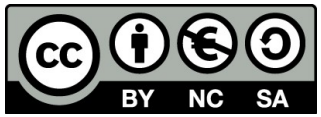
Laboratorio 9:

FSM temporizadas

interfaces gráficos de vídeo

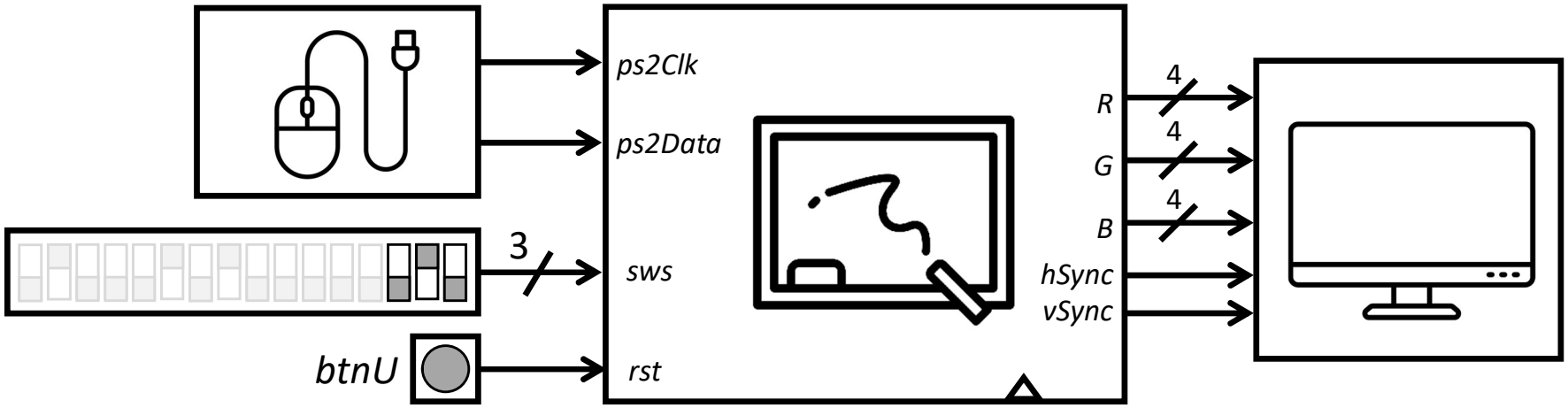
Diseño automático de sistemas

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



Presentación

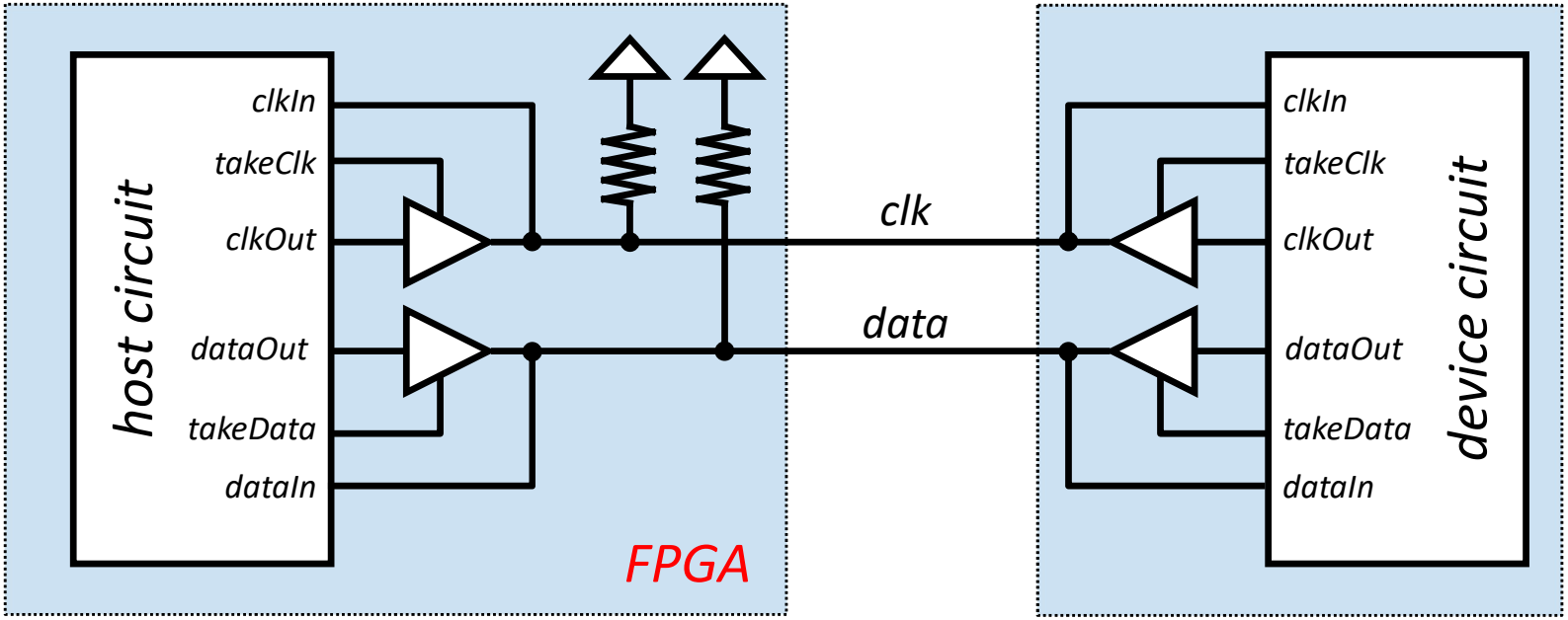
- Diseñar un terminal gráfico que dibuje en una pantalla VGA según lo indicado por un ratón con interfaz PS/2.
  - Visualizará "al vuelo" un cursor en la pantalla que se moverá al mover el ratón.
    - Inicialmente situado la esquina superior izquierda.
  - Si el botón izquierdo está pulsado, pintará los pixeles por donde pase el cursor.
    - En el color codificado por los 3 switches menos significativos.
  - Cuando se pulse el botón derecho, se borrará la pantalla.
  - Dispondrá de reset síncrono, que llevará al ratón a su posición inicial.



Protocolo PS/2

interfaz eléctrico

- El protocolo PS/2 es bidireccional:
  - Las líneas pueden usarse tanto para transmitir como para recibir información.
- Por ello las líneas CLK y DATA usan un interfaz *open collector:*
  - Por defecto, las líneas se mantienen en ALTA gracias a resistencias de *pull-up*.
  - Para transmitir 0 deben forzarse a BAJA, para transmitir 1 o recibir basta con liberarlas.

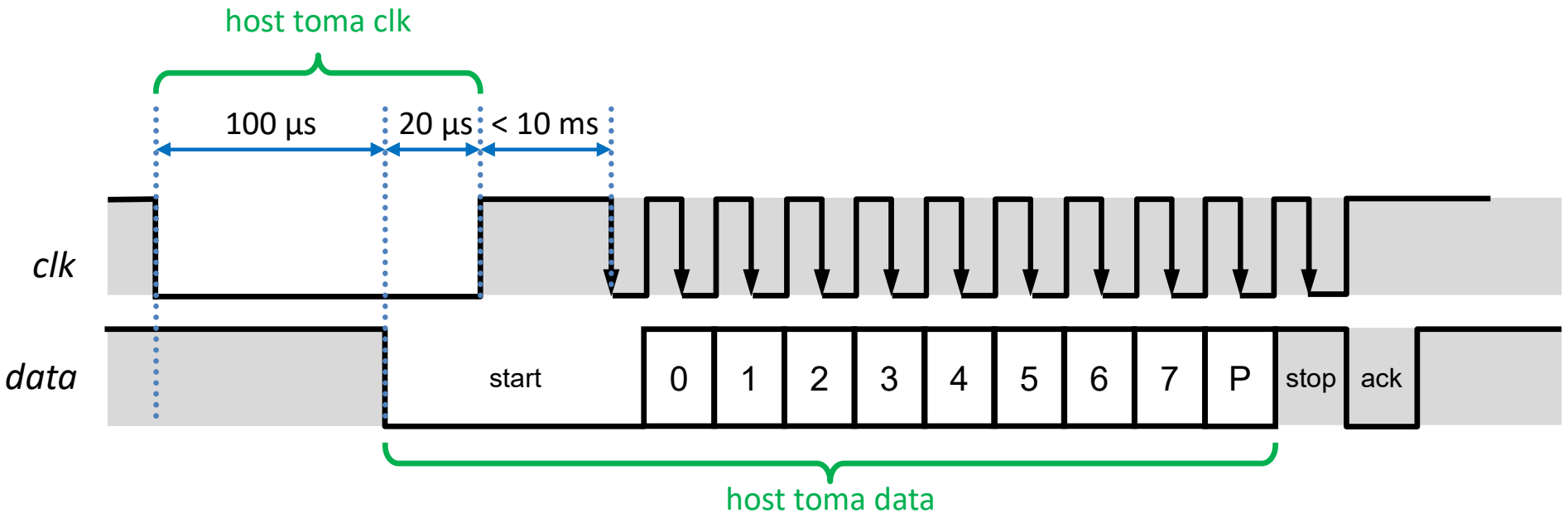



```
set_property PULLUP true [get_ports ps2Clk]
```

Protocolo PS/2

transmisión host-to-device

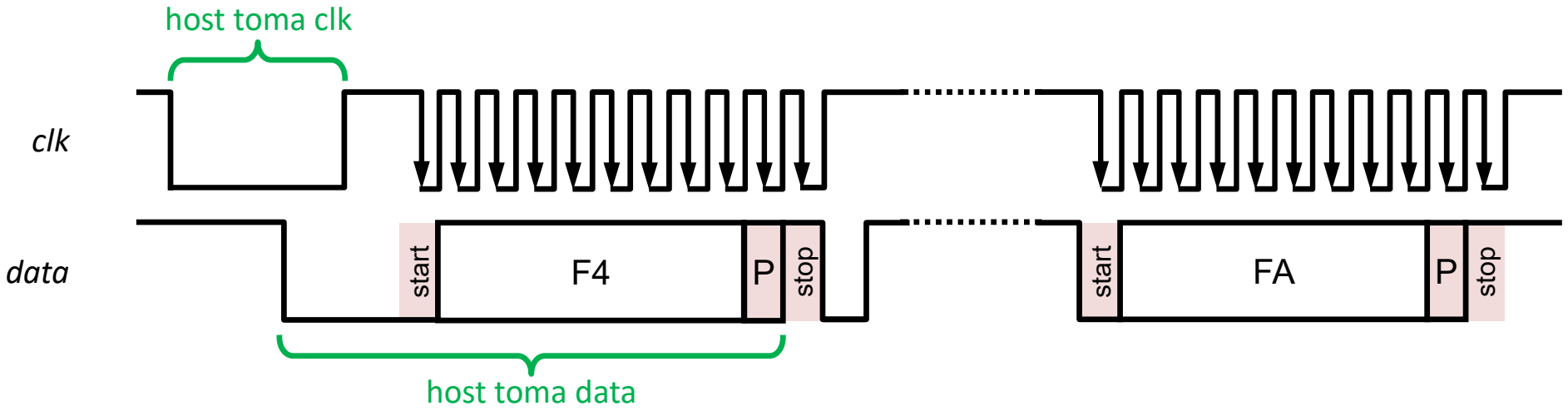
- El host puede en cualquier momento:
  - Inhabilitar las transmisiones forzando CLK a baja durante al menos 60 µs.
    - En cuyo caso los datos se almacenan temporalmente en el dispositivo.
  - Indicar al dispositivo su intención de enviarle información forzando DATA a baja.
    - Una vez el host libera CLK, el dispositivo generará la señal de reloj antes de 10 ms.
    - El dispositivo muestrea DATA a flancos de bajada, por lo que el host debe poner nuevos bits en DATA a flanco de subida de CLK (con un retraso máximo de 1 µs).
    - El host, tras enviar el bit de paridad, debe liberar DATA
    - El dispositivo reconoce la recepción poniendo momentáneamente DADA a baja.



Protocolo PS/2

ratón: scancodes (i)

- Tras su enchufado, el ratón envía 2 códigos y permanece inactivo:
  - Power-on satisfactorio (AA) seguido de Mouse ID (00).
- Por ello, el host debe habilitar el ratón:
  - Enviando Enable reporting (F4) y esperando la recepción de ACK (FA).



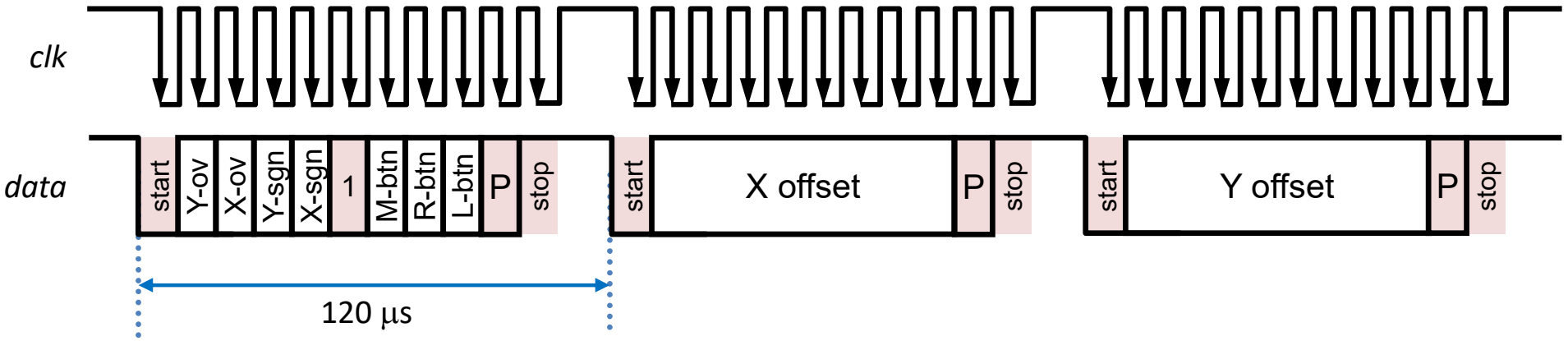
- Opcionalmente, el host puede programar el modo de muestreo.
  - Por defecto, la resolución es 4 ud/mm y la frecuencia de muestreo 100 SPS.
  - Sample rate (F3): responde con ACK (FA), espera un byte y responde con ACK (FA).
  - Resolution (E8): responde con ACK (FA), espera un byte y responde con ACK (FA).

J.M. Mendías 2024

Protocolo PS/2

ratón: scancodes (ii)

- Tras habilitar el ratón, cada vez que se mueve, el envía 3 códigos:
  - Status: indica el estado de los pulsadores, el signo del movimiento y si hay overflow
  - X offset: indica el incremento (en C2) de posición X desde el último envío.
  - Y offset: indica el incremento (en C2) de posición Y desde el último envío.



- Obsérvese que los offsets realmente son de 9 bits:
  - Se obtienen concatenando el signo del byte de status con los 8 bits del byte de offset.

Interfaz PS/2

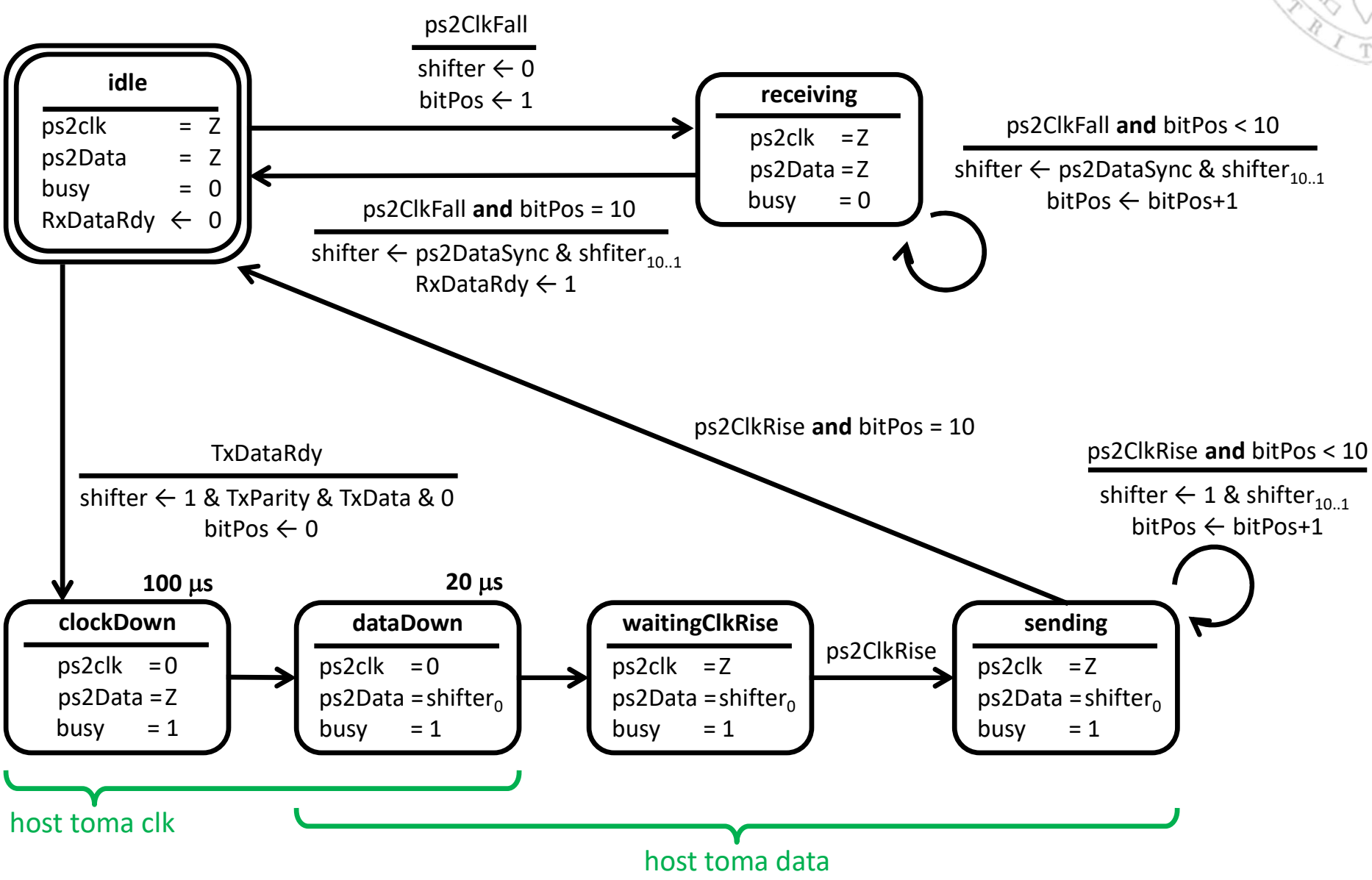
presentación

- Diseñar un transmisor/receptor PS/2 elemental:
  - Los puertos ps2Clk y Ps2Data serán ahora bidireccionales.
    - Deberá controlar cuando vuelcan datos o cuando se ponen en alta impedancia.
  - Para recepción se comportará como el p2receiver.
    - Pero ignorando el bit de paridad recibido.
  - Cada vez que se active TxDataRdy, transmitirá TxData en serie por el bus PS/2.
  - La señal de busy debe permanecer activa durante toda la transmisión.
  - Dispondrá de reset síncrono activo a alta.

J.M. Mendias
2024

Interfaz PS/2

FSMD temporizada



J.M. Mendías
2024

Interfaz PS/2

ps2interface.vhd

```
process (clk)
    ...
  begin
    RxData <= shifter(8 downto 1);
    case state is
      when idle | receiving =>
        ps2Clk  <= 'Z';
        ps2Data <= 'Z';
        busy    <= '0';
      when clockDown =>
        ...
    end case;
    if rising_edge(clk) then
      if rst='1' then
        ...
      elsif numCycles /= 0 then
        numCycles := numCycles - 1;
      else
        case state is
          when idle =>
            RxDataRdy <= '0';
            if txDataRdy='1' then
              state     := clkDown;
              numCycles := us2cycles(FREQ_KHZ, 100);
              shifter   := "1" & TxParity & TxData & "0";
              ...
            elsif ps2ClkFall='1' then
              state     := receiving;
              ...
            end if;
         when receiving =>
            ...
end process;
```

salidas tipo Moore

salidas en alta impedancia

temporizador

transferencias entre registros

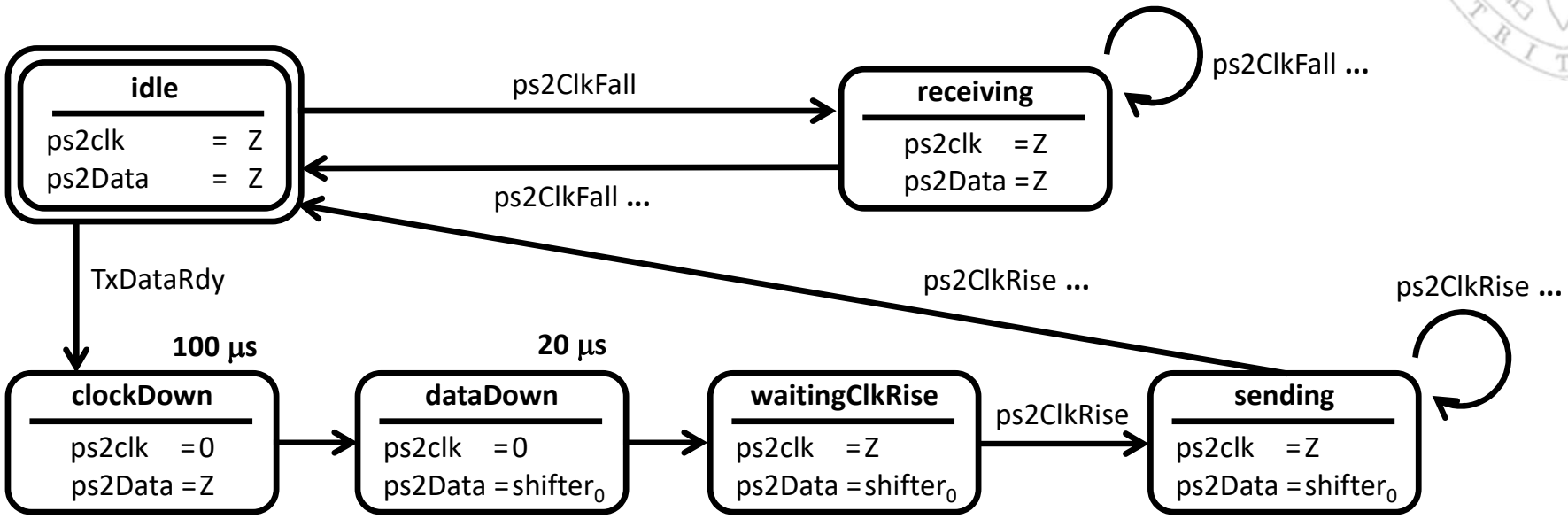
salto de estado temporizado

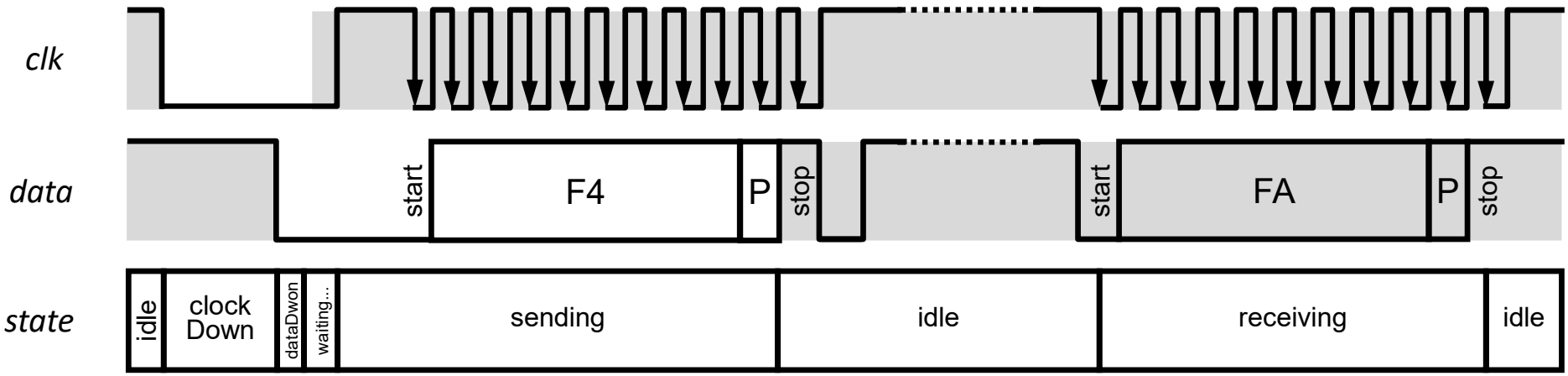
salto de estado no temporizado

J.M. Mendías 2024

Interfaz PS/2

análisis del comportamiento



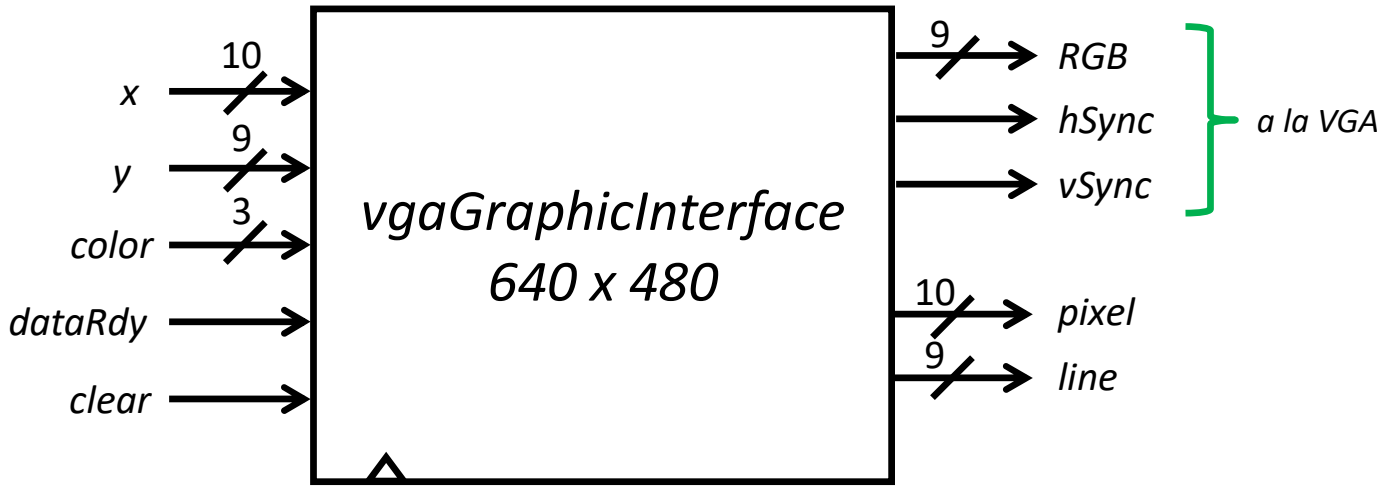


J.M. Mendías 2024

Interfaz gráfico de vídeo

presentación

- Diseñar un interfaz gráfico VGA:
  - Capaz de visualizar 640×480 pixeles de hasta 8 colores. Para ello dispondrá:
    - Una memoria de refresco (RAM) que almacena el color de cada pixel.
  - Estará controlado por 2 señales de tipo strobe que no se activarán simultáneamente:
    - dataRdy : almacenará en la posición (x,y) de la memoria de refresco color.
    - clear: borrará la pantalla, escribiendo 0 en todas las celdas de la memoria de refresco.
  - Por (line,pixel) indicará las coordenadas del pixel que está refrescando.
    - Por si se desea superponer gráficos "al vuelo".

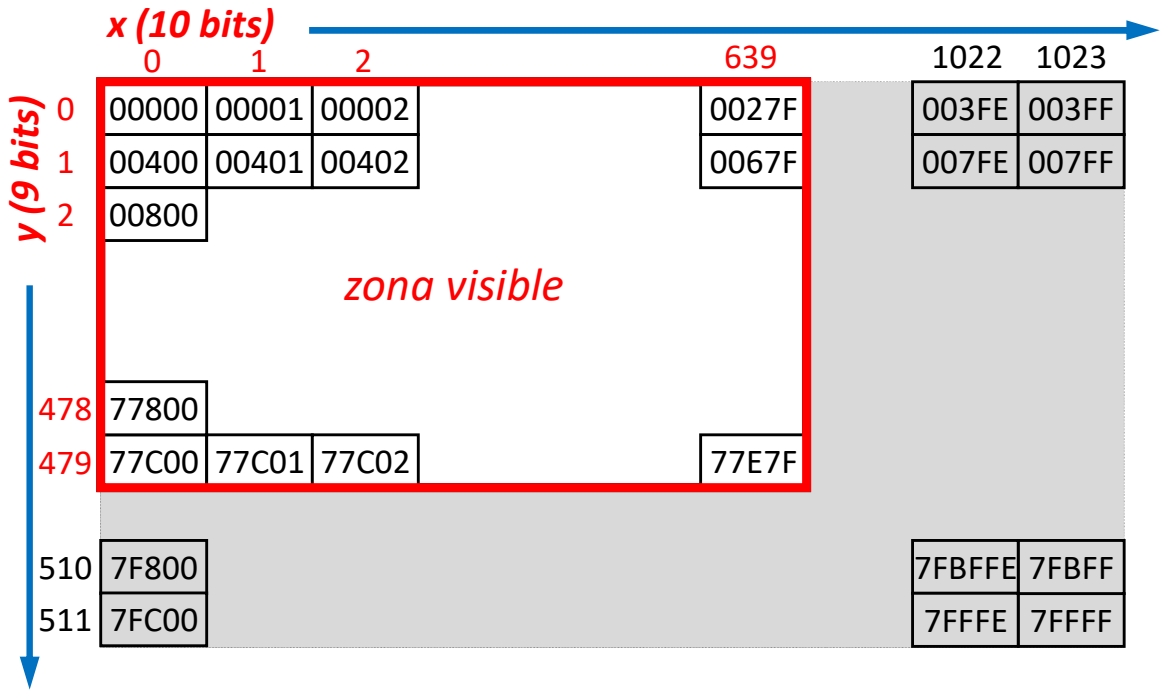


J.M. Mendías 2024

Interfaz gráfico de vídeo

memoria de refresco

- En una RAM se almacenan los 3 bits de color de cada pixel a visualizar:
  - La dirección de memoria se obtendrá concatenando las coordenadas de la posición.
    - Parte de la memoria queda desaprovechada, a costa de simplificar el acceso.
  - La memoria de refresco completa requiere de $3 \times 2^{10+9}$ b = 1536 Kb
    - Proyectable sobre 43 Block RAM en teoría, pero sobre 48 en la práctica.

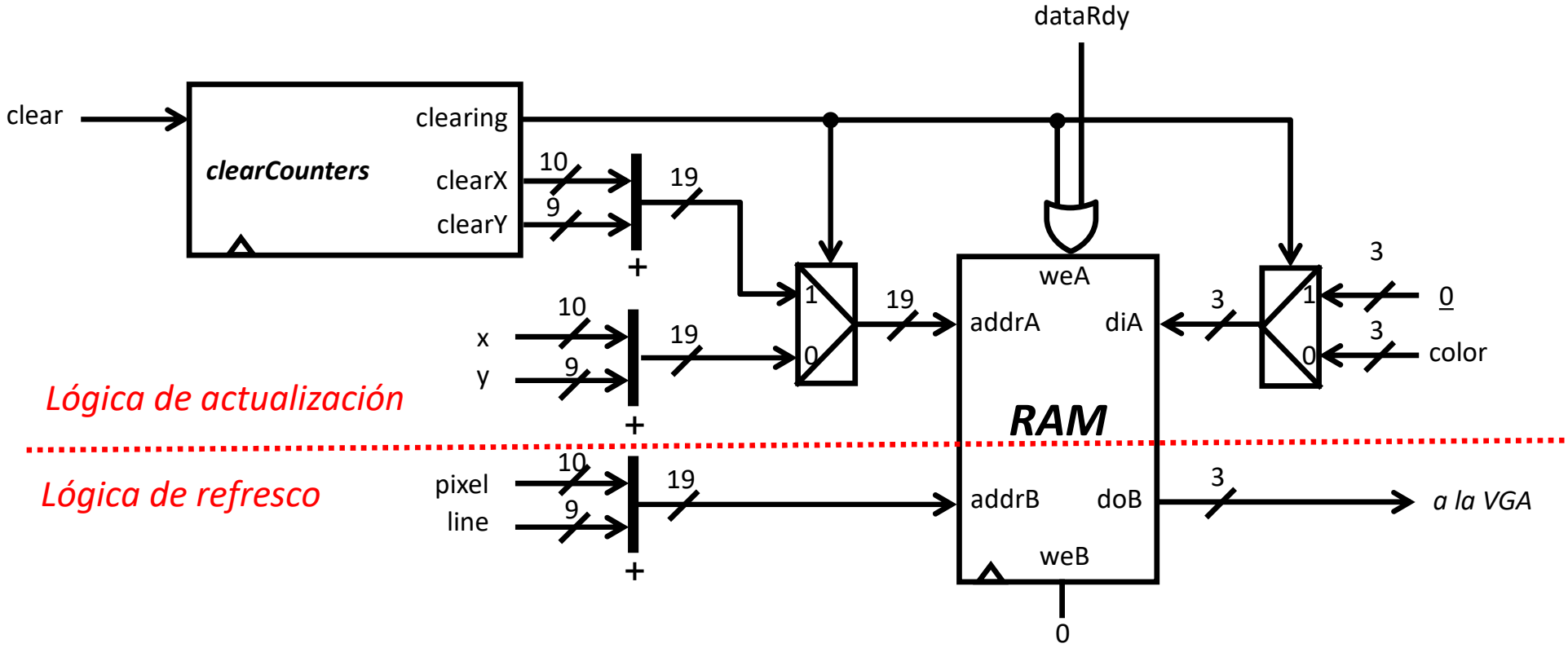


J.M. Mendías 2024

Interfaz gráfico de vídeo

procedimiento de actualización

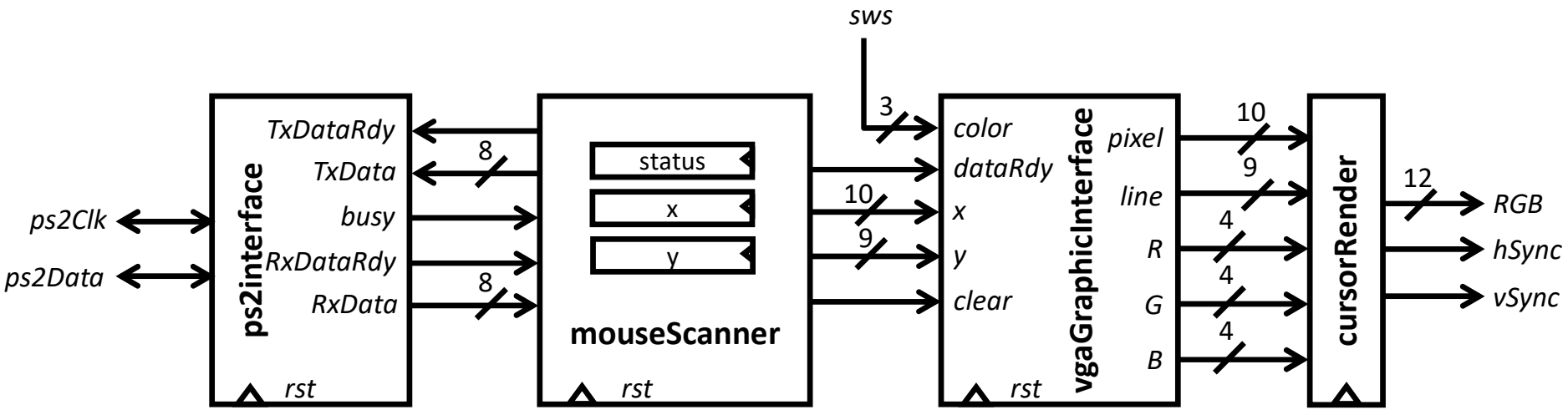
- La RAM será de tipo simple dual port, ya que al aislar la lógica de actualización de la de refresco se simplifica el diseño.
  - Un puerto dedicado a leer el pixel a refrescar y otro a escribir/borrar un pixel.
  - Al ser un interfaz de refresco, el modo de la Block RAM podrá ser cualquiera.
  - Para borrar la pantalla se barrerá la memoria escribiendo 0 en todas sus direcciones.



Diseño principal

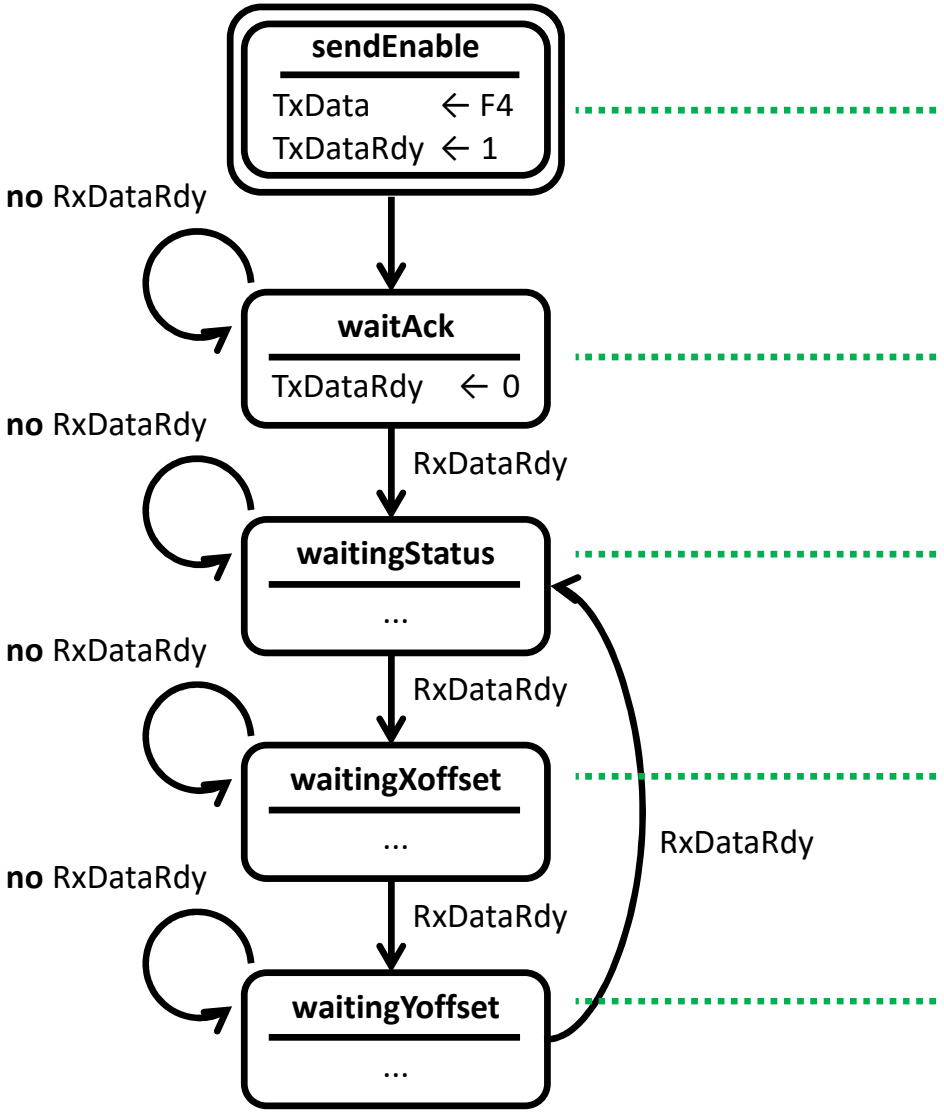
esquema RTL

- El diseño constará de:
  - Un ps2interface bidireccional conectado a un escáner de ratón (FSMD) que:
    - Habilitará el ratón e indefinidamente recibirá tramas de 3 bytes.
    - Gestionará la actualización de los registros (x,y) que almacenan la posición del cursor.
    - Gestionará el borrado de la pantalla y cuando debe pintarse alguno de sus puntos.
    - Inicialmente x e y valdrán 0 y deberán actualizarse sin salirse de los bordes.
  - Un vgaGraphicInterface.
  - Un módulo combinacional que superponga "al vuelo" sobre la imagen visualizada un cursor en la posición (x,y).



Diseño principal

esquema RTL: escáner de ratón



habilita el ratón enviado el comando "F4"

espera la recepción del ACK

carga en status el primer byte recibido

actualiza la posición X del cursor según el segundo byte recibido y el signo indicado en status, sin salirse de los bordes

actualiza la posición Y del cursor según el tercer byte recibido y el signo indicado en status, sin salirse de los bordes.
ordena pintar en la posición (x,y) si el pulsador izquierdo del ratón está pulsado.

J.M. Mendías 2024

Diseño principal

esquema RTL: pintando el cursor

```
cursorRender:
process (pixel, line, x,  y)
  -- Puntero 16x16: 0-negro, 1-blanco, 2-transparente
  constant SIZE : natural := 16;
  type pointerRom is array(0 to SIZE*SIZE-1) of natural range 0 to 2;
  constant rom: pointerRom := (
    0,0,2,2,2,2,2,2,2,2,2,2,2,2,2,2,
    0,1,0,2,2,2,2,2,2,2,2,2,2,2,2,2,
    0,1,1,0,2,2,2,2,2,2,2,2,2,2,2,2,
    0,1,1,1,0,2,2,2,2,2,2,2,2,2,2,2,
    0,1,1,1,1,0,2,2,2,2,2,2,2,2,2,2,
    ...
  );
  variable xAddr : natural range 0 to 15;
  variable yAddr : natural range 0 to 15;
begin
  RGB <= ...;
  if ... then
    xAddr := ...;
    yAddr := ...;
    case rom(...) is
      when 0 => RGB <= (others => '0');
      when 1 => RGB <= (others => '1');
      when 2 => null;
    end case;
  end if;
end process;
```

Por defecto visualiza lo indicado por el interfaz gráfico de vídeo

Superpone el cursor cuando se refrescan los píxeles que ocupa su mapa de bits de 16×16

Excepto en los píxeles transparentes

Tareas

1. En la carpeta `DAS/source` crear las carpeta `lab9`.
2. Descargar de la Web los ficheros en:
   - `common:` `vgaGraphicInterface.vhd`, `ps2Interface.vhd`
   - `lab9:` `lab9.vhd` y `lab9.xdc`
3. Completar `common.vhd` con la declaración del nuevo componente.
4. Completar el código omitido en los ficheros:
   - `vgaGraphicInterface.vhd`, `ps2Interface.vhd` y `lab9.vhd`
5. En la carpeta `DAS/projects` crear el proyecto `lab9`.
6. Incluir en el proyecto (sin copiarlos) los siguientes fuentes:
   - `common.vhd`, `synchronizer.vhd`, `edgedetector.vhd`, `ps2Interface.vhd`, `vgaRefresher.vhd`, `vgaGraphicInterface.vhd`, `lab9.vhd` y `lab9.xdc`
7. Sintetizar, implementar y generar el fichero de configuración.
8. Conectar el ratón y el monitor a la placa y encenderla.
9. Volcar el fichero de configuración `lab9.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/