE,BRW])
    [C,0,PDSSpace32,X4,X,1,0]     ->  [IDLEX,Z,Z,0,0,0];
    [C,1,^hF,X4,X,1,0]            ->  [IDLEX,Z,Z,0,0,0];
    [C,1,PDSSpace32,X4,1,1,0]     ->  [S0X,   0,0,1,0,1];
    [C,0,PDSSpace32,X4,1,1,0]     ->  [S1X,   0,0,1,0,1];
    [C,0,PDSSpace32,X4,1,1,0]     ->  [S2X,   0,0,1,0,1];
    [C,0,PDSSpace32,X4,1,1,0]     ->  [S3X,   0,0,1,0,1];
    [C,0,PDSSpace32,X4,1,1,0]     ->  [S4X,   0,0,1,0,1];
    [C,0,PDSSpace32,X4,1,1,0]     ->  [S5X,   0,0,1,0,1];
    [C,0,PDSSpace32,X4,1,1,0]     ->  [S6X,   1,1,1,0,1];
    [C,0,PDSSpace32,X4,X,1,0]     ->  [IDLEX,Z,Z,0,0,0];

test_vectors 'Check Writes'
    ([Clk,TS,Addr,RW,Buf_En,Reset]
                                  ->  [CStateX,TA,TBI,FlashOE,FlashWE,BRW])
    [C,0,DataSpace,X,1,0]     ->  [IDLEX,Z,Z,0,0,0];
    [C,1,DataSpace,0,1,0]     ->  [S0X,   0,0,0,0,0];
    [C,0,DataSpace,0,1,0]     ->  [S1X,   0,0,0,0,0];
    [C,0,DataSpace,0,1,0]     ->  [S2X,   0,0,0,0,0];
    [C,0,DataSpace,0,1,0]     ->  [S3X,   0,0,0,0,0];
    [C,0,DataSpace,0,1,0]     ->  [S4X,   0,0,0,0,0];
    [C,0,DataSpace,0,1,0]     ->  [S5X,   0,0,0,0,0];
    [C,0,DataSpace,0,1,0]     ->  [S6X,   1,1,0,0,0];
    [C,0,DataSpace,X,1,0]     ->  [IDLEX,Z,Z,0,0,0];

    [C,0,CmdSpace,X,1,0]      ->  [IDLEX,Z,Z,0,0,0];
    [C,1,CmdSpace,0,1,0]      ->  [S0X,   0,0,0,0,0];
    [C,0,CmdSpace,0,1,0]      ->  [S1X,   0,0,0,1,0];
    [C,0,CmdSpace,0,1,0]      ->  [S2X,   0,0,0,1,0];
    [C,0,CmdSpace,0,1,0]      ->  [S3X,   0,0,0,1,0];
    [C,0,CmdSpace,0,1,0]      ->  [S4X,   0,0,0,0,0];
    [C,0,CmdSpace,0,1,0]      ->  [S5X,   0,0,0,0,0];
```

```
    [C,0,CmdSpace,0,1,0]    ->  [S6X,  1,1,0,0,0];
    [C,0,CmdSpace,X,1,0]    ->  [IDLEX,Z,Z,0,0,0];

test_vectors 'Check PDS24'
    ([Clk,TS,Addr,RW,Buf_En,Reset]
                               ->  [CStateX,TA,TBI,FlashOE,FlashWE,BRW])
    [C,0,PDSSpace24,X,1,0]  ->  [IDLEX,Z,Z,0,0,0];
    [C,1,PDSSpace24,0,1,0]  ->  [S0X,  0,0,0,0,0];
    [C,0,PDSSpace24,0,1,0]  ->  [S1X,  0,0,0,0,0];
    [C,0,PDSSpace24,0,1,0]  ->  [S2X,  0,0,0,0,0];
    [C,0,PDSSpace24,0,1,0]  ->  [S3X,  0,0,0,0,0];
    [C,0,PDSSpace24,0,1,0]  ->  [S4X,  0,0,0,0,0];
    [C,0,PDSSpace24,0,1,0]  ->  [S5X,  0,0,0,0,0];
    [C,0,PDSSpace24,0,1,0]  ->  [S6X,  1,1,0,0,0];
    [C,0,PDSSpace24,X,1,0]  ->  [IDLEX,Z,Z,0,0,0];


end FLASH_KEL_PAL
```

```
module FLASH_GORDON_III_JW_SM
options '-r3'
title  'FLASH GORDON III JW UPGRADE FOR POWER SEQUENCING'
"*************************************************************************
"*                                                                       *
"* TITLE:       FG3_SMV3                                                 *
"* BY:          JOHN WALKER                                              *
"* FOR:         APPLE COMPUTER INC.                                      *
"* DATE:        11/10/95                                                 *
"* PLD:         LATTICE GAL26CV12, 15nS, 28 PIN DIP                      *
"* ABEL REV:    MAC ABEL 4.2                                             *
"*                                                                       *
"* DESCRIPTION:  THIS PAL CONTROLS THE POWER/TRISTATE SEQUENCING ON THE  *
"*               JW MODIFIED FLASH GORDON III PROGRAMMER BOARD           *
"*                                                                       *
"*************************************************************************

     FG3_SMV3   device   'P26CV12';    "JEDEC FILE NAME IS 'FG3_SMV3.JED'


"**********************  REVISION HISTORY  *******************************
"
"     CREATED:   (V1) ON 11/6/95  - INITIAL ATTEMPT, MAY HAVE PROBLEMS WITH
"                     SINGLE FLOP TO SYNC BUTTON_PRESSED, AND MAY HAVE
"                     PROBLEMS WITH CLOCKED TERMS ON SUCH A SLOW CLOCK (~125MS)
"
"                (V2) ON 11/10/95 - ADDED HOLD ON S4, S9 TO WAIT FOR SYNC
"                     VERSION OF BUTTON_PRESSED TO GO HIGH (WAIT FOR
"                     RESET OF LATCH TO OCCUR, OR USER TO LET GO OF BUTTON)
"
"                (V3) ON 11/10/95 - ADDED TERM TO S0, S5 TO AVOID THE PROBLEM
"                     CAUSED WHEN .AR CAUSED SYNC_BPRESS TO GO LOW DURING RESET,
"                     WHICH CAUSES SM TO ADVANCE AS IF THE BUTTON WAS PRESSED.
"                     SM NOW WATCHES BUTTON_PRESSED AS WELL AS SYNC_BPRESS. MAY
"                     NEED TO REMOVE .AR TERMS, AND USE WHEN STATEMENTS FOR RESET.
"


"*********************  POWER PIN DEFINITIONS  ****************************

    VCC                  pin   7;          "POWER
    GND                  pin   21;         "GROUND


"*********************  INPUT PIN DEFINITIONS  ****************************

    CLK                  pin    1;         "CLOCK INPUT
    IN2                  pin    2;         "NOT USED
    IN3                  pin    3;         "NOT USED
    IN4                  pin    4;         "NOT USED
    BUTTON_PRESSED       pin    5;         "LOW TRUE WHEN BUTTON IS BEING PRESSED
    RESET                pin    6;         "LOW TRUE WHEN IN RESET
    IN7                  pin    8;         "NOT USED
    IN8                  pin    9;         "NOT USED
    IN9                  pin    10;        "NOT USED
    IN10                 pin    11;        "NOT USED
    IN11                 pin    12;        "NOT USED
    IN12                 pin    13;        "NOT USED
    IN13                 pin    14;        "NOT USED
    IN14                 pin    28;        "NOT USED
```

```
"************  IN/OUT PIN DEFINITIONS AND MACROCELL TYPE  ***************

    RESET_LATCH          pin   15  istype 'reg';  "LOW TRUE, RESET SWITCH DEBOUNCE LATCH
    EN_12                pin   16  istype 'reg';  "HIGH TRUE, TURN ON SIMM +12V
    EN_5V                pin   17  istype 'reg';  "HIGH TRUE, TURN ON SIMM +5V
    BUF_EN               pin   18  istype 'reg';  "LOW TRUE, ENABLE BUFFERS
    SYNC_BPRESS          pin   19  istype 'reg';  "SOMEWHAT SYNCRONIZED BUTTON_PRESSED
    SM0                  pin   20  istype 'reg';  "STATE MACHINE BIT 0
    SM1                  pin   22  istype 'reg';  "STATE MACHINE BIT 1
    SM2                  pin   23  istype 'reg';  "STATE MACHINE BIT 2
    SM3                  pin   24  istype 'reg';  "STATE MACHINE BIT 3
    CNT0                 pin   25  istype 'reg';  "COUNTER BIT 0
    CNT1                 pin   26  istype 'reg';  "COUNTER BIT 1
    CNT2                 pin   27  istype 'reg';  "COUNTER BIT 2


"*********************  GROOP/CONSTANT DEFINITIONS  ************************

    XXXXXX       = .X.;
    X            = .X.;
    ON           = 1;
    OFF          = 0;
    HIGH         = 1;
    LOW          = 0;
    TRUE         = 1;
    FALSE        = 0;

    CTRL_SM = [SM3, SM2, SM1, SM0];
    COUNT   = [CNT2, CNT1, CNT0];


"************************  STATE ASSIGNMENTS  *****************************

    S0  = ^h0;       S8  = ^h8;
    S1  = ^h1;       S9  = ^h9;
    S2  = ^h2;       SA  = ^hA;
    S3  = ^h3;       SB  = ^hB;
    S4  = ^h4;       SC  = ^hC;
    S5  = ^h5;       SD  = ^hD;
    S6  = ^h6;       SE  = ^hE;
    S7  = ^h7;       SF  = ^hF;


"*********************  COMBINATORIAL EQUATIONS  **************************
equations

    RESET_LATCH.OE = 1;

    CNT0.AR = (RESET == 0);        "ALL REGISTERS GET RESET WHEN RESET SEEN
    CNT1.AR = (RESET == 0);        "THESE EXTRA EQUATIONS ARE A REMINDER
    CNT2.AR = (RESET == 0);        "SWITCH TO WHEN STATEMENT AND DONT USE .AR

    SM0.AR = (RESET == 0);
    SM1.AR = (RESET == 0);
    SM2.AR = (RESET == 0);
    SM3.AR = (RESET == 0);

    BUF_EN.AR   = (RESET == 0);    "MAY BE A PROBLEM SINCE CLKED ON 10MS, COULD CRASH CPU
    EN_5V.AR    = (RESET == 0);    "PDS PAL WILL PROBABLY FILTER
    EN_12.AR    = (RESET == 0);    "SHOULD PROBABLY INVERT BUF_EN OUTPUT, SO GOES HIGH ON RES

    SYNC_BPRESS.AR = (RESET == 0);
```

```
    RESET_LATCH.AR = (RESET == 0);

    SYNC_BPRESS := BUTTON_PRESSED;  "ATTEMPT TO SYNC BUTTON_PRESSED, NEED ANOTHER FLOP


state_diagram  CTRL_SM;

"************************  STATE S0  **********************************

    STATE S0:
       BUF_EN        := 1;
       EN_5V         := 0;
       EN_12         := 0;
       RESET_LATCH   := 1;
       COUNT         := ^h0;
       IF ((SYNC_BPRESS == 0) & (BUTTON_PRESSED == 0)) THEN S1  "WITHOUT EXTRA PARENS
       ELSE S0;                                                 "ABEL INTERPRETS IT WRONG!!!

"************************  STATE S1  **********************************

    STATE S1:
       BUF_EN        := 1;
       EN_5V         := 1;
       EN_12         := 0;
       RESET_LATCH   := 1;
       COUNT         := (COUNT + 1);
       IF (COUNT >= ^h2) THEN S2
       ELSE S1;

"************************  STATE S2  **********************************

    STATE S2:
       BUF_EN        := 1;
       EN_5V         := 1;
       EN_12         := 1;
       RESET_LATCH   := 1;
       COUNT         := (COUNT + 1);
       IF (COUNT >= ^h4) THEN S3
       ELSE S2;

"************************  STATE S3  **********************************

    STATE S3:
       BUF_EN        := 0;
       EN_5V         := 1;
       EN_12         := 1;
       RESET_LATCH   := 1;
       COUNT         := (COUNT + 1);
       IF (COUNT == ^h7) THEN S4
       ELSE S3;

"************************  STATE S4  **********************************

    STATE S4:
       BUF_EN        := 0;
       EN_5V         := 1;
       EN_12         := 1;
       RESET_LATCH   := 0;
       COUNT         := ^h0;
       IF (SYNC_BPRESS == 1) THEN S5
       ELSE S4;
```

```
"************************  STATE S5  **********************************

    STATE S5:
       BUF_EN        := 0;
       EN_5V         := 1;
       EN_12         := 1;
       RESET_LATCH   := 1;
       COUNT         := ^h0;
       IF ((SYNC_BPRESS == 0) & (BUTTON_PRESSED == 0)) THEN S6
       ELSE S5;

"************************  STATE S6  **********************************

    STATE S6:
       BUF_EN        := 1;
       EN_5V         := 1;
       EN_12         := 1;
       RESET_LATCH   := 1;
       COUNT         := (COUNT + 1);
       IF (COUNT >= ^h2) THEN S7
       ELSE S6;

"************************  STATE S7  **********************************

    STATE S7:
       BUF_EN        := 1;
       EN_5V         := 1;
       EN_12         := 0;
       RESET_LATCH   := 1;
       COUNT         := (COUNT + 1);
       IF (COUNT >= ^h4) THEN S8
       ELSE S7;

"************************  STATE S8  **********************************

    STATE S8:
       BUF_EN        := 1;
       EN_5V         := 0;
       EN_12         := 0;
       RESET_LATCH   := 1;
       COUNT         := (COUNT + 1);
       IF (COUNT == ^h7) THEN S9
       ELSE S8;

"************************  STATE S9  **********************************

    STATE S9:
       BUF_EN        := 1;
       EN_5V         := 0;
       EN_12         := 0;
       RESET_LATCH   := 0;
       COUNT         := ^h0;
       IF (SYNC_BPRESS == 1) THEN S0
       ELSE S9;

"************************  STATE SA  **********************************

    STATE SA:
       GOTO S0;

"************************  STATE SB  **********************************
```

```
    STATE SB:
       GOTO S0;

"************************  STATE SC  **********************************

    STATE SC:
       GOTO S0;

"************************  STATE SD  **********************************

    STATE SD:
       GOTO S0;

"************************  STATE SE  **********************************

    STATE SE:
       GOTO S0;

"************************  STATE SF  **********************************

    STATE SF:
       GOTO S0;

end FLASH_GORDON_III_JW_SM
```

```
module FLASH_GORDON_III_JW_DECODE
options '-r3'
title  'FLASH GORDON III JW UPGRADE, TRISTATE DECODER'
"*************************************************************************
"*                                                                       *
"* TITLE:       FG3_DECV1                                                *
"* BY:          JOHN WALKER                                              *
"* FOR:         APPLE COMPUTER INC.                                      *
"* DATE:        11/9/95                                                  *
"* PLD:         LATTICE GAL16V8, 15nS, 20 PIN DIP                        *
"* ABEL REV:    MAC ABEL 4.2                                             *
"*                                                                       *
"* DESCRIPTION:  THIS PAL REPLACES THE 74138 DECODER. IT DUPLICATES THE  *
"*               138'S FUNCTIONALITY, AND ALLOWS TRISTATE OF ITS OUTPUTS *
"*                                                                       *
"*************************************************************************

     FG3_DECV1  device  'P16V8';   "JEDEC FILE NAME IS 'FG3_DECV1.JED'


"**********************  REVISION HISTORY  *********************************
"
"    CREATED:  (V1) ON 11/9/95
"


"*********************  POWER PIN DEFINITIONS  ****************************

    VCC                  pin   20;           "POWER
    GND                  pin   10;           "GROUND


"*********************  INPUT PIN DEFINITIONS  ****************************

    CLK                  pin    1;          "CLOCK INPUT
    FLASH_WE             pin    2;          "LOW TRUE, ENABLE FLASH WRITES
    SIMM_OE              pin    3;          "LOW TRUE, ENABLE FLASH OUTPUT DATA
    BA2                  pin    4;          "BUFFERED ADDRESS BIT #2
    IN4                  pin    5;          "NOT USED
    IN5                  pin    6;          "NOT USED
    IN6                  pin    7;          "NOT USED
    IN7                  pin    8;          "NOT USED
    IN8                  pin    9;          "NOT USED
    BUF_EN               pin    11;         "LOW TRUE, ENABLE OUTPUTS


"************  IN/OUT PIN DEFINITIONS AND MACROCELL TYPE  ***************

    IO0                  pin   12  istype 'com';  "NOT USED
    LED                  pin   13  istype 'reg';  "LED GOES ON WHEN LOW
    IO2                  pin   14  istype 'com';  "NOT USED
    IO3                  pin   15  istype 'com';  "NOT USED
    FLASH_WE0            pin   16  istype 'com';  "FLASH BANK 0 IS WR ENABLED WHEN LOW
    FLASH_WE1            pin   17  istype 'com';  "FLASH BANK 1 IS WR ENABLED WHEN LOW
    SIMM_OE0             pin   18  istype 'com';  "FLASH BANK 0 IS OUTPUT ENABLED WHEN LOW
    SIMM_OE1             pin   19  istype 'com';  "FLASH BANK 1 IS OUTPUT ENABLED WHEN LOW


"*********************  GROOP/CONSTANT DEFINITIONS  ***********************

    XXXXXX       = .X.;
    X            = .X.;
```

```
    ON              = 1;
    OFF             = 0;
    HIGH            = 1;
    LOW             = 0;
    TRUE            = 1;
    FALSE           = 0;


"********************  COMBINATORIAL EQUATIONS  *************************
equations

    LED.OE          = (BUF_EN == 0);      "JUST IN CASE ABEL TRIES ANOTHER 16V8 MODE
    FLASH_WE0.OE    = (BUF_EN == 0);
    FLASH_WE1.OE    = (BUF_EN == 0);
    SIMM_OE0.OE     = (BUF_EN == 0);
    SIMM_OE1.OE     = (BUF_EN == 0);

    LED := !LED;                          "WHEN BUF_EN, TOGGLE LED ON CLK BOUNDRY

    WHEN ((FLASH_WE == 0) & (SIMM_OE == 1) & (BA2 == 0))
         THEN FLASH_WE0 = 0;
         ELSE FLASH_WE0 = 1;

    WHEN ((FLASH_WE == 0) & (SIMM_OE == 1) & (BA2 == 1))
         THEN FLASH_WE1 = 0;
         ELSE FLASH_WE1 = 1;

    WHEN ((FLASH_WE == 1) & (SIMM_OE == 0) & (BA2 == 0))
         THEN SIMM_OE0 = 0;
         ELSE SIMM_OE0 = 1;

    WHEN ((FLASH_WE == 1) & (SIMM_OE == 0) & (BA2 == 1))
         THEN SIMM_OE1 = 0;
         ELSE SIMM_OE1 = 1;

end FLASH_GORDON_III_JW_DECODE
```