Application Note : Fast A500's

Title:	Application Note : Fast A500's
Reference:	
Issue:	1.4
Replaces:	1.3
Applicability:	
Distribution:	on request
Status:	*draft - not for publication*
Author:	Mike Muller
Date:	12 December 1988

Published by: Acorn Computers Ltd, Fulbourn Road, Cherry Hinton, Cambridge, CB1 4JN, England

This application note describes a two-PAL circuit to be added to an A500 which enables the I0 system to run asynchronously from the main memory system and ARM. This allows the processor memory clock to be sped up giving higher performance while maintaining hardware compatibility for peripherals which can still be run at 8 MHz. As a consequence of the synchronisation overheads, I0 cycles take longer to complete. For IO bound tasks a degradation in system performace may be observed. As the main system clock is increased care must be taken to ensure that all components are still within, specification. The memory system can be run at up to two times the speed of the IO system. The circuit assumes that the VIDC will be run at 24 MHz and uses this clock to produce an 8 MHz reference clock. An asynchronous clock for the ARM is derived from the on-board oscillator using a crystal oscillating at its third harmonic.

While many A500s will operate at 14/7:8 MHz on the bench they become unreliable at elevated temperatures. It has been found that 12/6:8 MHz systems can perform reliably. However the EPROMs are one of the first components to fail and it is advised that the "ROM" speed is slowed from "200nS @ 8MHz" to "325nS @ 8MHz". For those of you with 100nS rather than 120nS DRAMs, your margins are better. Similarly some of the Video modes require different settings of the FIFO request parameter in MEMC.

WARNING WARNING WARNING

Don't push it too far - Your machine will corrupt data

WARNING WARNING WARNING

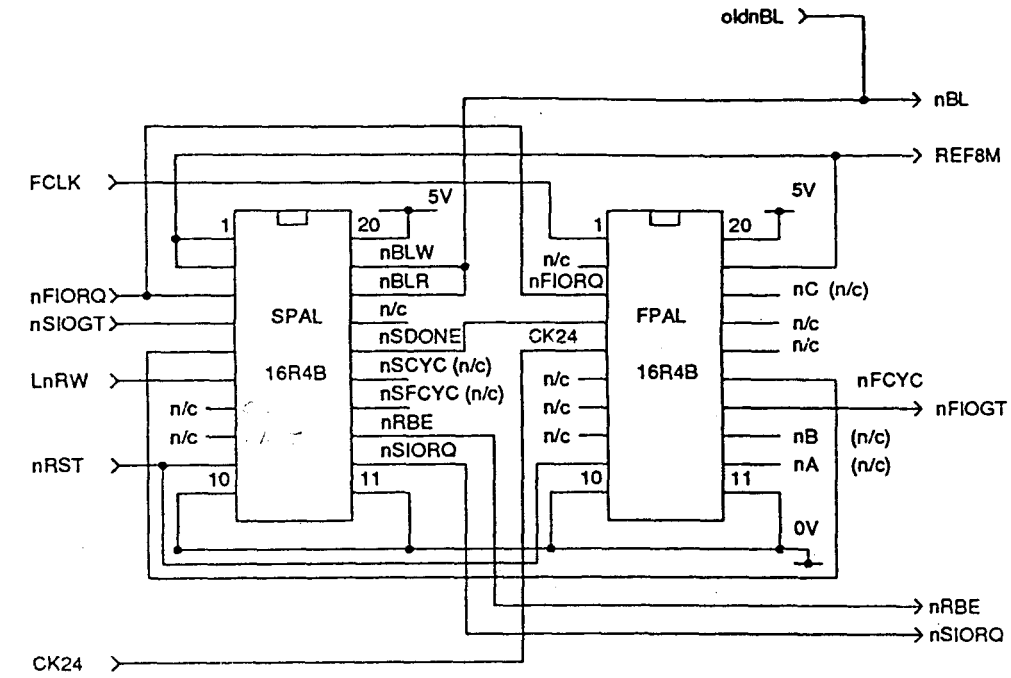


Figure 1: Circuit Diagram

1. Fast PAL

The "Fast PAL" implements the divide by three counter and a small state machine clocked on the rising edge of " FCLK". The asynchronous input "nSDONE" is clocked on the rising edge of "FCLK" to produce "nSIOGT". This potentially meta-stable output is not used until the next rising edge of "FCLK" by both the FPAL itself and MEMC. The fastest that the ARM cycle will complete, and therefore addresses become invalid, is 1.5 FCLK's after "nSDONE" is issued which occurs 0.5 of a REF8M before the I0 cycle completes.

2. Slow PAL

The "Slow PAL" implements logic to generate "nSIORQ", "nRBE", "nBLR" and "nBLW". The asynchronous output "nFCYC" from the fast PAL is used to initiate the IO cycle. While it would be faster to use a latched version of "nFIORQ", lack of PAL terms prevents this. This compromise adds about 0.75 of an "FCLK" to the synchronisation overhead. A synchronised term "nSFCYC" is produced which is clocked again to produce " nSCYC". This term then produces "nSIORQ". When "nSIORQ" and "nSIOGT" are both low on the rising edge of REF8M "nSYC" is cleared and "nSDONE" is set. "nSIORQ" is maintained until "nSDONE" is asserted and REF8M has gone low. "nSDONE" is cleared by the asynchronous input "nFCYC" which may cause "nSDONE" to become metastable, however it is not used again until the next clock edge by "nSCYC". "nSDONE" must be cleared quickly to ensure that the "Fast PAL" does not overrun the "Slow PAL" and initiate another JO cycle. The "nBLR" and "nBLW signals are generated as pseudo open drain outputs using a tri-state control. For write cycles "nBL" is asserted by the

fast PAL state transition in the first N-cycle and maintained until "nSDONE" is asserted and "nSIORQ" has gone high. For read cycles it is asserted at the end of the IO cycle and cleared by the fast state machine returning to idle. To ensure that the "nFCYC" signal correctly clears "nSDONE", "FCLK" must be less than two times " REF8M".

```
      1/fclk  >  output delay   metastability   MAX( MEMC setup | FPAL setup )
      1/sclk  >  output delay   metastability   SPAL setup
      1/fclk  >  0.5 * 1/sclk   SPAL setup
Rune for Arthut 1.2 *fx 162,&c3,2
Note:  To change the ROM speeds from BASIC
                                 @8MHz   @12MHz    @14MHz
        >SYS &1A,&00,&F0    :    450      300       250
        >SYS &1A,&50,&F0    :    325      200       185
        >SYS &1A,&A0,&F0    :    200      130       110
For A series MMI pal
                                          - k2 (delta - omega)
      failure  = (k1  * fclk * fdata ) e
                                          - 4.3E-9(delta - 34.5E-9)
      3.2E-9   = (E-7 * 12E6 * 3E6   ) e
      delta    = 36 ns
```

3. A500 Modifications

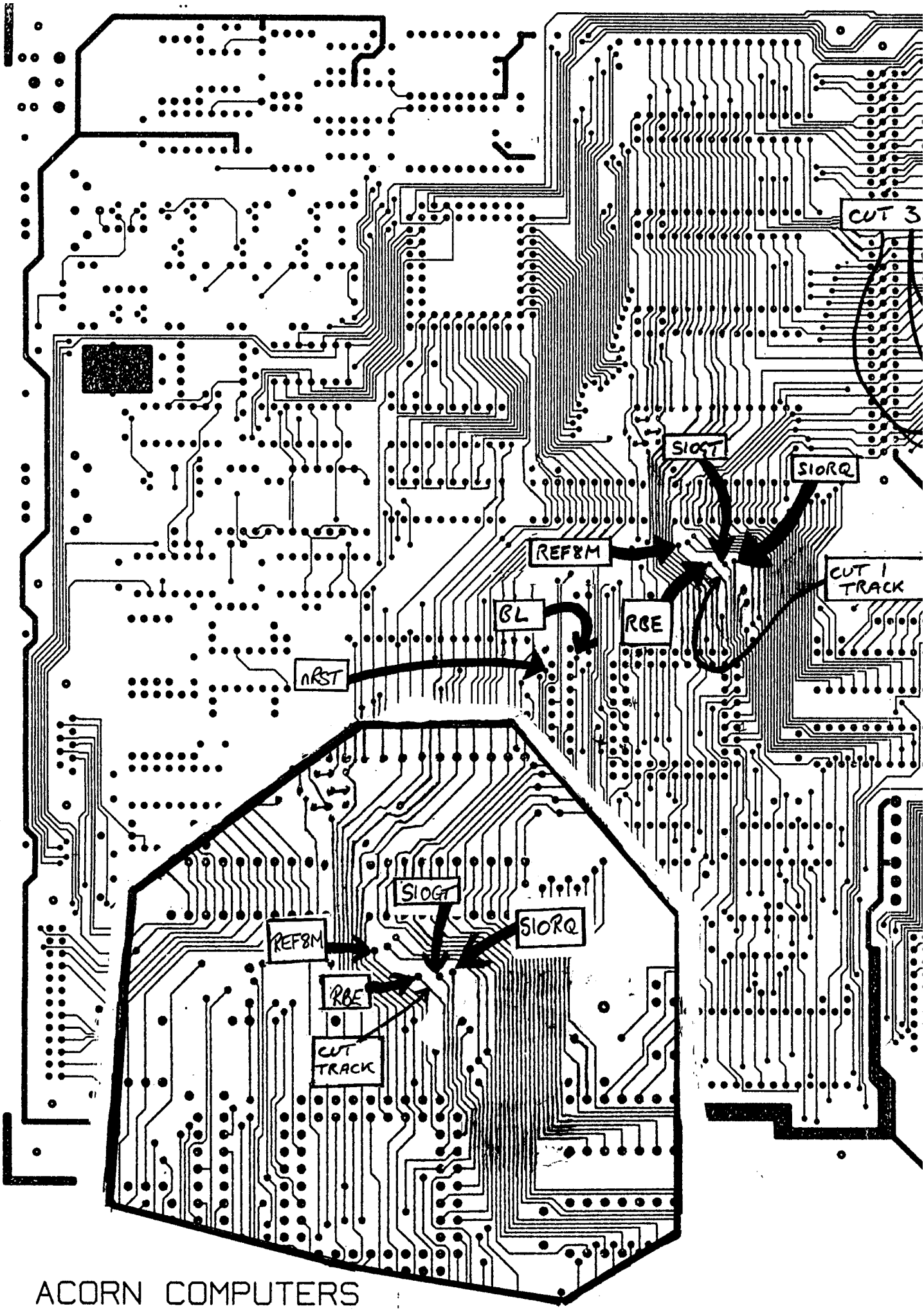
To implement this circuit perform the following modifications :-

Solder FastIOld on top of IC18 using pins 2, 6, 7, & 8 for extra stability (these are unused inputs), and pins 10 & 20 for power. Cut off pins 12, 13, 16, 17, & 18 (these are either unused outputs or feedback only terms). Pin 11 must be connected to ground.

Similarly, solder SlowIO1e on top of IC28 using pins 7, & 8 for extra stability and pins 10 & 20 for power. Cut off pins 14, 15, & 17. Pin 11 must be connected to ground.

The mod can be done without removing the board from the case, although the disc tray has to be taken off. The attached layout shows where the signals can be obtained, an enlargement is inset to make things clearer. Four tracks have to be cut, and all signals can be obtained from plated-through-holes. The existing pull-up on "nIOGT" is now connected to "nFlOGT", which does not require it, and a new pull-up must be fitted to "nSIOGT". This is best done between pins 4 and 20 of SlowIO1e. Similarly, a pull-up is required for BL and this can be fitted between pins 11 and 20 of IC29.

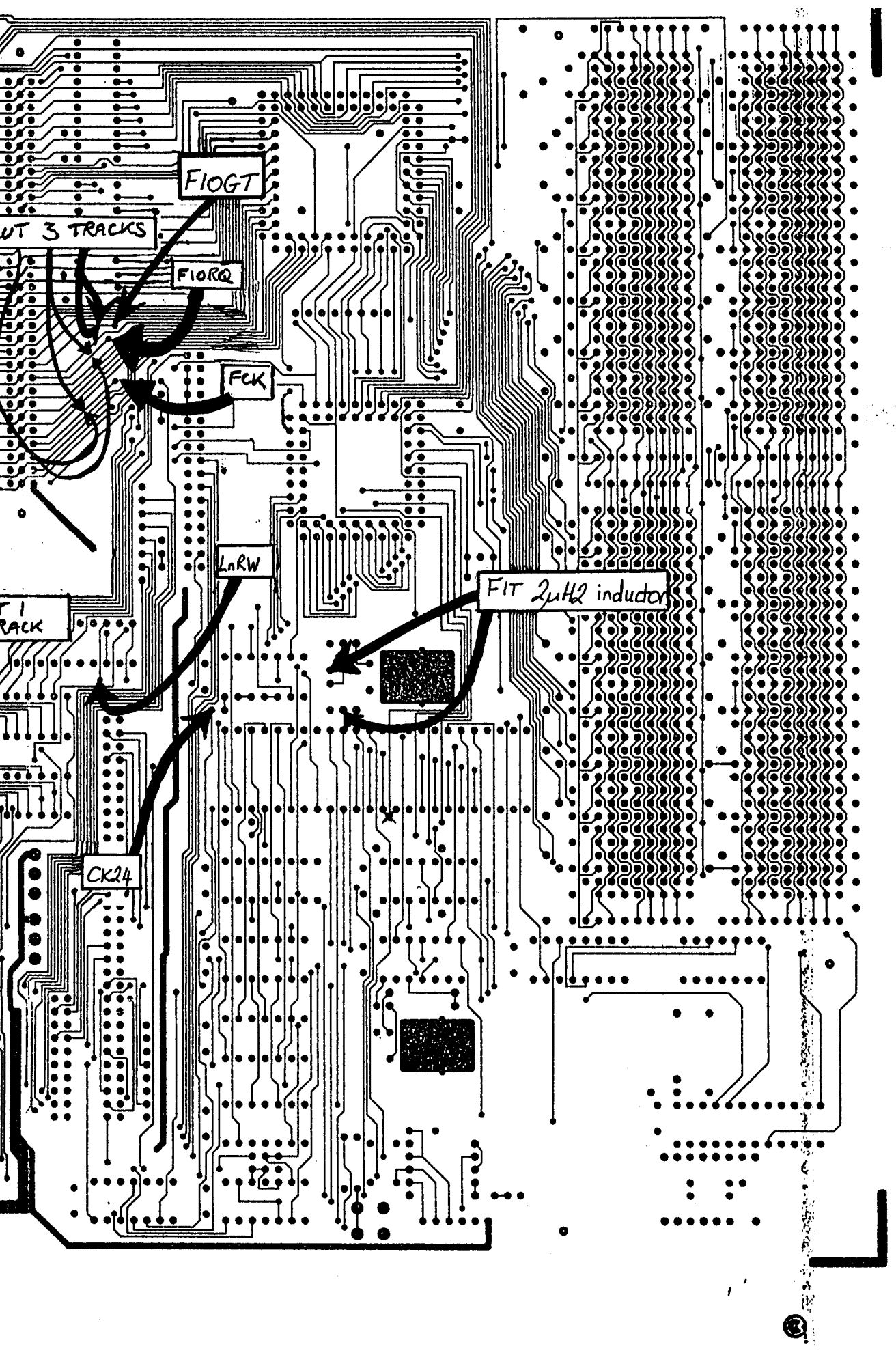
```
 REF8M      MEMC output unused  all circuits driven from new REF8M
nIORQ       MEMC output becomes nFIORQ  all  other circuits nSIORQ
nIOGT       MEMC input  becomes nFIOGT  all  other cicruits nSIOGT
nRBE        IOC  output unused  all circuits driven from new  nRBE
nBL         wire-ored to existing circuits ( twice nBLR and nBLW )
nRST        input
LnRW        input
CK24        input
LK1         break
fit         R100 = 220R   R101 = 100K   Rpullup for nSIOGT = 1K2
            Rpullup for BL = 2K2
            C32  = 10pF   C33  = 33pF
            X4   = (12MHz)
            inductor = 2uH2
```



ACORN COMPUTERS

SKB 0000 025A

COMPONENT SIDE


F1OGT
UT 3 TRACKS
F1ORQ
FCK
LnRW
FIT 2µH2 inductor
T 1
RACK
CK24