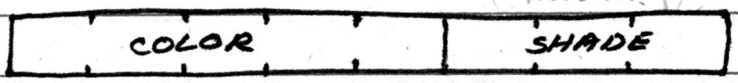
(6.)

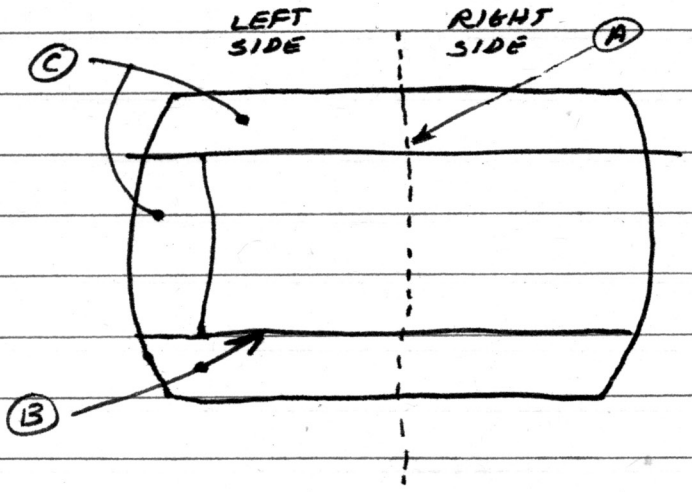
John Perkins
Vabeach Va
23451

IN LOOKING OVER PAST ARCADIANS, I HAVE NOTICED CONFUSION REGARDING THE FUNCTION OF THE VARIOUS I/O PORTS. MAYBE MY OWN RESEARCH WILL HELP — FOR OUTPUTS:

$S(0)$ = RIGHT REG 0	$S(4)$ = LEFT REG 0 } BC
$S(1)$ = RIGHT REG 1	$S(5)$ = LEFT REG 1
$S(2)$ = RIGHT REG 2	$S(6)$ = LEFT REG 2 } FC
$S(3)$ = RIGHT REG 3	$S(7)$ = LEFT REG 3

(intensity)





A - LEFT/RIGHT BOUNDARY

B - MEMORY PARTITION (ONLY MEMORY ABOVE THE PARTITION WILL BE DISPLAYED)

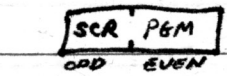
C - BACKGROUND AREA (OUTSIDE MEMORY AREA)

THE SCREEN IS LOGICALLY DIVIDED INTO A LEFT AND RIGHT SIDE WITH A MOVABLE BOUNDARY. $S(0)$ - $S(3)$ CONTROLS THE COLOR AND SHADE RIGHT OF THE BOUNDARY WHILE $S(4)$ - $S(7)$ DOES THE SAME FOR THE LEFT SIDE. BALLY BASIC CONTINUALLY SETS $S(4)$ AND $S(5)$ TO THE COLOR/SHADE DEFINED BY BC AND $S(6)$ AND $S(7)$ TO THE COLOR/SHADE DEFINED BY FC.

TRYING TO SET THEM INDEPENDENTLY IS FUTILE WHILE BASIC IS IN CONTROL. HOWEVER $S(\phi)$ THROUGH $S(3)$ CAN BE SET TO 4 DIFFERENT COLOR/SHADES AND DISPLAYED BY MOVING THE LEFT/RIGHT BOUNDARY $S(9)$ TO A POSITION WHERE THE RIGHT SIDE IS VISABLE.

TRY THIS PROGRAM:

```
10 S(Ø) = 3Ø [SET BACKGROUND BLUE]
20 S(1) = 85 [RED]
30 S(2) = 153 [GREEN]
40 S(3) = 125 [GOLD]
50 S(9) = Ø
RUN
```

NOTICE THAT THREE COLORS ARE DISPLAYED — BLUE (VIOLET) BACKGROUND GREEN PROGRAM LISTING AND RED "GARBAGE" AT THE TOP OF THE SCREEN. MOVING THE BOUNDARY $S(9)=\phi$ ALLOWS ONE TO SEE THE STORED PROGRAM. THE PROGRAM IS ACTUALLY STORED RIGHT IN SCREEN MEMORY USING THE EVEN BIT POSITIONS. 

EACH PIXEL EQUATES TO TWO BITS OF MEMORY — 4 PIXELS TO AN 8 BIT BYTE. THE TWO BITS OF EACH PIXEL CAN HAVE 4 REPRESENTATIONS

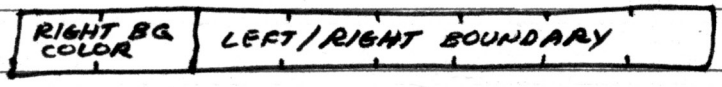
00 = $S(\phi)$ LEFT -OR- $S(\phi)$ RIGHT
01 = $S(5)$ LEFT -OR- $S(1)$ RIGHT
10 = $S(6)$ LEFT -OR- $S(2)$ RIGHT
11 = $S(7)$ LEFT -OR- $S(3)$ RIGHT

BALLY BASIC SETS THE SCREEN BOUNDARY ALL THE WAY RIGHT [redacted] SO THAT ONLY THE LEFT REGISTERS

S(4) - S(7) ARE USED. SINCE IT ALSO SETS S(4) AND S(5) TO BC AND S(6) AND S(7) TO FC, ONLY THE ODD BITS OF MEMORY SHOW ON THE SCREEN.

(A 00 IS THE SAME COLOR AS A 01)
(A 10 IS THE SAME COLOR AS A 11)

BY STORING THE PROGRAM IN THE EVEN BITS IT CAN OCCUPY SCREEN MEMORY (AS EVERY OTHER BIT) AND YET BE INVISIBLE. HOWEVER BY MOVING THE SCREEN BOUNDARY S(9) ALL THE WAY LEFT, THE RIGHT SIDE REGISTERS S(0) - S(3) ARE USED. SINCE THE PROGRAM SETS THESE TO 4 DIFFERENT COLORS THE STORED PROGRAM BECOMES "VISIBLE" AS GARBAGE AT THE TOP OF THE SCREEN. THE FOURTH COLOR WILL BE VISIBLE AS THE GO KEY IS PRESSED A FEW TIMES SCROLLING THE TEXT UP INTO THE STORED PROGRAM AREA. BY USING OTHER COLORS THE CONTRAST MAY BE BETTER. NOW SET S(9) = 5 AND SEE 6 COLORS - BLACK ON WHITE TO THE LEFT OF THE BOUNDARY AND VIOLET, RED, GREEN, AND GOLD ON THE RIGHT SIDE. ACTUALLY S(9) HAS TWO FUNCTIONS:



THE LEAST SIGNIFICANT 6 BITS SET THE BOUNDARY POSITION (4 PIXELS OR ONE MEMORY BYTE PER UNIT). THE MOST SIGNIFICANT 2 BITS CHOOSE THE COLOR REGISTER ASSOCIATED WITH THE LEFT SIDE BACKGROUND AND THE RIGHT SIDE BACKGROUND. TRY S(9) = 135.

S(10) CONTROLS HOW MANY RASTER LINES ARE DISPLAYED FROM MEMORY AS OPPOSED TO BEING [redacted] PART OF THE BACKGROUND. S(10) = 204 DISPLAYS ALL OF MEMORY (RAM) ALLOWING VISUAL INSPECTION OF THE RUNNING PROGRAM (NOT THAT THERE'S ANY USEFUL INFO THERE)

I WILL INCLUDE MORE INFORMATION ON THE OTHER PORTS LATER. INCIDENTALLY, S(13) DOES WIERD THINGS BECAUSE YOU ARE MODIFYING AN INTERRUPT VECTOR CAUSING LOSS OF PROGRAM CONTROL WHEN THE TIMER INTERRUPT OCCURS (EVERY 1/60TH SECOND). THE ONLY RELIABLE VALUES ARE OF NOT MUCH WORTH. — [redacted]