SCIENCE RESEARCH COUNCIL
RUTHERFORD & APPLETON LABORATORIES

COMPUTING DIVISION

DISTRIBUTED COMPUTING NOTE 387

PERQ
TECHNICAL NOTE 8

issued by
R W Witty

Rotating Quarters Program

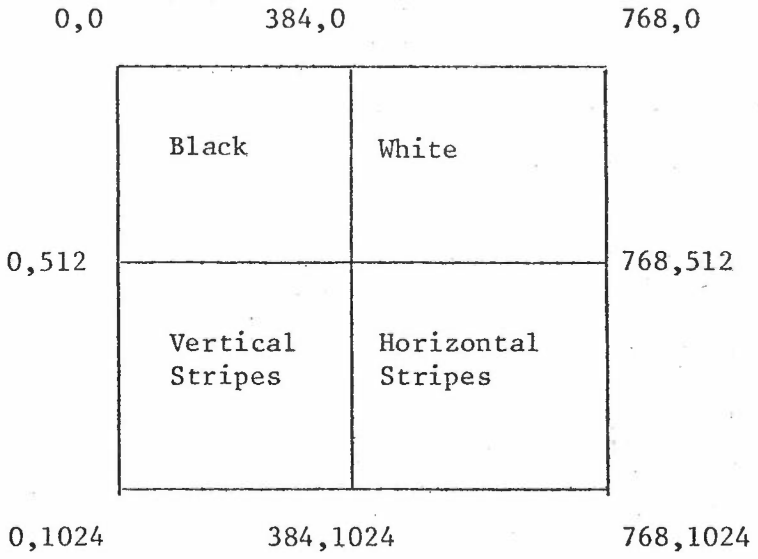
1 April 1981

Distribution:

F R A Hopgood
R W Witty
D A Duce
C J Prosser
W P Sharpe
J Brown
I D Benest
A S Williams
C Wadsworth
RL Support/PERQ/General file

1. INTRODUCTION

This little demonstration program draws four distinct patterns, one in each quarter of the screen and then rotates them counter-clockwise.



2. ARRAY DECLARATIONS

Note that when handling the screen as a 2-dimensional array, the horizontal, x, pixel within scan line is the 2nd dimension which is the 'fastest moving'. The 1st dimension is the scan line.

The screen should be declared as

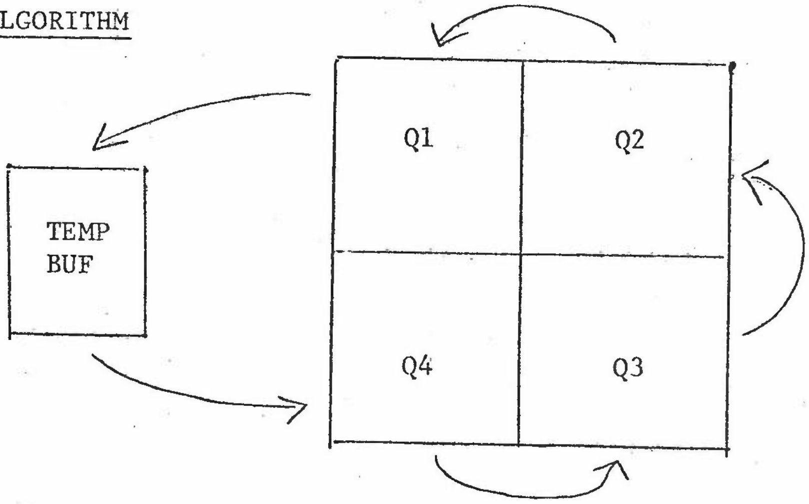
[1 ... 1024,1 ... 768]

but the compiler gives an error message

"Implementation Restriction"

so the program cheats to allow compilation. I do not know why it works!

3. ALGORITHM



The quarters are rotated counter-clockwise. A temporary buffer is needed to hold Q1.

```
program move4(input,output);

imports screeen from screen;
imports raster  from raster;
imports memory  from memory;

type
  buf = packed array[1..512,1..384] of boolean;

  scr = packed array[1..512,1..768] of boolean;<cheat s.b.*2>

  screanptr = ^scr;




var
  i,j,segno : integer;
  scrptr    : screanptr;
  bufptr    : ^buf;


begin

  reset(input); rewrite(output);

  <create 4 quarter pics>
  scrptr:=makeptr(screenseg,0,screanptr);
  for j:=1 to 512 do
  for i:=1 to 384 do
  begin
    scrptr^[j,i]:=true;
    scrptr^[j,i+384]:=false;

    if odd(j div 4) then scrptr^[j+512,i]:=true
                    else scrptr^[j+512,i]:=false;

    if odd(i div 4) then scrptr^[j+512,i+384]:=true
                    else scrptr^[j+512,i+384]:=false;
  end;

  <set up temp buffer>
  CreateSegment(segno,48,1,48);
  new(segno,4,bufptr);
  <rotate 4 quarters >
  for i:=1 to 100 do
  begin
    <copy q1 to buf>
    Rasterop(RRpl,
             384,
             512,
             0,
             0,
             24,
             bufptr,
             0,
             0,
             48,
             scrptr);

    <copy q2 to q1>
    Rasterop(RRpl,
             384,
             512,
             0,
             0,
             48,
             scrptr,
             384,
             0,
             48,
             scrptr);

    <copy q3 to q2>
    Rasterop(RRpl,
             384,
             512,
             384,
             0,
             48,
             scrptr,
             384,
             512,
             48,
             scrptr);

    <copy q4 to q3>
    Rasterop(RRpl,
             384,
             512,
             384,
             512,
             48,
             scrptr,
             0,
             512,
             48,
             scrptr);

    <copy buff to q4>
    Rasterop(RRpl,
             384,
             512,
             0,
             512,
             48,
             scrptr,
             0,
             0,
             24,
             bufptr);


 <waste time>
 for j:=1 to 20000 do segno:=segno;

 end;
end.
```