Laboratorio 6:

FSM con flags

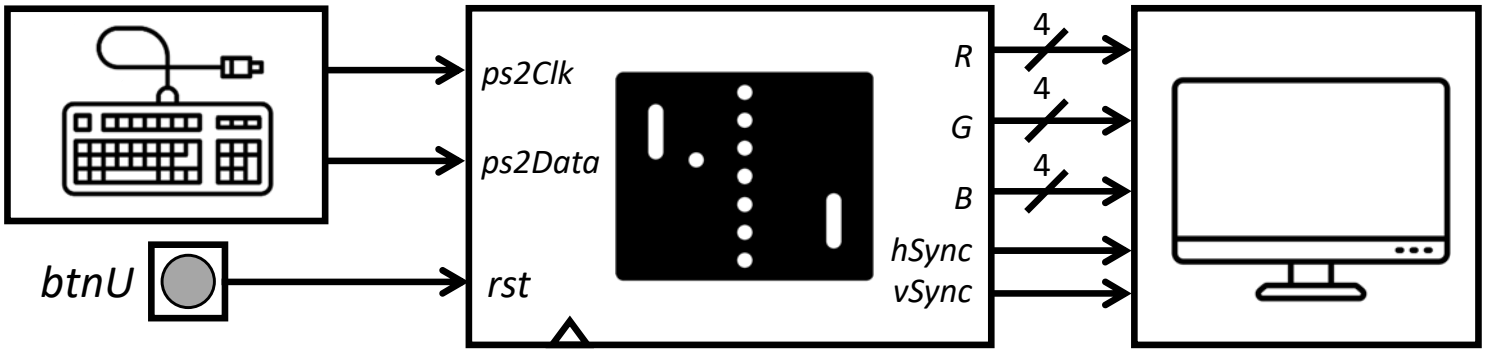
visualización en un monitor VGA

Diseño automático de sistemas

José Manuel Mendías Cuadros
Dpto. Arquitectura de Computadores y Automática
Universidad Complutense de Madrid

Presentación

- Diseñar el clásico juego del pong:
  - El juego se visualizará en blanco y negro sobre un monitor.
    - Las señales de vídeo se generarán en tiempo real, es decir, sin usar memoria de refresco.
  - Los jugadores utilizarán el teclado:
    - El jugador izquierdo usará la tecla Q para subir la raqueta y la tecla A para bajarla.
    - El jugador derecho usará la tecla P para subir la raqueta y la tecla L para bajarla.
    - El inicio de un juego lo indicará una pulsación de la tecla SPC.
  - Dispondrá de reset síncrono.



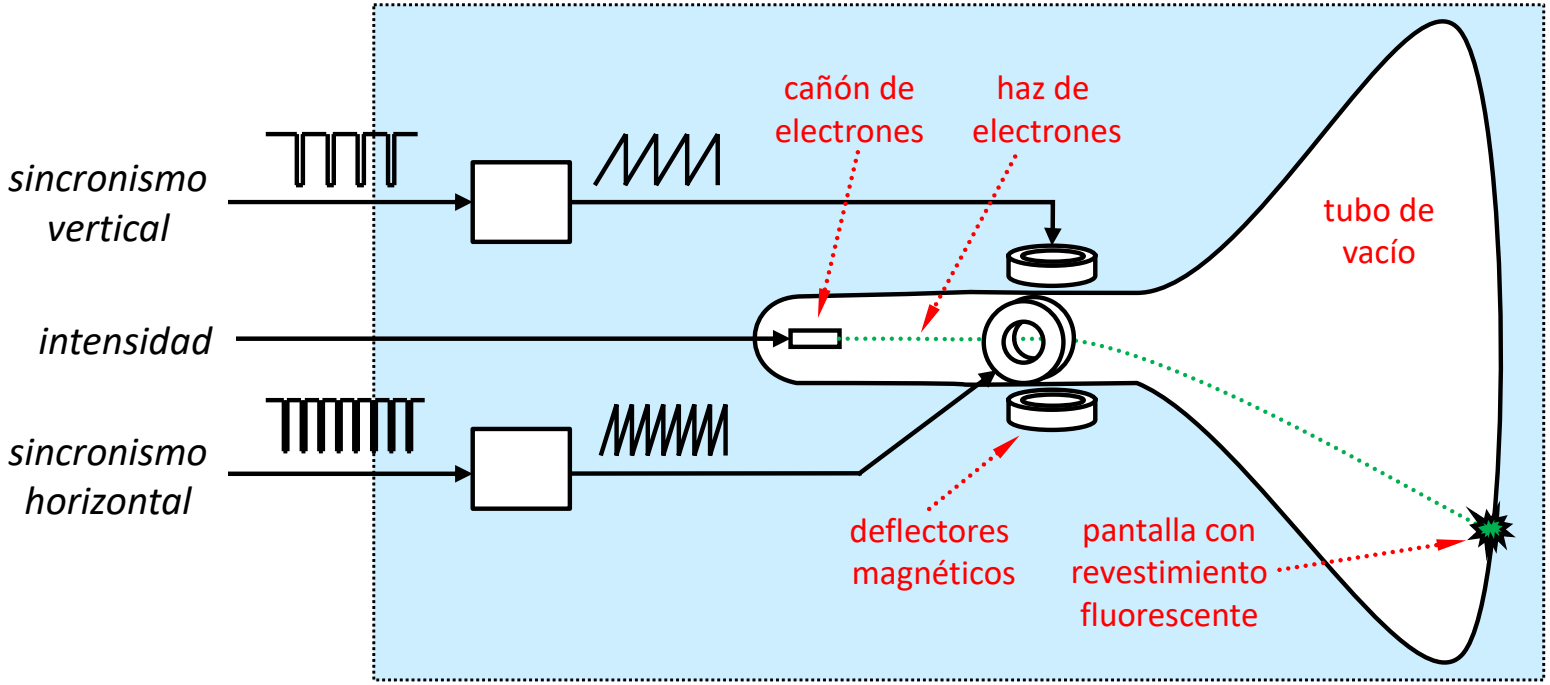
http://www.pong-story.com/intro.htm

J.M. Mendías
2024

Estándar de vídeo VGA

algo de historia: monitores CRT (i)

- El estándar VGA nació en la época de los monitores con tecnología de de tubo de rayos catódigos (CRT) formados por:
  - Un tubo de vacío de forma piramidal cuya base está recubierta de un material fluorescente.
  - Un filamento que produce un haz de electrones (y varios para pantallas en color).
  - Un par de bobinas deflectoras perpendiculares que permiten modificar la trayectoria del haz de electrones.

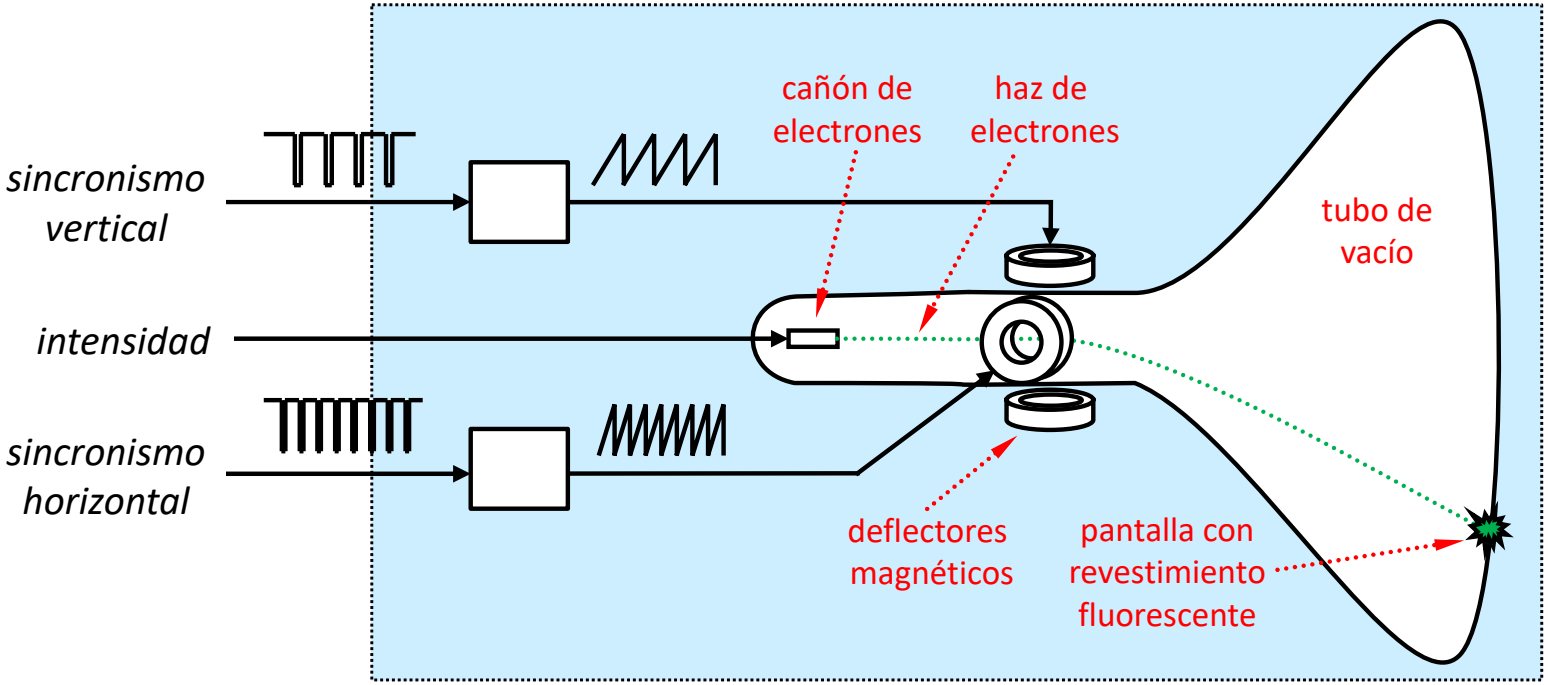


J.M. Mendías
2024

Estándar de vídeo VGA

algo de historia: monitores CRT (i)

- El choque del haz de electrones con el material fluorescente hace que éste se ilumine:
  - El tipo de material fluorescente determina el color de la luz.
  - La densidad del haz de electrones determina la intensidad de la luz.
  - La desviación inducida por las bobinas determina el lugar de impacto del haz.
  - El tamaño del punto de impacto determina la resolución de la pantalla.

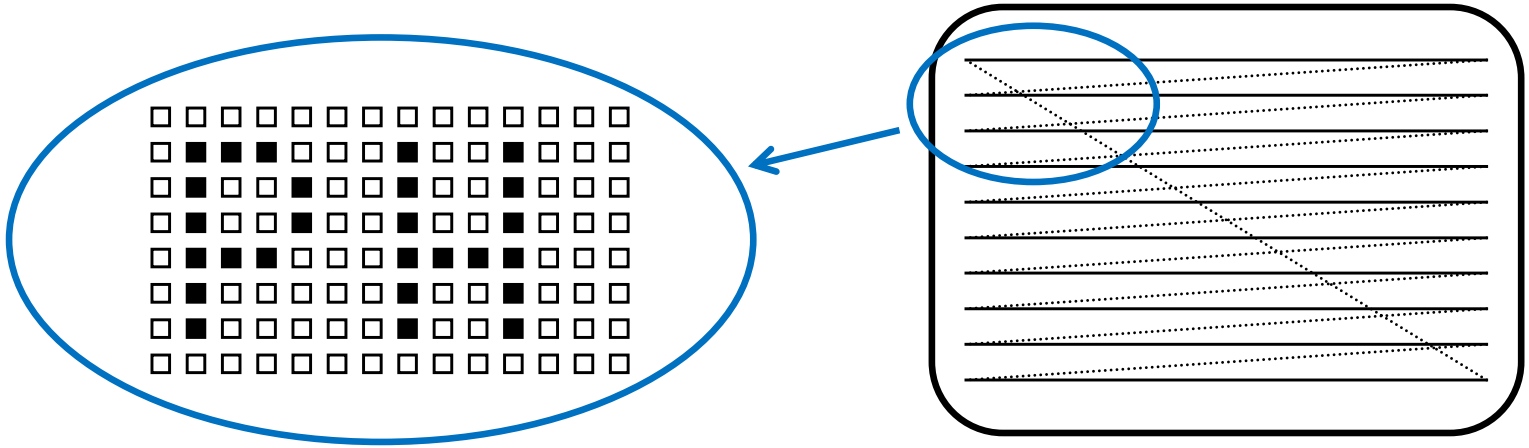


J.M. Mendias
2024

Estándar de vídeo VGA

algo de historia: monitores CRT de barrido (i)

- En los CRT de barrido (raster scan) el haz barre periódicamente la pantalla de una forma sistemática:
  - Comenzando por la esquina superior izquierda, recorre horizontalmente cada una de las filas de pixels.
  - Cuando alcanza el final de la fila, apaga momentáneamente el cañón y se coloca al comienzo de la siguiente fila (horizontal retrace).
  - Cuando todas las filas han sido recorridas y se ha alcanzado la la esquina inferior derecha, se apaga el cañón y se retorna al comienzo (vertical retrace).

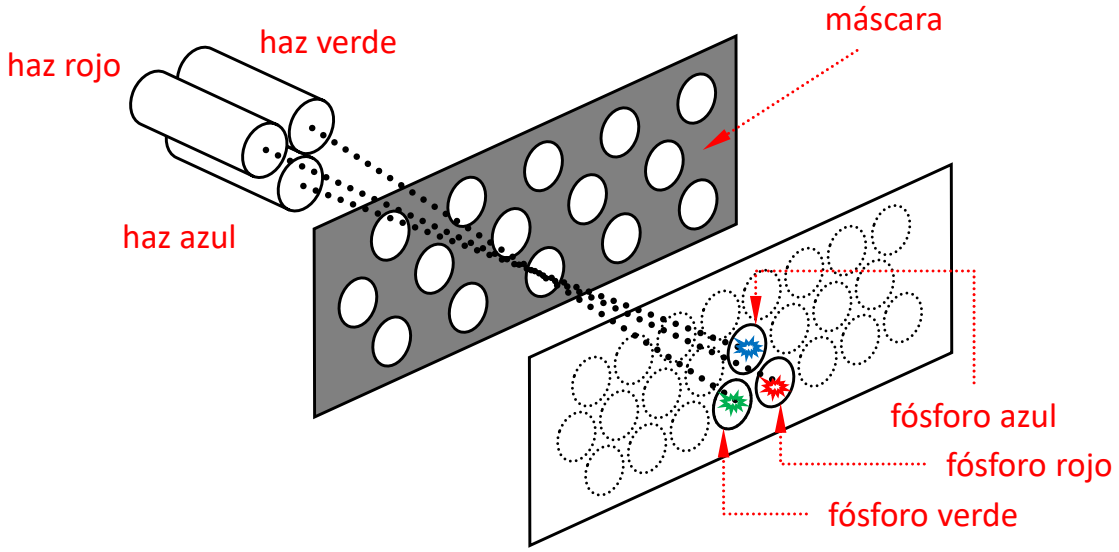


J.M. Mendías 2024

Estándar de vídeo VGA

algo de historia: monitores CRT de barrido (ii)

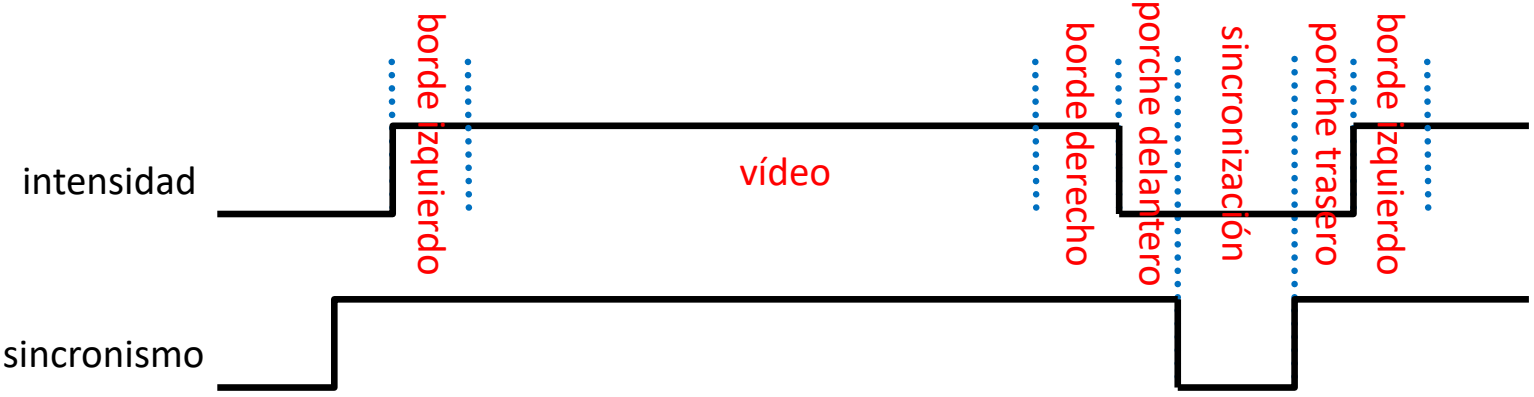
- Para controlar el barrido, el interfaz envía a la pantalla tres señales:
  - Sincronización horizontal: señal digital que marca el comienzo de cada línea.
  - Sincronización vertical: señal digital que marca el comienzo de cada pantalla (frame).
  - Intensidad: señal analógica que regula la intensidad del haz de electrones.
- En un monitor CRT en color:
  - La pantalla se cubre con tríos de puntos de fósforo de colores diferentes (rojo, verde, azul) colocados muy próximos.
  - Existen 3 cañones de electrones controlados por señales de intensidad distintas.



Estándar de vídeo VGA

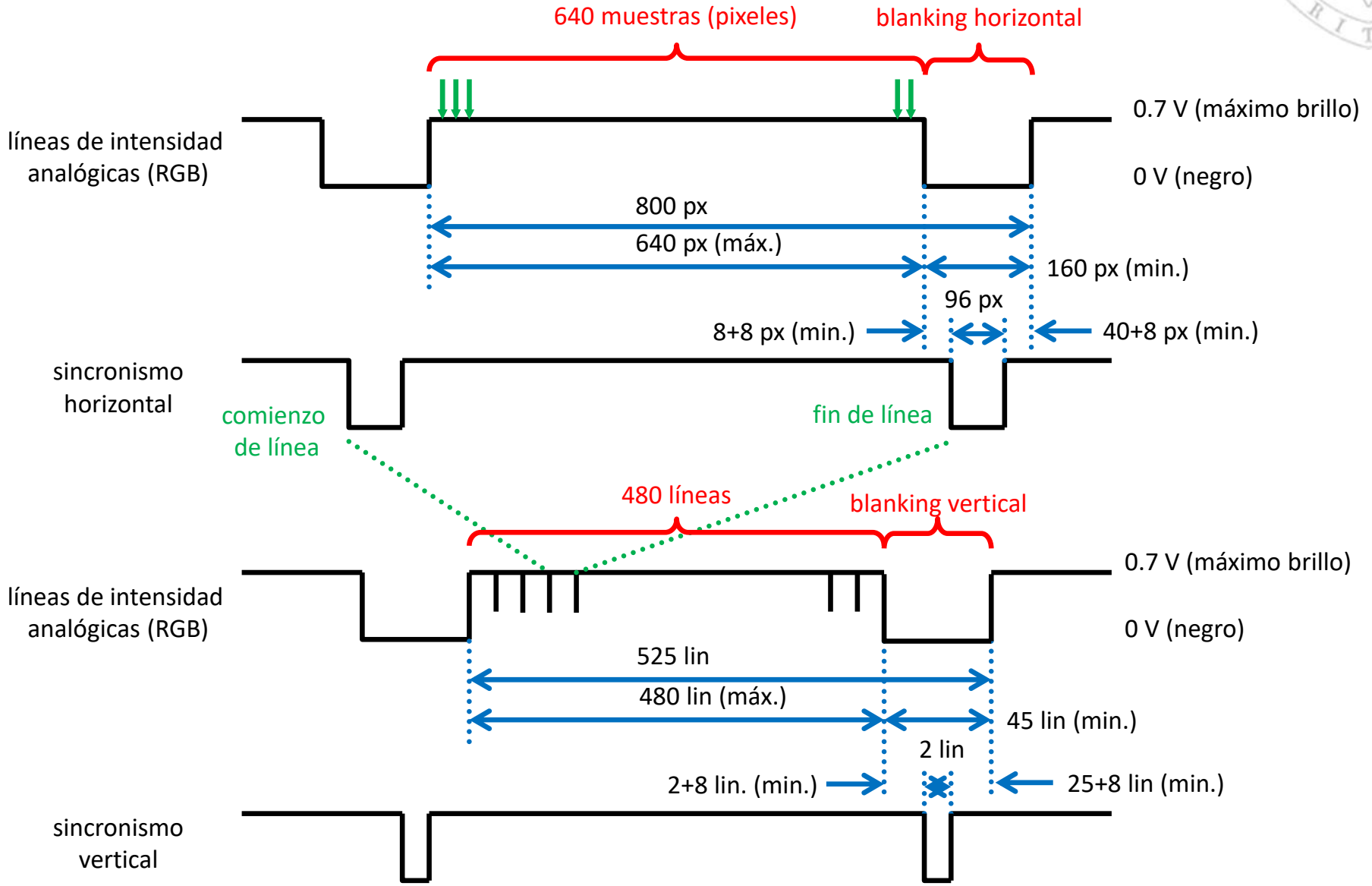
640×480 píxeles (i)

- Frecuencias:
  - Del reloj de muestreo del monitor (dot clock frequency) = 25,175 MHz
  - De sincronización horizontal (line frequency) = 31,469 KHz
  - De sincronización vertical (field frequency) = 59,94 Hz
- 800 píxeles por línea (640 visibles) = 25,175 MHz ÷ 31,496 KHz
  - 8 píxeles de porche delantero
  - 96 píxeles de sincronización horizontal
  - 40 píxeles de porche trasero
  - 8 píxeles de borde izquierdo
  - 640 píxeles de vídeo
  - 8 píxeles de borde derecho
- 525 líneas por pantalla (480 visibles) = 31,496 KHz ÷ 59,94 Hz
  - 2 líneas de porche delantero
  - 2 líneas de sincronización vertical
  - 25 líneas de porche trasero
  - 8 líneas de borde superior
  - 480 líneas de vídeo
  - 8 líneas de borde inferior



Estándar de vídeo VGA

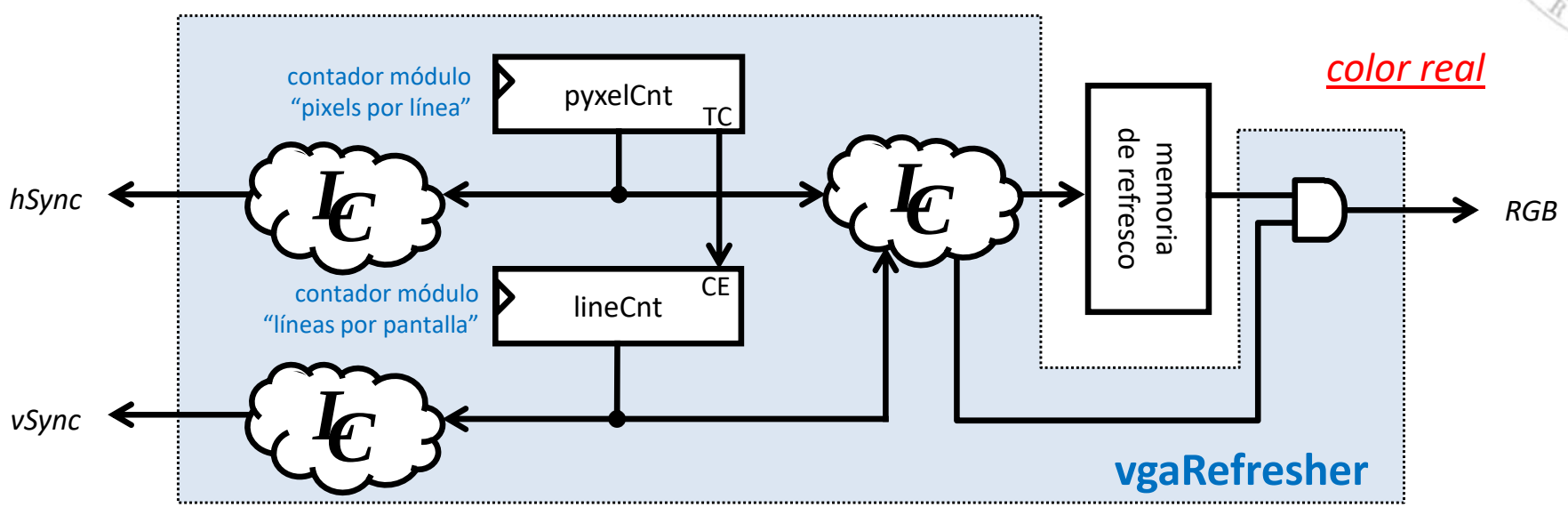
640×480 píxeles (ii)

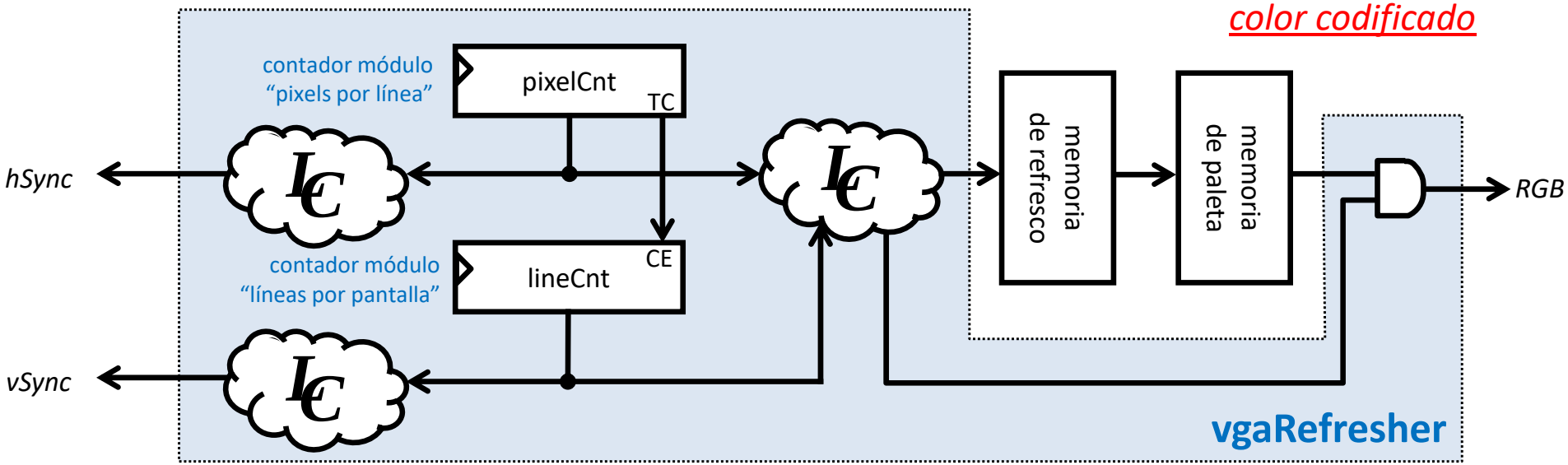


J.M. Mendías
2024

Interfaces de vídeo

estructura

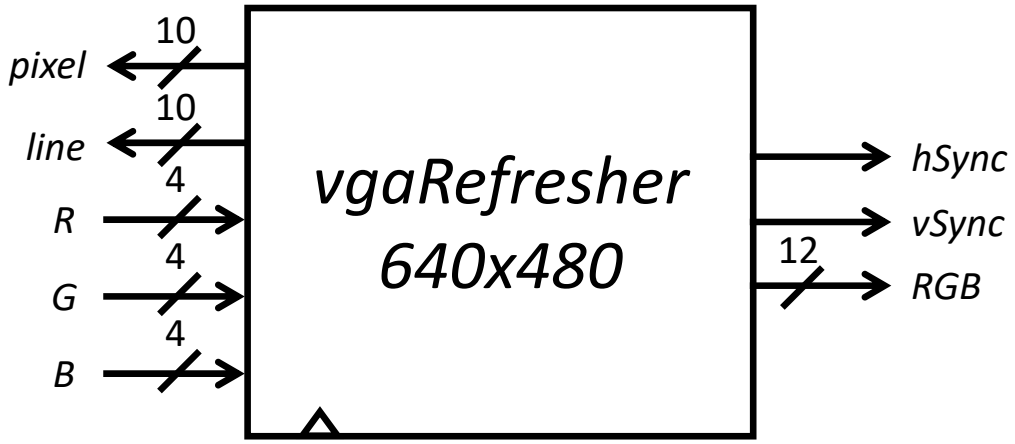




Interfaz VGA

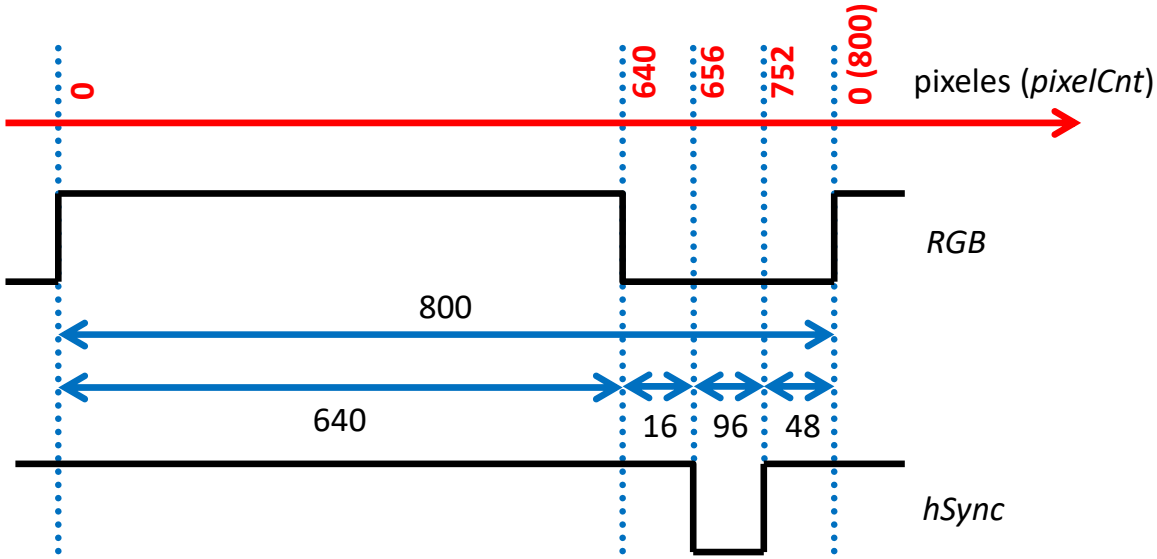
presentación

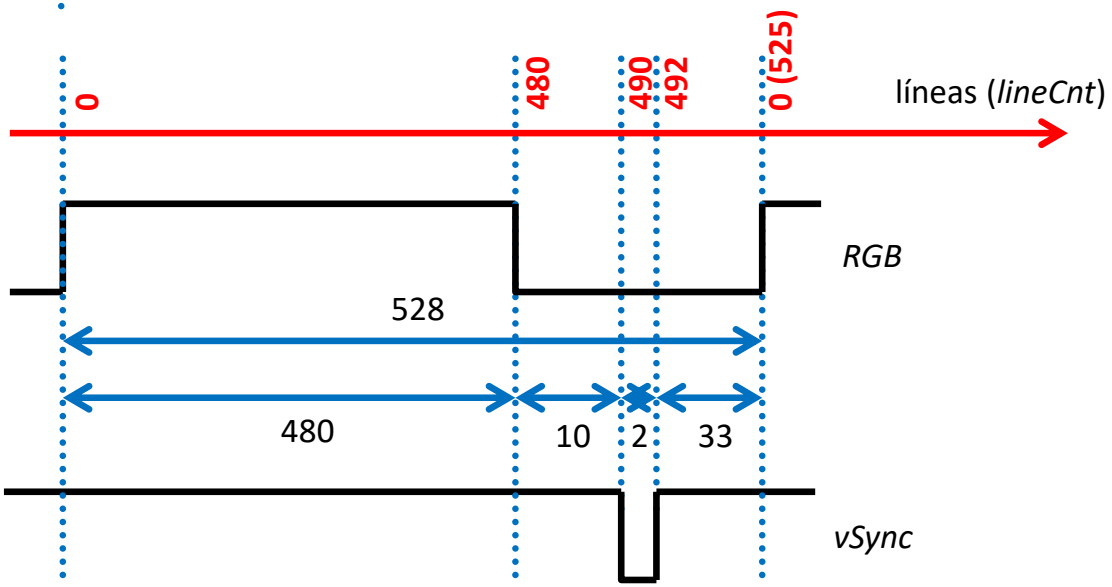
- Diseñar un interfaz VGA de 640×480 pixeles que constantemente:
  - Genere las señales de sincronismo hSync y vSync
  - Ponga a BAJA las señales de intensidad RGB durante los periodos de blanking
    - En cualquier otro momento RGB será equivalente a (R, G, B)
  - Lleve cuenta del pixel y línea que está barriendo en cada momento.
    - De modo que el color indicado por (R, G, B) se visualice en el pixel en la posición indicada.
    - Pixel y line deberán seguir la secuencia de barrido de un monitor VGA.



Interfaz VGA

esquema RTL (i)

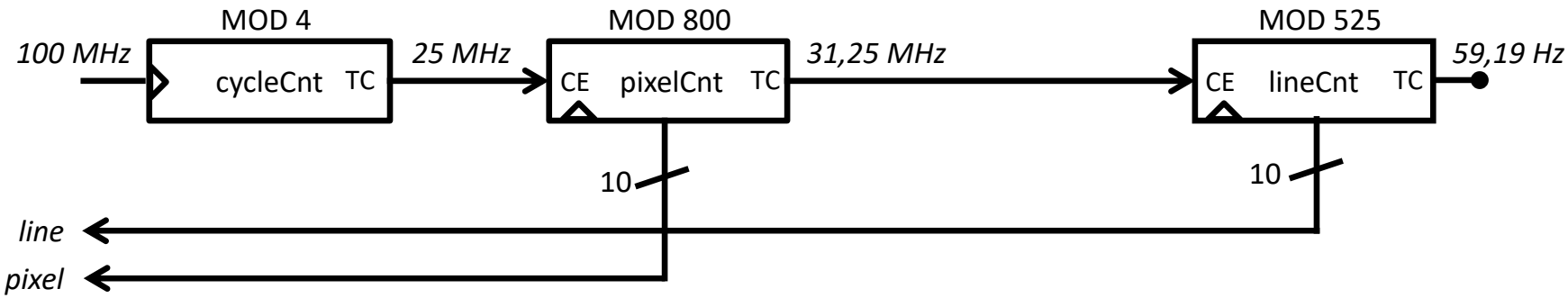




- RGB debe valer 0 cuando:
  - pixelCnt ≥ 640 o lineCnt ≥ 480
- hSync debe valer 0 cuando:
  - pixelCnt ≥ 656 y pixelCnt < 752
- vSync debe valer 0 cuando:
  - lineCnt ≥ 490 y lineCnt < 492

Interfaz VGA

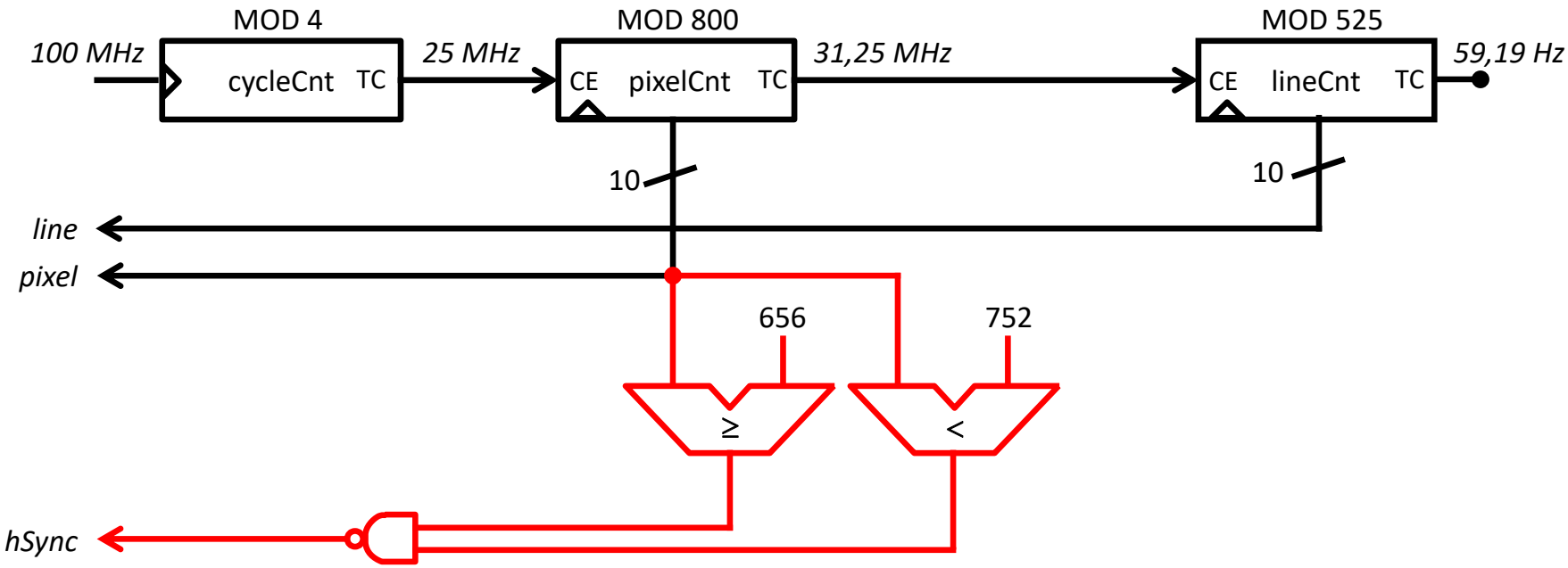
esquema RTL (ii)



Las frecuencias no son exactamente las definidas por el estándar, pero los monitores suelen tener cierta tolerancia. En el peor de los casos podrán perderse algunos de los píxeles finales de cada línea (3-4).

Interfaz VGA

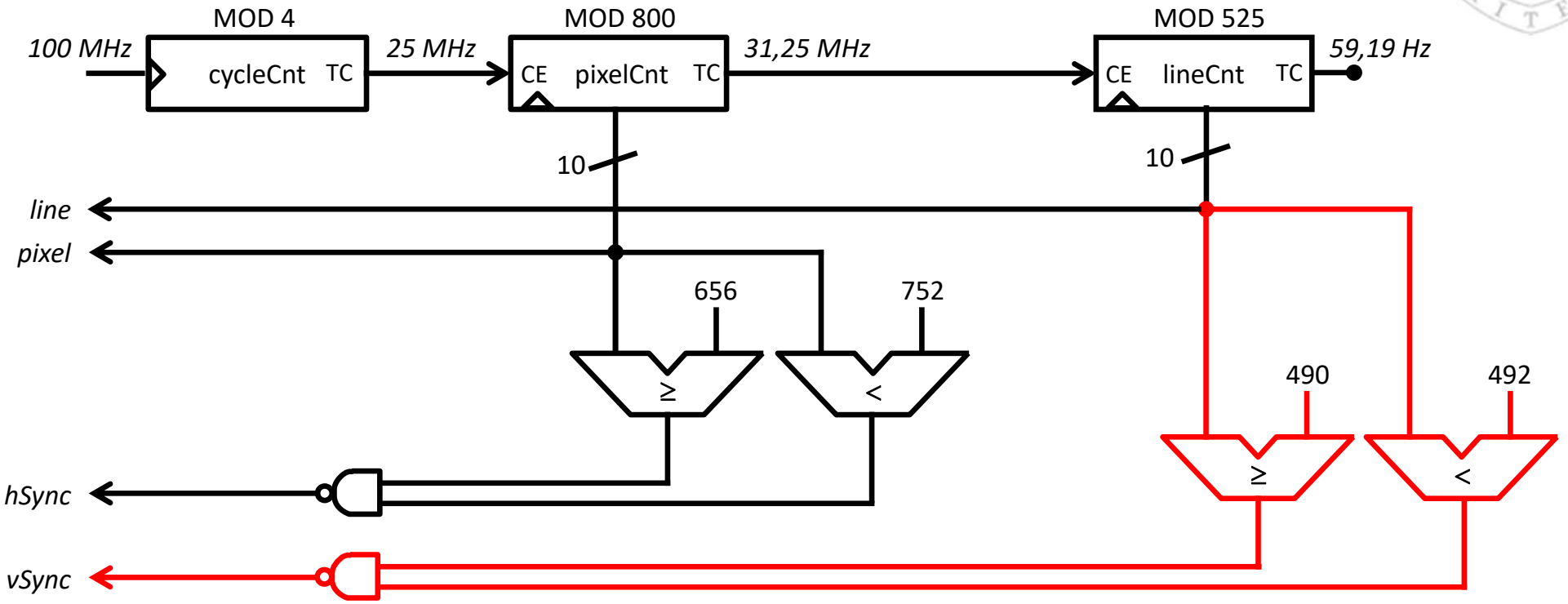
esquema RTL (ii)



Las frecuencias no son exactamente las definidas por el estándar, pero los monitores suelen tener cierta tolerancia. En el peor de los casos podrán perderse algunos de los píxeles finales de cada línea (3-4).

Interfaz VGA

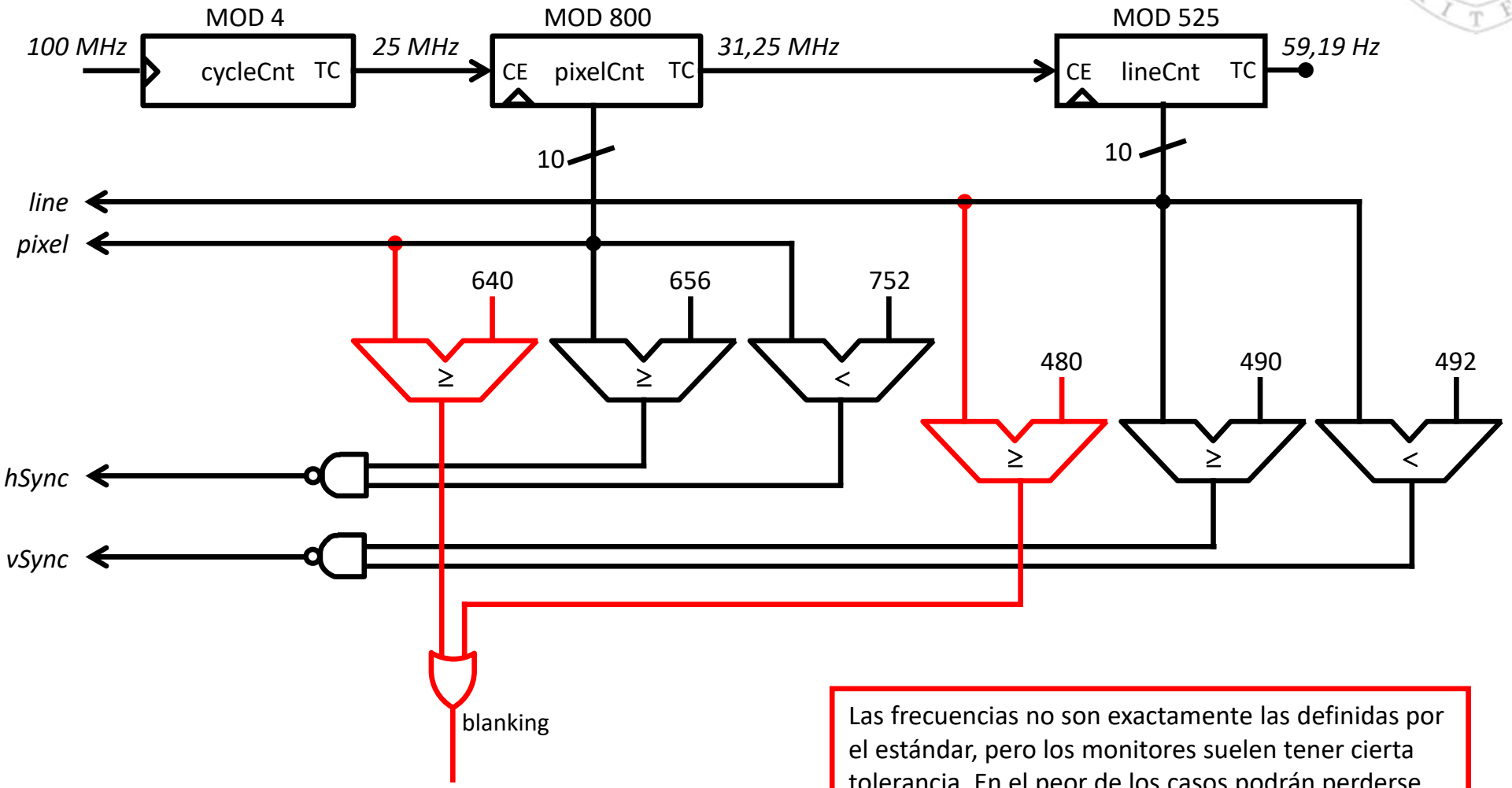
esquema RTL (ii)



Las frecuencias no son exactamente las definidas por el estándar, pero los monitores suelen tener cierta tolerancia. En el peor de los casos podrán perderse algunos de los píxeles finales de cada línea (3-4).

Interfaz VGA

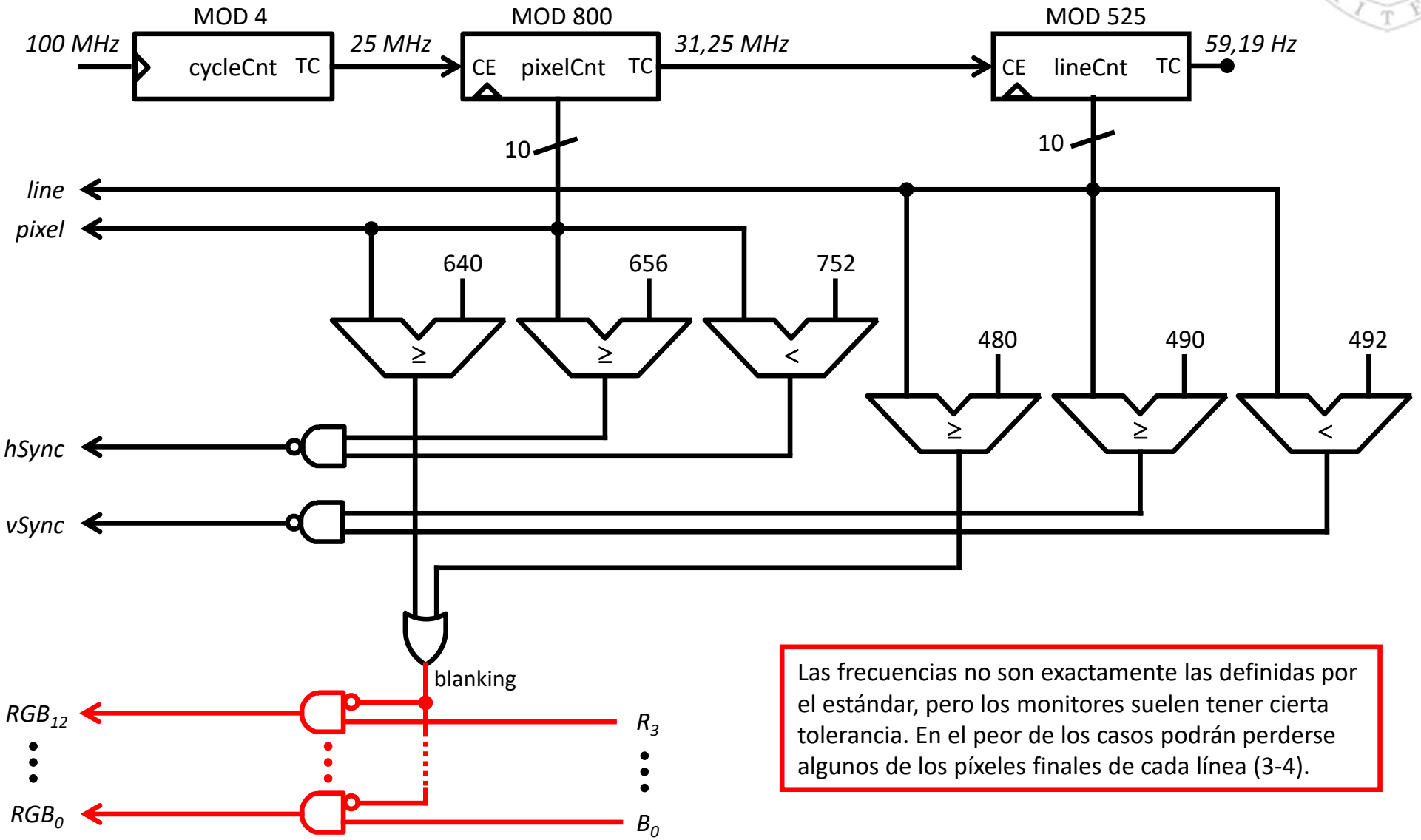
esquema RTL (ii)



Las frecuencias no son exactamente las definidas por el estándar, pero los monitores suelen tener cierta tolerancia. En el peor de los casos podrán perderse algunos de los píxeles finales de cada línea (3-4).

Interfaz VGA

esquema RTL (ii)



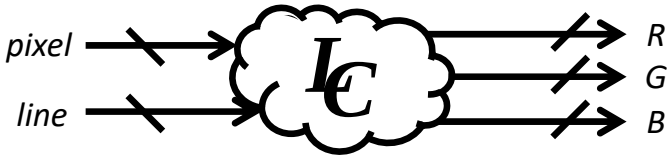
Las frecuencias no son exactamente las definidas por el estándar, pero los monitores suelen tener cierta tolerancia. En el peor de los casos podrán perderse algunos de los píxeles finales de cada línea (3-4).

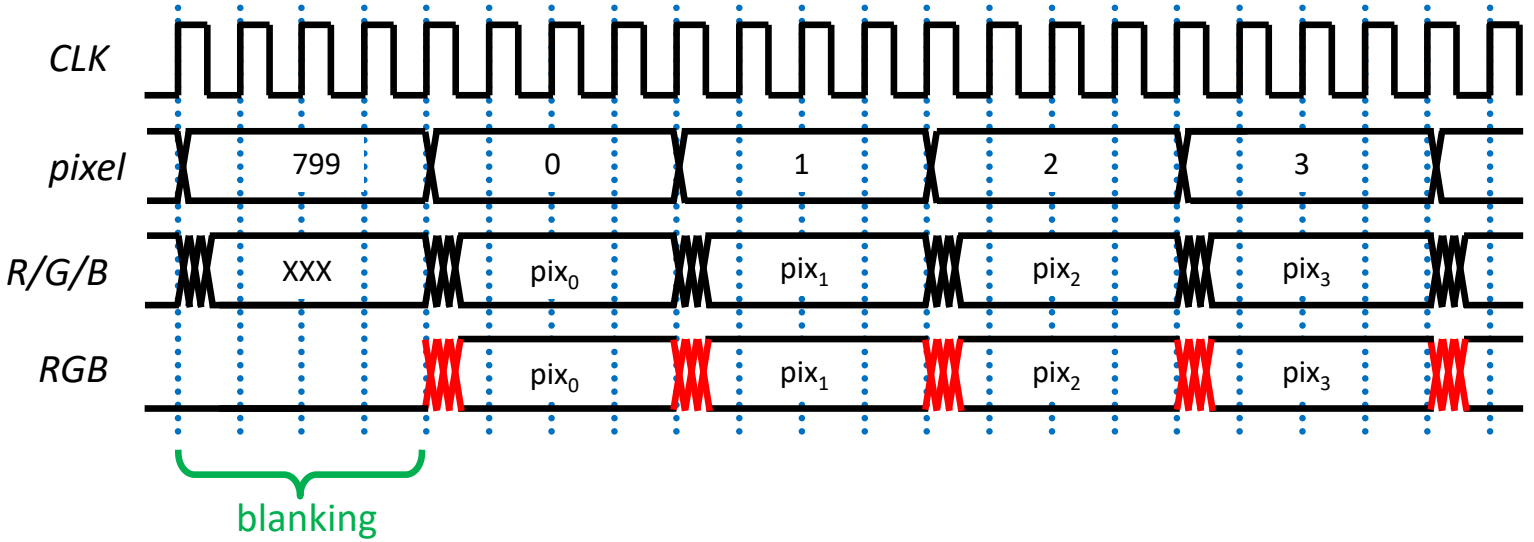
J.M. Mendías 2024

Interfaz VGA

análisis del comportamiento (i)

- Para que en cada posición se refresque el pixel que corresponde:
  - Las entradas R/G/B deben cambiar al mismo ritmo que las salidas pixel/line.
- Si R/G/B las calcula un circuito combinacional:
  - Los retardos combinacionales (distintos para cada bit) pueden provocar que en la pantalla se visualicen gitches:



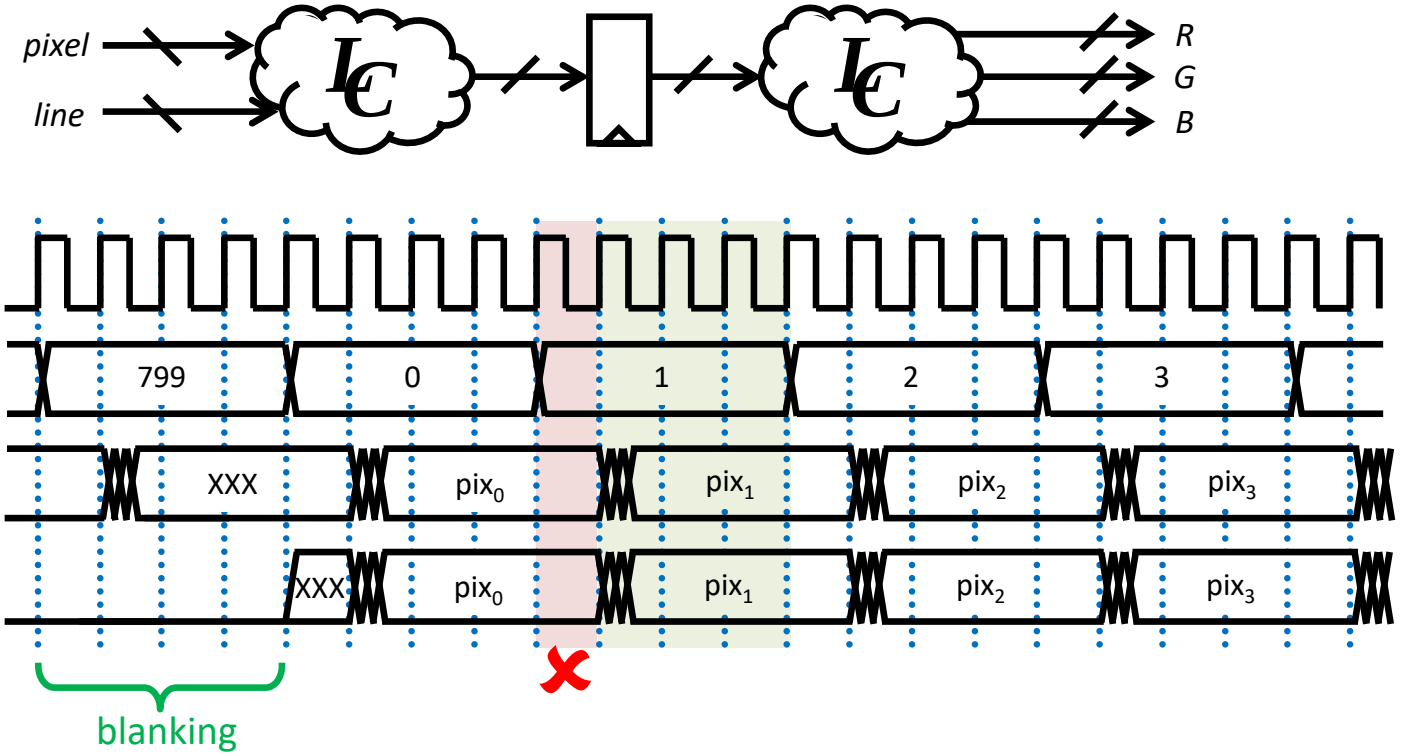


J.M. Mendias
2024

Interfaz VGA

análisis del comportamiento (ii)

- Si R/G/B las calcula un circuito secuencial de cierta latencia, además:
  - Cambiarán desfasadas un número fijo de ciclos respecto a los cambios de pixel/line.
  - Si el latencia es pequeña, el pixel correcto solo estará estable una fracción del tiempo.
  - Si la latencia es mayor, los píxeles se visualizarán desplazados hacia la derecha.
    - No obstante, no es habitual, porque la latencia debe ser superior al ratio entre la frecuencia del host y la frecuencia de envío de pixeles a pantalla (25 MHz).

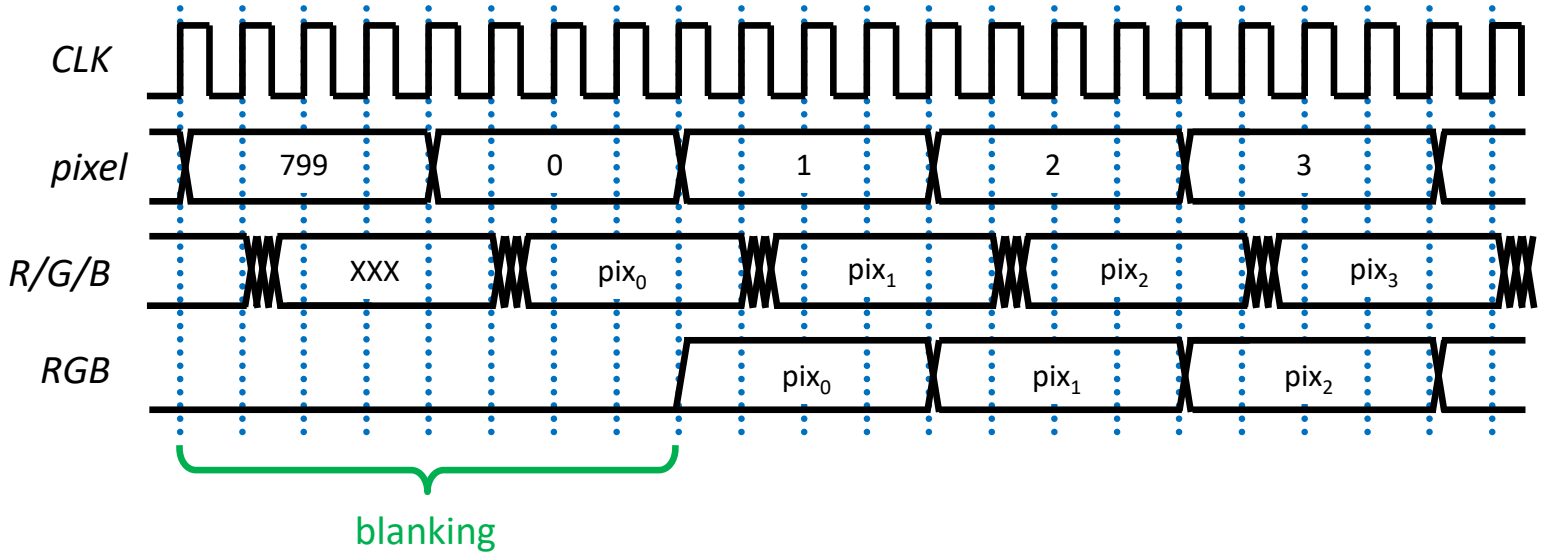


J.M. Mendías
2024

Interfaz VGA

análisis del comportamiento (iii)

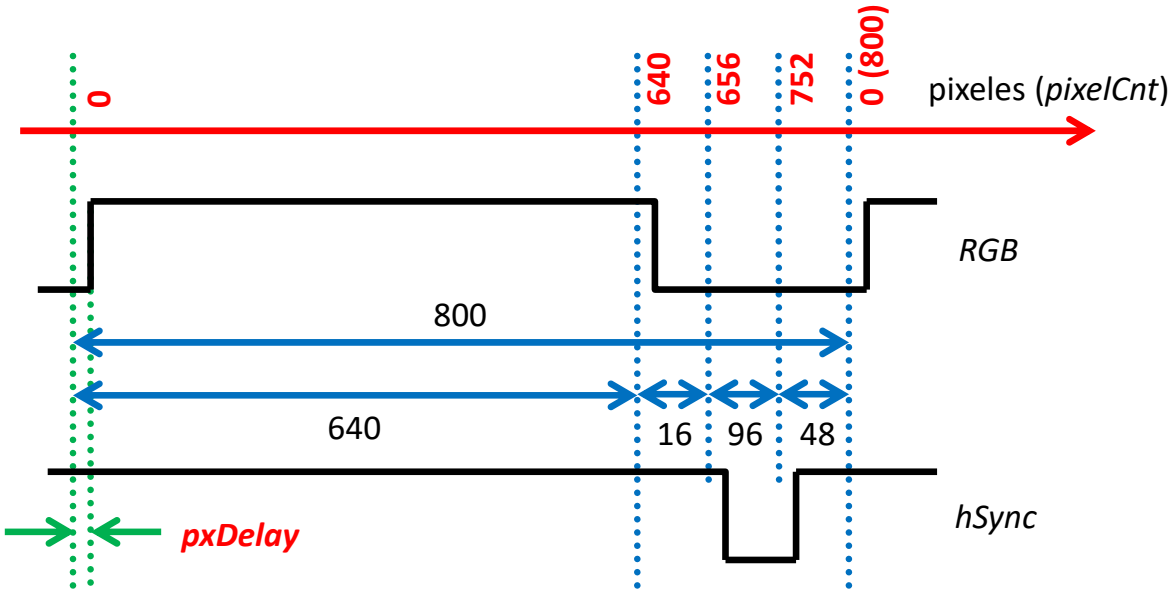
- Si la latencia es pequeña, el problema se resuelve registrando todas las salidas del interfaz.
  - Tanto las líneas RGB, como las de sincronización hSync/vSync
    - Recuérdese, que el monitor usa hSync/vSync (no pixel/line) para ubicar cada pixel.
  - Para tener un margen mayor, la carga debe realizarse en el último ciclo de cada pixel.



Interfaz VGA

esquema RTL (iii)

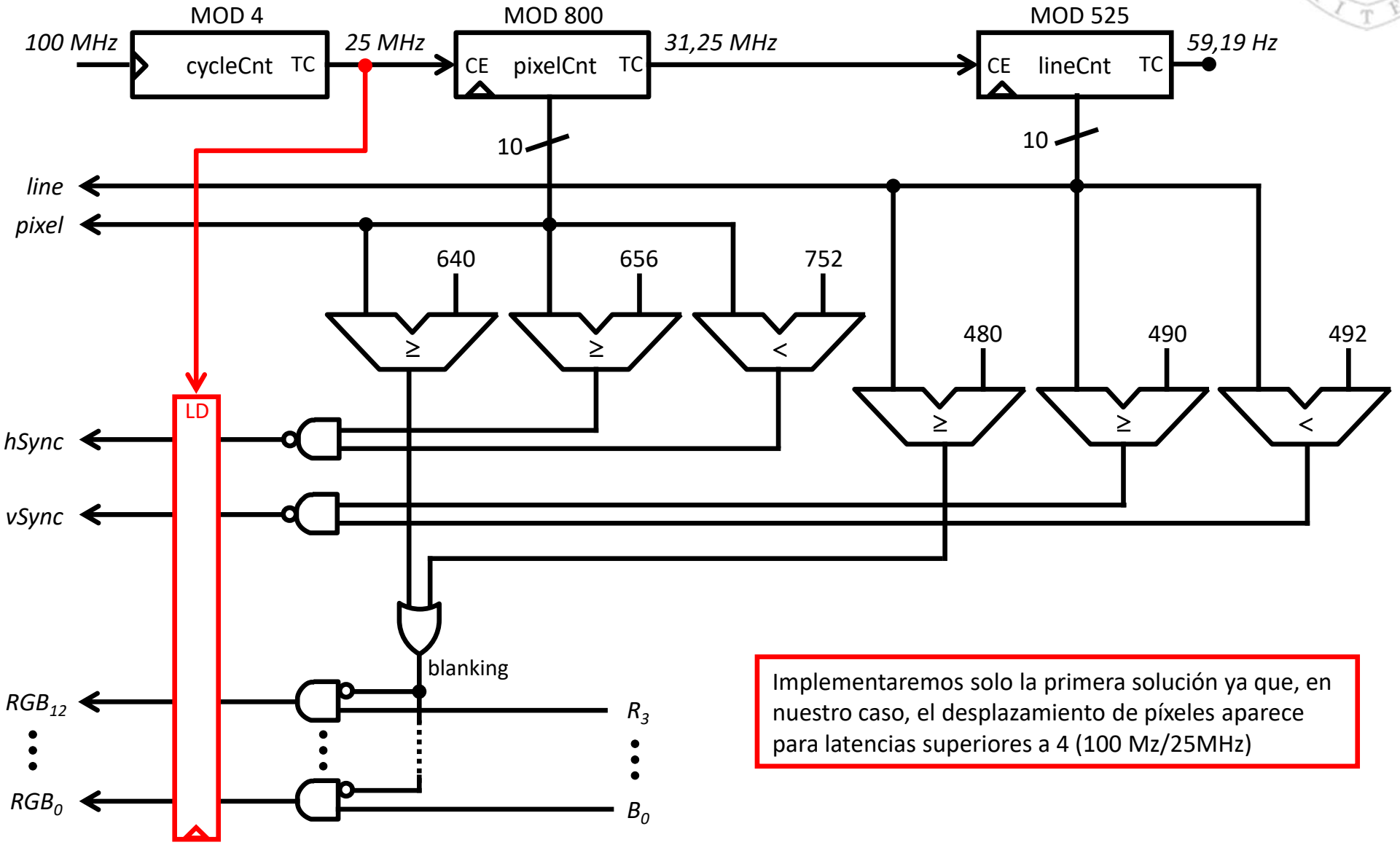
- Si la latencia es tal, que los pixel se visualizan desplazados, bastaría con retrasar hSync y blanking tantos píxeles como sea el desplazamiento.



- RGB debe valer 0 cuando:
  - pixelCnt < pxDelay o pixelCnt ≥ 640+pxDelay o lineCnt ≥ 480
- hSync debe valer 0 cuando:
  - pixelCnt ≥ 656+pxDelay y pixelCnt < 752+pxDelay
- vSync debe valer 0 cuando:
  - lineCnt ≥ 494 y lineCnt < 496

Interfaz VGA

esquema RTL (iv)



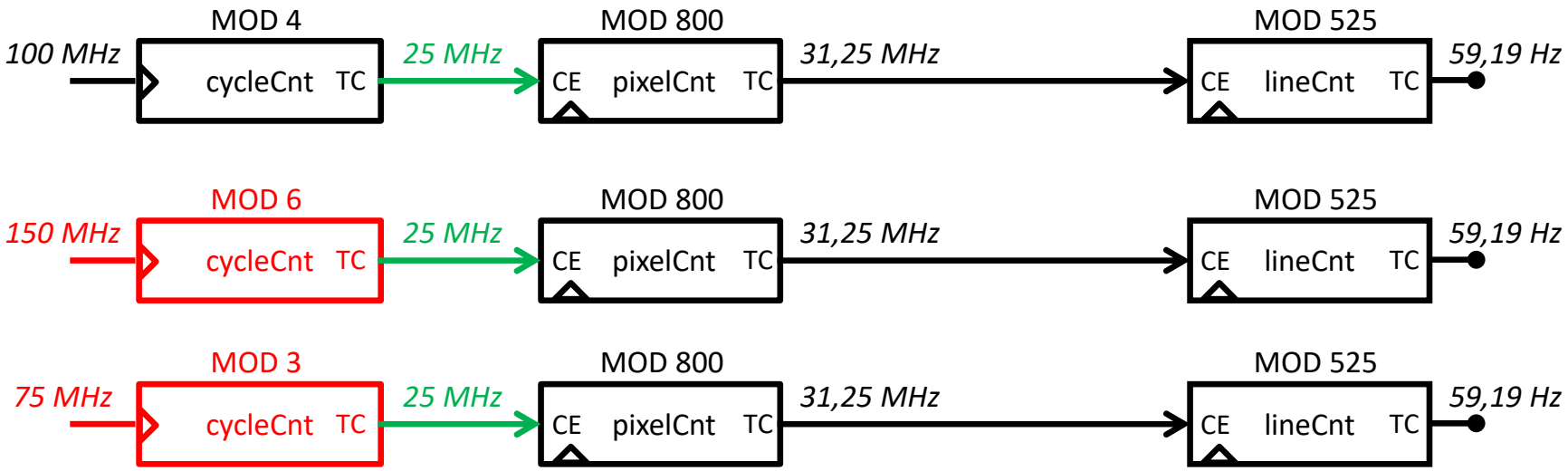
Implementaremos solo la primera solución ya que, en nuestro caso, el desplazamiento de píxeles aparece para latencias superiores a 4 (100 Mz/25MHz)

J.M. Mendias
2024

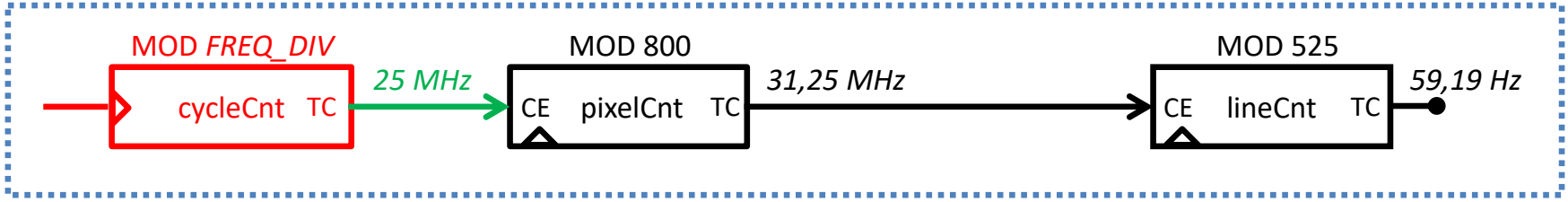
Interfaz VGA

generalización de la frecuencia de reloj

- Ajustando el límite de cycleCnt puede generalizarse el diseño para que pueda funcionar con distintas frecuencias de reloj.



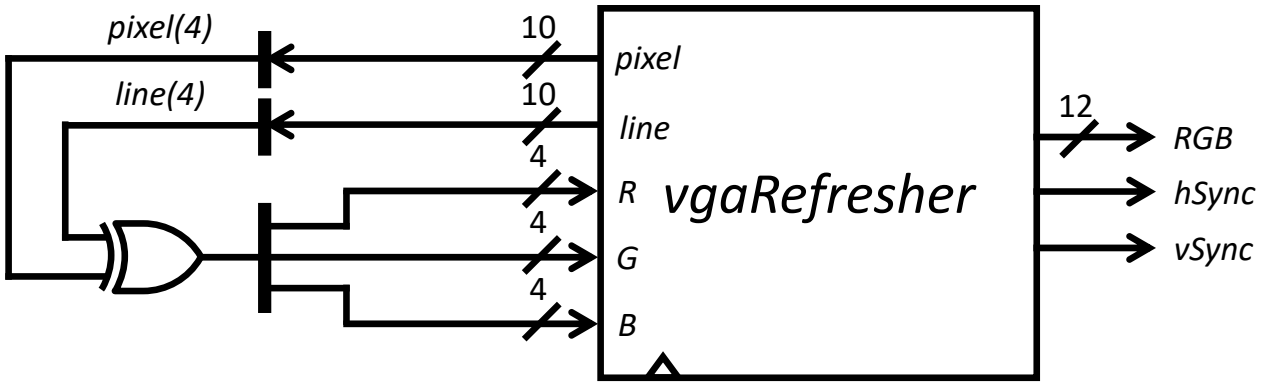
- Solo debe asegurarse que pixelCnt cuente a 25 Mhz.



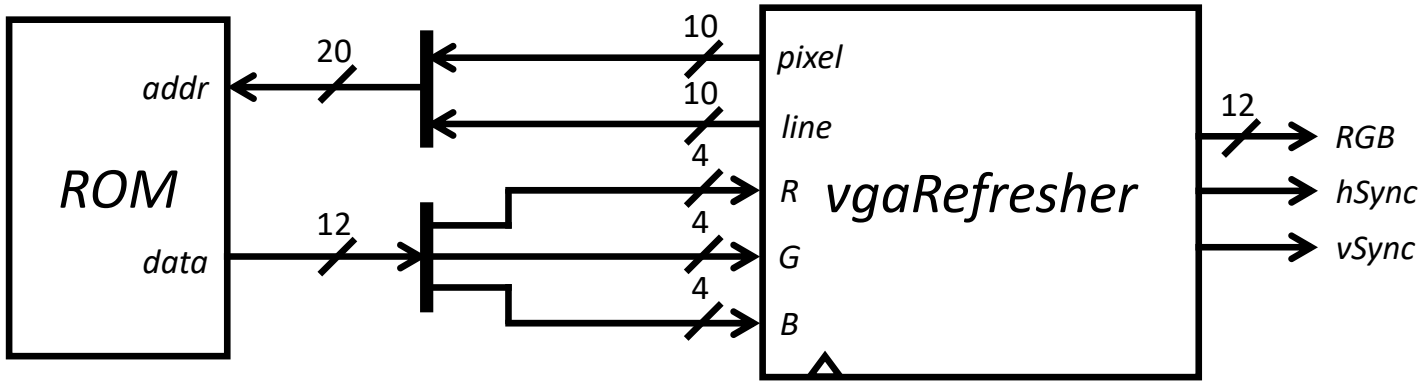
Interfaz VGA

aplicación al diseño

- Para visualizar un damero de 16×16 píxeles bastaría:



- Para visualizar una imagen fija almacenada bastaría:



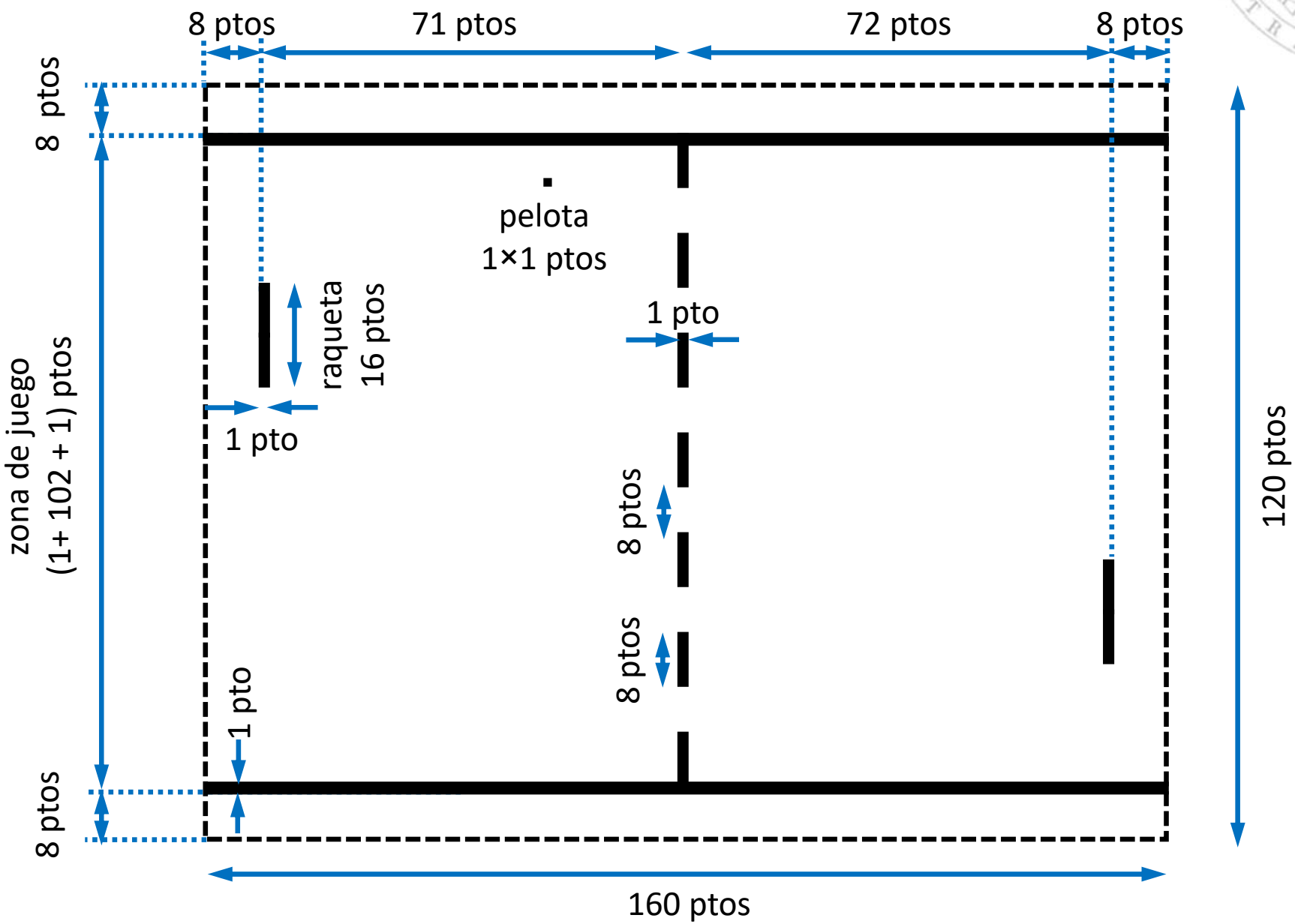
PONG

dinámica del juego

- La dinámica del juego será la siguiente:
  - Los elementos del juego se visualizará en blanco sobre y fondo negro.
  - Para que los elementos del juego sean de mayor tamaño (sin aumentar el coste del diseño) se realizará un redimensionamiento hardware:
    - Todas las dimensiones y coordenadas se medirán en puntos lógicos.
    - Cada punto lógico es un cuadrado de 4×4 píxeles reales.
    - Para ello bastará con ignorar los bits 2 menos significativos de pixel y line.
  - La velocidad de movimiento de raquetas y pelota será de 50 ptos/s
  - La pelota siempre está en movimiento y avanza en diagonal en 4 únicos sentidos que cambian ordenadamente al rebotar:
    - Sentido izquierda-arriba: se decrementa 1 pto sus posiciones x e y
    - Sentido izquierda-abajo: se decrementa 1 pto su posición x e incrementa 1 pto su posición y
    - Sentido derecha-abajo: se incrementa 1 pto sus posiciones x e y
    - Sentido derecha-arriba: se incrementa 1 pto su posición x e decrementa 1 px su posición y
  - Cada vez que la pelota salga por el fondo se esperará la pulsación de SPC para iniciar un nuevo juego:
    - En el nuevo juego la pelota saldrá desde el centro de la red con el mismo sentido y posición y que tenía en el juego anterior.

PONG

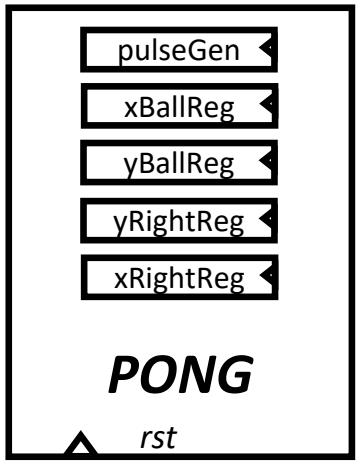
mapa del juego



PONG

esquema RTL (i)

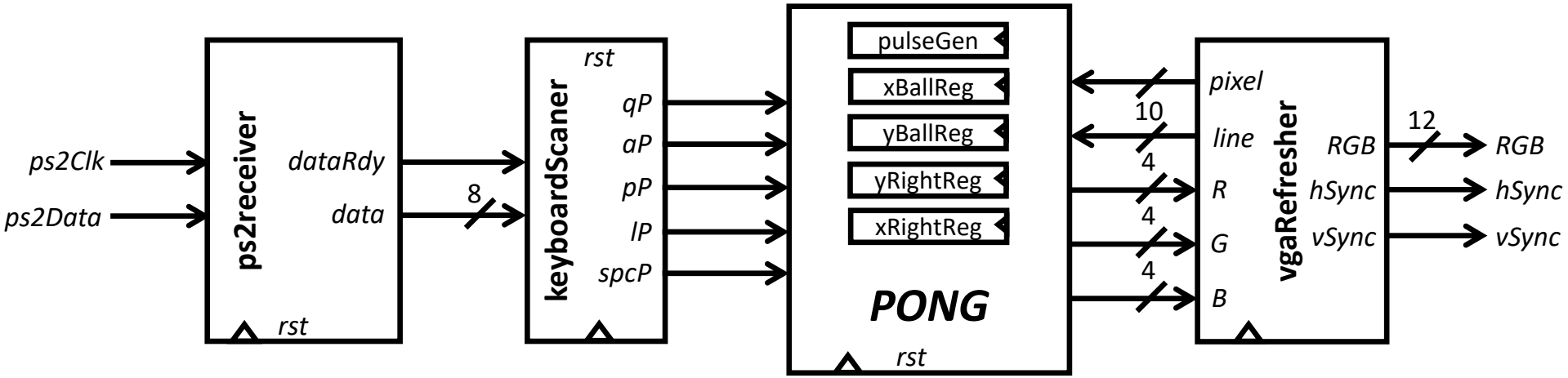
- Para gestionar la dinámica del juego, el diseño constará de:
  - Un contador que genere 50 pulsos/segundo mientras haya una partida en juego.
  - Dos registros que almacenen la posición y sentido de movimiento de la pelota:
    - Se inc/decrementarán a cada pulso según el sentido de movimiento que lleven.
    - Cambiarán de sentido cuando detecten un choque con un borde del campo o una raqueta.
  - Dos registros que almacenen la posición del lateral superior de cada raqueta:
    - Se inc/decrementará a cada pulso según las teclas que estén pulsadas.
  - Un bloque combinacional que detecte:
    - El final de la partida, de modo que detenga el contador de pulsos.
    - El inicio de una nueva partida, de modo que reinicie el contador de pulsos.



PONG

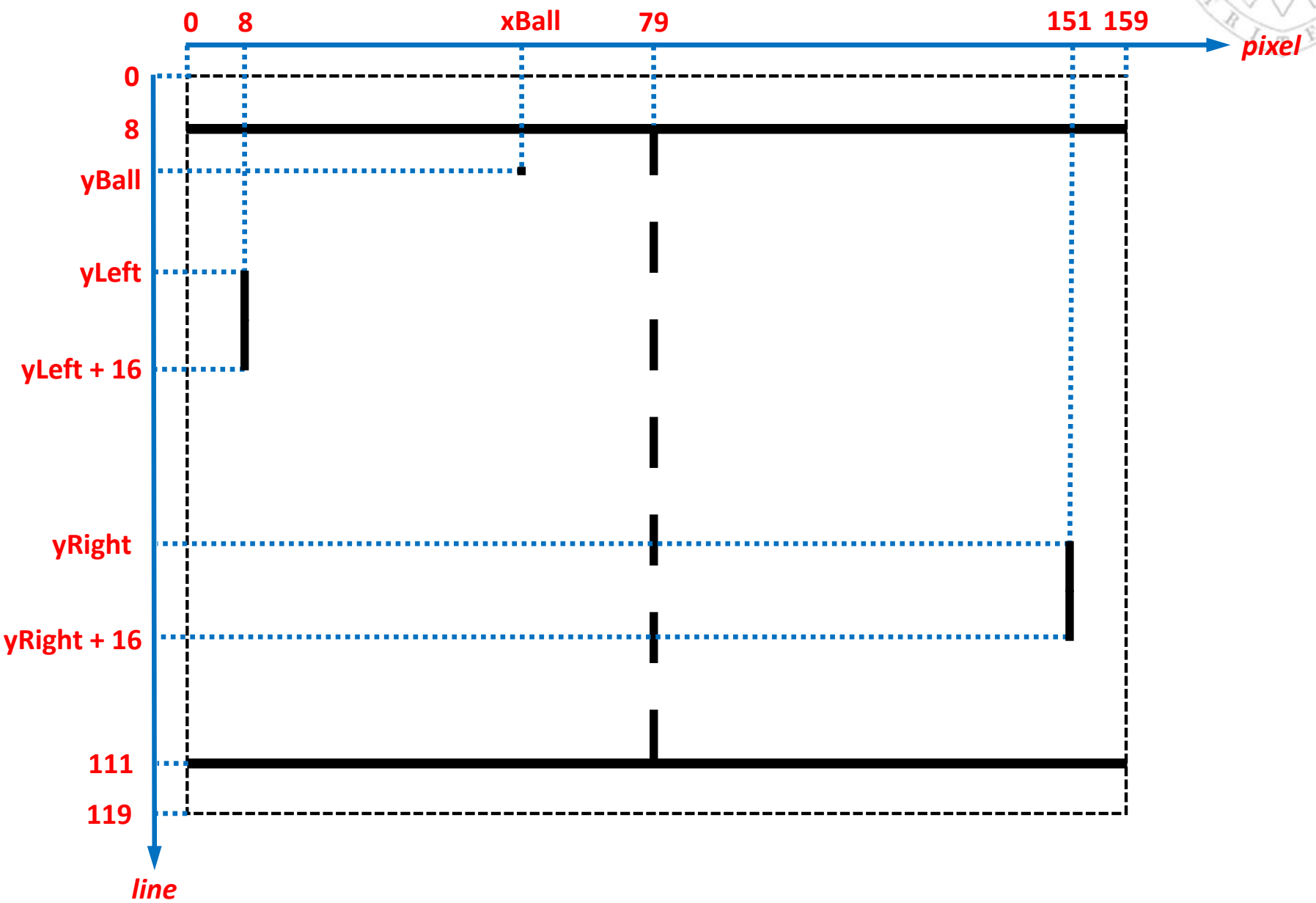
esquema RTL (i)

- Para la interacción con el jugador, dispondrá de:
  - Un ps2receiver conectado a un escáner de teclado (FSM con flags) que indique el estado de cada una de las teclas representativas del juego.
    - Dado que varias de ellas pueden estar siendo pulsadas a la vez.
  - Un VGArefresher conectado a un bloque de lógica combinacional que dibuje:
    - El campo, las raquetas y la pelota en función de sus posiciones.
- Para la visualización no usará memoria de refresco.
  - Dibujará "al vuelo" cada uno de los elementos gráficos según el valor de line y pyxel.



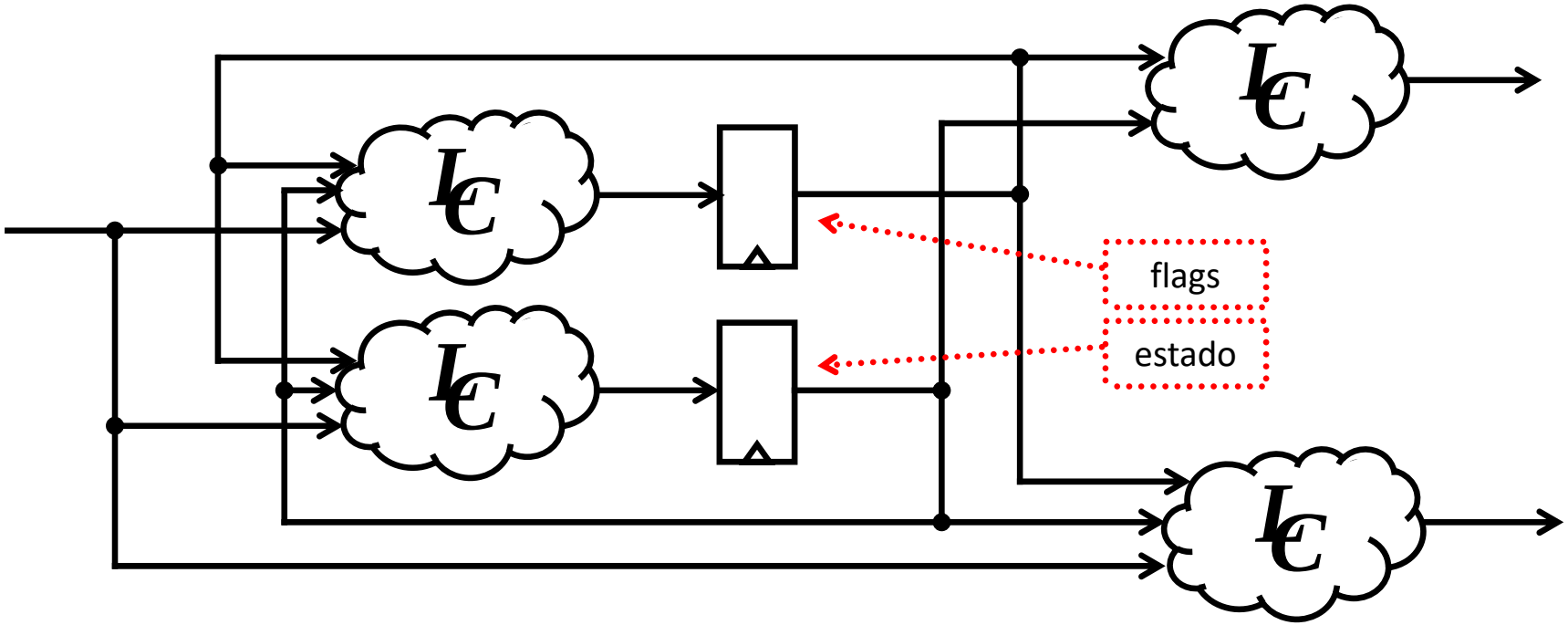
PONG

esquema RTL (iii)



FSM con flags

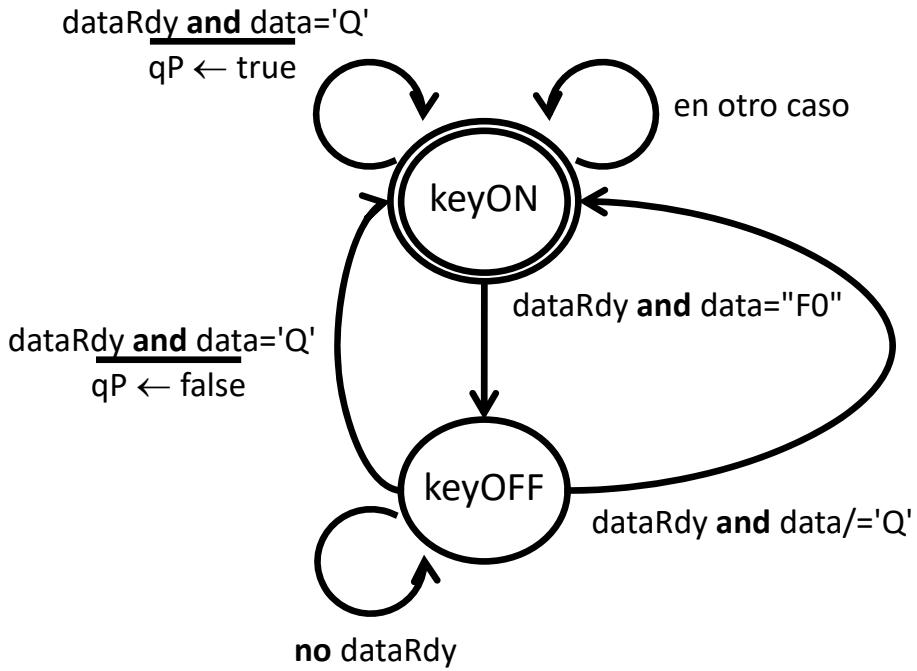
- Un caso particular de las FSMD son las FSM con flags
  - El estado "principal" del sistema se almacena en el registro de estado.
  - Existe un registro de flags (activados o desactivados según el estado principal) que "matiza" el comportamiento del sistema.
  - El uso de flags permiten reducir el número de estados de la FSM
    - Guardan información que de otra manera sería necesaria reflejar explícitamente mediante diferentes ramas del diagrama principal de estados.



FSM con flags

escáner de teclado

```
architecture syn of lab6pong is
  signal qP, ... : boolean := False;   -- Flags
  ...
begin
  keyboardScanner:
  process( clk )
    type states is (keyON, keyOFF);
    variable state : states;
  begin
    if rising_edge(clk) then
      if rstSync='1' then
        ...
      elsif dataRdy='1' then
        case state is
          when keyON =>
            case data is
              when X"F0" => state := keyOFF;
              when X"15" => qP <= true;
              ...
            end case;
          when keyOFF =>
            state := keyON;
            case data is
              when X"15" => qP <= false;
              ...
            end case;
        end case;
      end if;
    end if;
  end process;
```



Ejemplo de FSM con un único flag para almacenar el estado de la tecla Q (pulsada o no pulsada)

Tareas

damero

1. En la carpeta **DAS/source** crear las carpeta **lab6damero**.
2. Descargar de la Web los ficheros en:
   - **common:** **vgaRefresher.vhd**
   - **lab6damero:** **lab6damero.vhd** y **lab6damero.xdc**
3. Completar **common.vhd** con la declaración del nuevo componente.
4. Completar el código omitido en el fichero:
   - **vgaRefresher.vhd**
5. En la carpeta **DAS/projects** crear el proyecto **lab6damero**.
6. Incluir en el proyecto (sin copiarlos) los siguientes fuentes:
   - **common.vhd**, **vgaRefresher.vhd**, **lab6damero.vhd** y **lab6damero.xdc**
7. Sintetizar, implementar y generar el fichero de configuración.
8. Conectar el monitor a la placa y encenderla.
9. Volcar el fichero de configuración **lab6damero.bit**

Tareas

pong

1. En la carpeta **`DAS/source`** crear las carpeta **`lab6pong`**.
2. Descargar de la Web los ficheros en:
   - **`lab6pong:`** **`lab6pong.vhd`** y **`lab6pong.xdc`**
3. Completar **`common.vhd`** con la declaración del nuevo componente.
4. Completar el código omitido en el fichero **`lab6pong.vhd`**
5. En la carpeta **`DAS/projects`** crear el proyecto **`lab6pong`**.
6. Incluir en el proyecto (sin copiarlos) los siguientes fuentes:
   - **`common.vhd`**, **`synchronizer.vhd`**, **`edgeDetector.vhd`**, **`ps2receiver.vhd`**, **`vgaRefresher.vhd`**, **`lab6pong.vhd`** y **`lab6pong.xdc`**
7. Sintetizar, implementar y generar el fichero de configuración.
8. Conectar el teclado y el monitor a la placa y encenderla.
9. Volcar el fichero de configuración **`lab6pong.bit`**

Acerca de *Creative Commons*

- Licencia CC (Creative Commons)

  - Ofrece algunos derechos a terceras personas bajo ciertas condiciones. Este documento tiene establecidas las siguientes:

Reconocimiento (*Attribution*):
En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.

No comercial (*Non commercial*):
La explotación de la obra queda limitada a usos no comerciales.

Compartir igual (*Share alike*):
La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Más información: https://creativecommons.org/licenses/by-nc-sa/4.0/