Standard CPC display (excluding hsync/vsync etc)

R0 = 63 (value – 1)

R1 = 40 (value)

R4 = 38 (value – 1)

R6 = 25 (value)
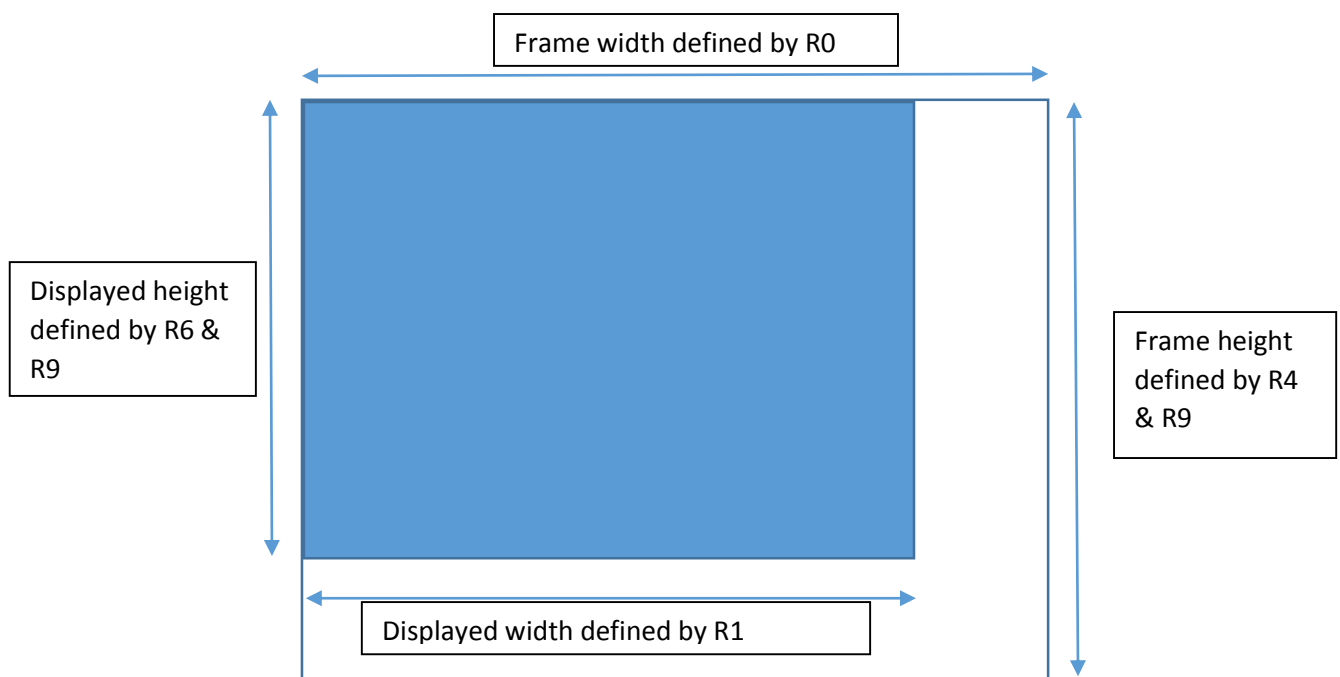
R9 = 7 (value-1)

Frame width of 64 equals horizontal frequency of 15Khz which matches the television/monitor frequency.
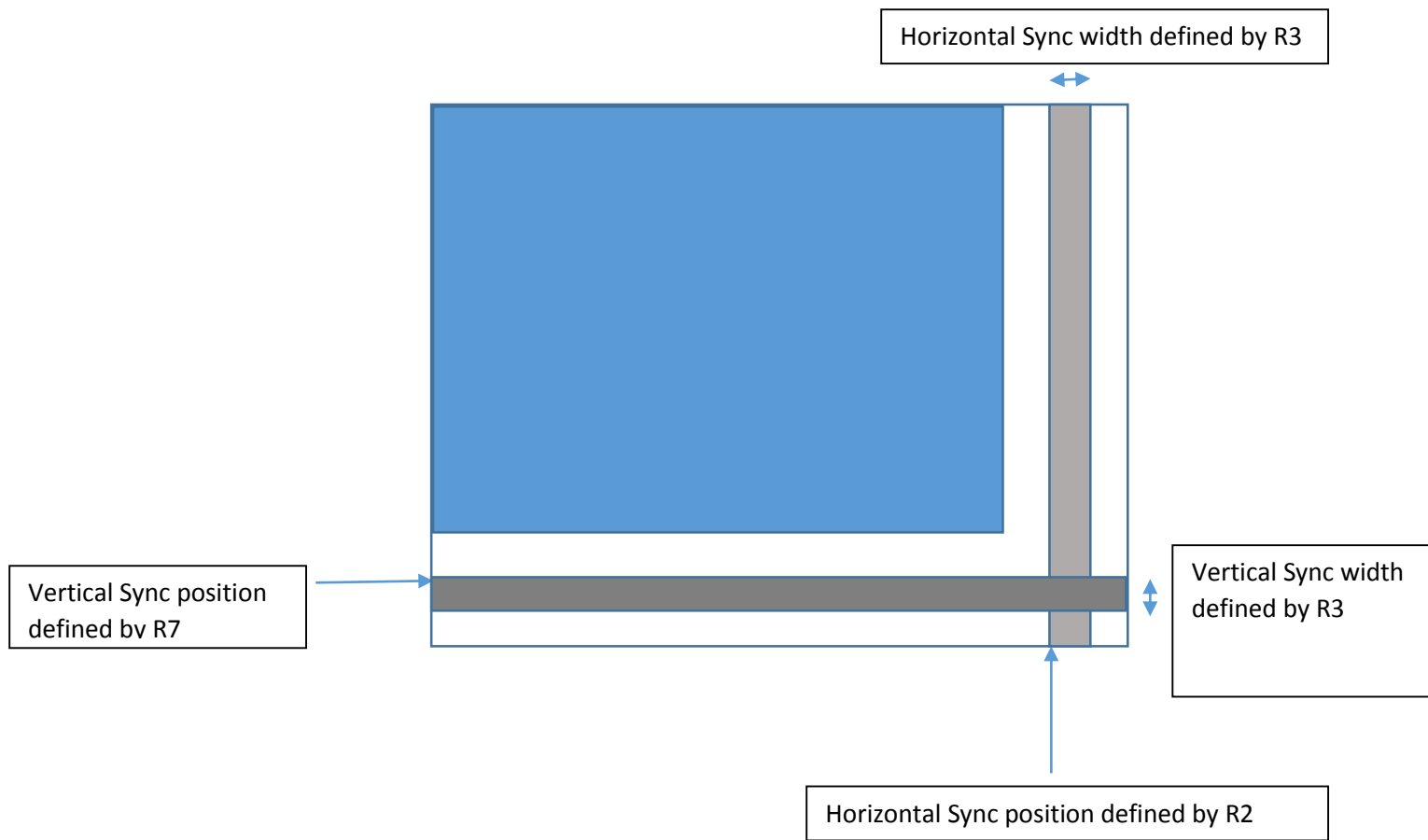
Displayed in scanlines is R6*(R9+1) = 200 lines

Frame height is (R4+1)*(R9+1) = 312.

312 lines is for 50Hz.

```
                  ┌─────────────────────────────────┐
                  │   Frame width defined by R0     │
                  └─────────────────────────────────┘
 ┌──────────────┐
 │ Displayed    │                                     ┌──────────────┐
 │ height       │                                     │ Frame height │
 │ defined by   │                                     │ defined by   │
 │ R6 & R9      │                                     │ R4 & R9      │
 └──────────────┘                                     └──────────────┘
                  ┌─────────────────────────────────┐
                  │  Displayed width defined by R1  │
                  └─────────────────────────────────┘
```

Standard CPC display (showing hsync and vsync)

Horizontal Sync width defined by R3

Vertical Sync position defined by R7

Vertical Sync width defined by R3

Horizontal Sync position defined by R2

Standard CPC display how monitor displays it.

When the start of the horizontal sync is seen by the monitor it starts the next line. This means border is effectively split between left and right side of the display. Border to the left of the HSYNC is the right border, border to the right of the HSYNC is the left border.

When the start of the vertical sync is seen by the monitor it starts the next frame. This means border is effectively split between top and bottom of the display. Border above the VSYNC is the bottom border, Border below the VSYNC is the top border.
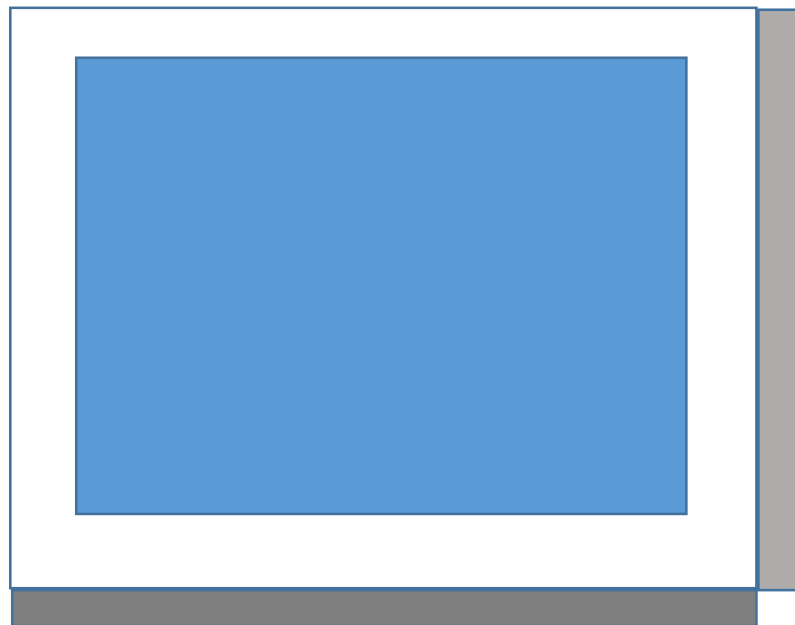
Moving HSYNC

Decreasing R2 moves HSYNC to the left. On the monitor the displayed area moves to the right. ("Left" border is larger, "right" border is smaller)

Increasing R2 moves HSYNC to the right. On the monitor the displayed area moves to the left. ("Left border" is smaller, "right" border is larger)

Moving VSYNC

Decreasing R7 moves VSYNC up. On the monitor the displayed area moves down. ("Top" border is larger, "bottom" border is smaller)

Increasing R7 moves VSYNC down. On the monitor the displayed area moves up. ("Top border" is smaller, "Bottom" border is larger)

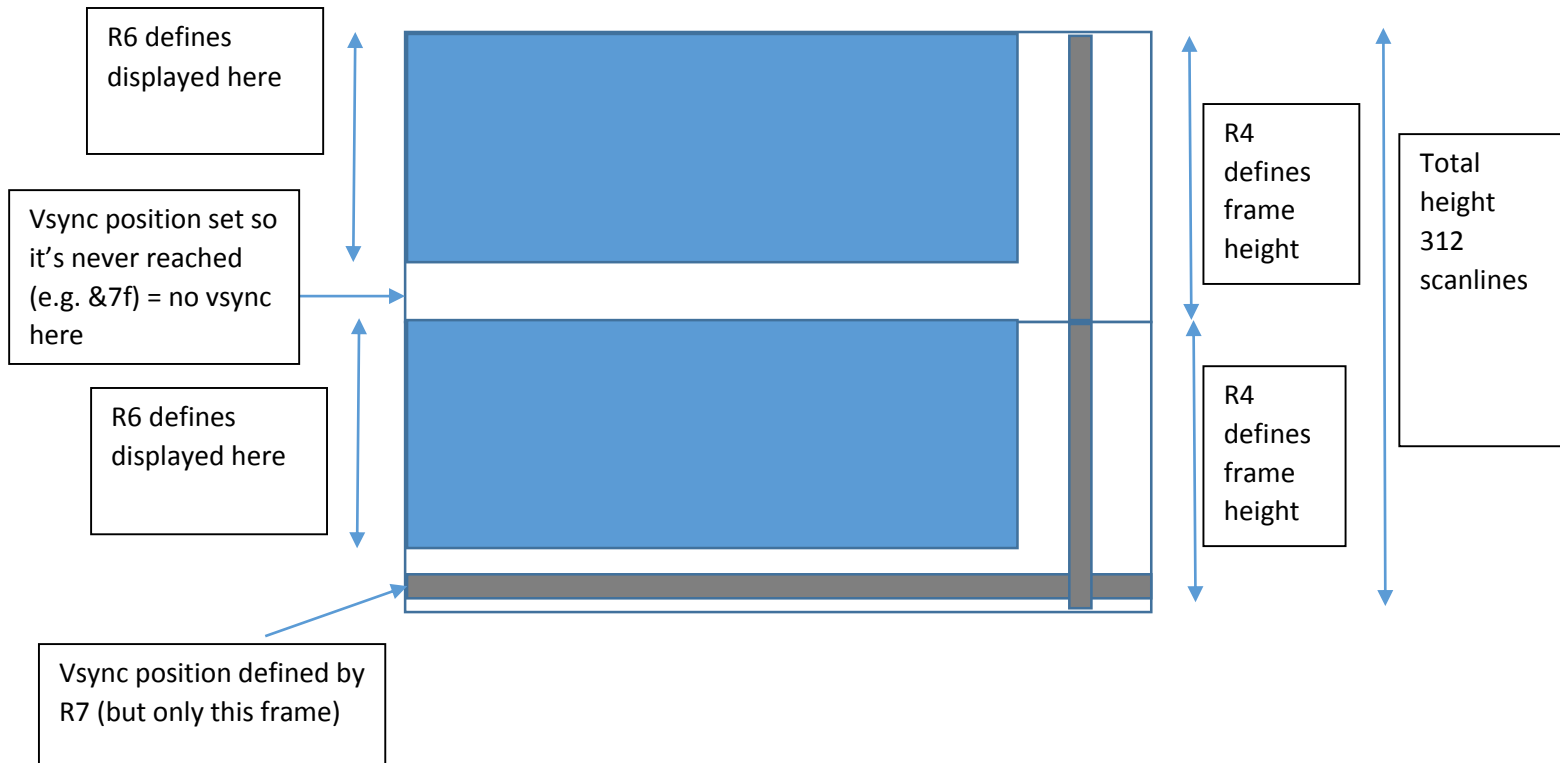## Overscan ("32KB method")

If we change:

- HSYNC position using R2
- VSYNC position using R7
- Horizontal displayed using R1
- Vertical displayed using R6

We get a larger display that fills the monitor. Using R12/R13 we can make CPC address 32KB of RAM. We get 32KB overscan.

Rupture technique

- We make multiple frames using R4 to define the height of each.
- Set Vsync position so it is not triggered until we want it to.
- IMPORTANT: Make sure all the frames add up to 312 scan-lines so that display is stable.

R6 defines displayed here

Vsync position set so it's never reached (e.g. &7f) = no vsync here

R6 defines displayed here

Vsync position defined by R7 (but only this frame)

R4 defines frame height

R4 defines frame height

Total height 312 scanlines

Normally R6 is set so that we don't see border between each frame.

If we don't set the total frame height to be 312 lines then some monitors will not display it correctly.
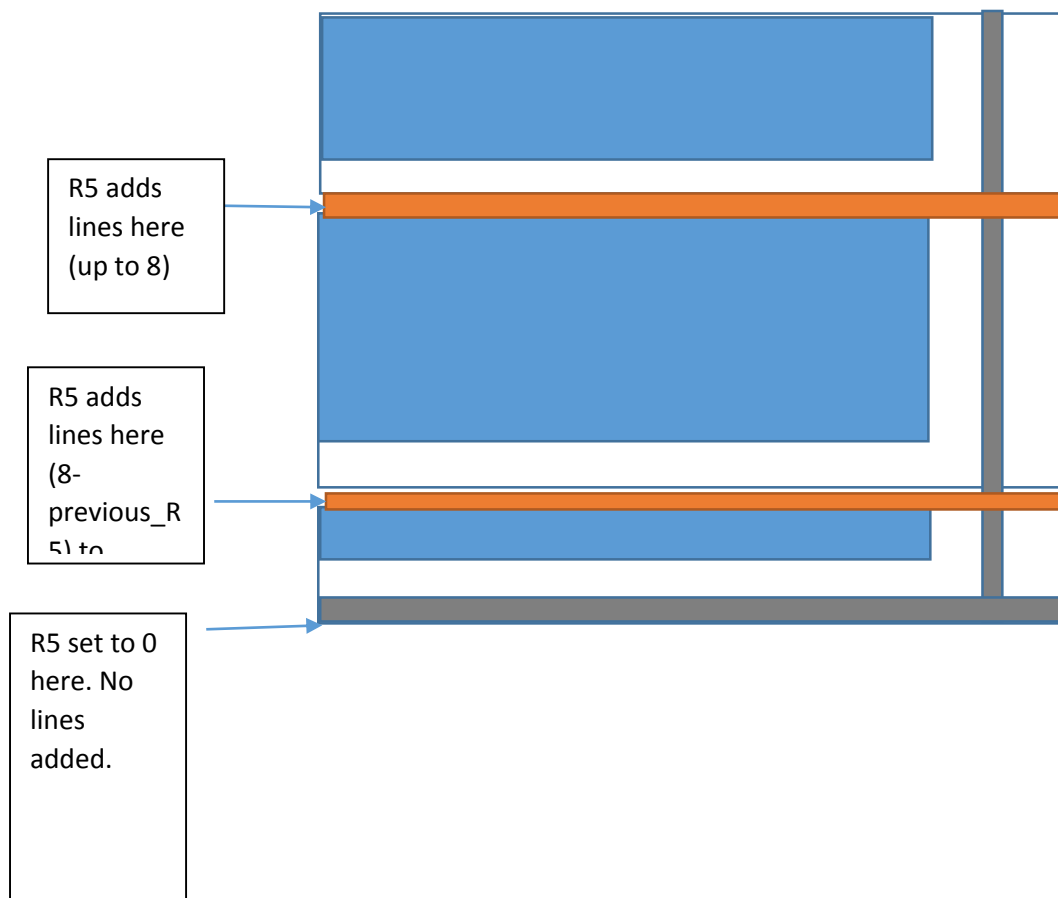
NOTE: Rupture requires good timing. We can split the screen as we want adding as many frames as we want with differing heights. Each frame can have it's own R12/R13 values so we can scroll them independently or have one static and one scrolling.

For example you want 3 sections, with heights 8 and 12. Remainder is 30-29 which leaves last section as 19. When programming R4 use 8-1, 12-1 and 19-1.

MORE will be added here to show interrupt positions.

Vertical scrolling using R5

- We need to use rupture.
- 3 frames works best
-

R5 adds lines here (up to 8)

R5 adds lines here (8-previous_R 5) to

R5 set to 0 here. No lines added.

- For the top section, we increase/decrease R5. Increase to move 2nd section down, decrease to move 1st section up. We use at most (R9+1) scanlines because the normal scroll is (R9+1) scanlines.
- For the middle section, we compensate for the extra lines we added. The value here is (R9+1)-value. The total of R5 for the first and middle sections add up to 8.
- We compensate to make the last section start always at the same scanline on the screen and so that the VSYNC we trigger always starts at the same scanline on the screen which is important to maintain stability.

MORE will be added here.