



Arquitecturas Reconfigurables

Tutorial 4: bitmaps

Profesores: Sergio Cuenca y Antonio Martínez

sergio@dtic.ua.es

Dept. Tecnología Informática y Computación

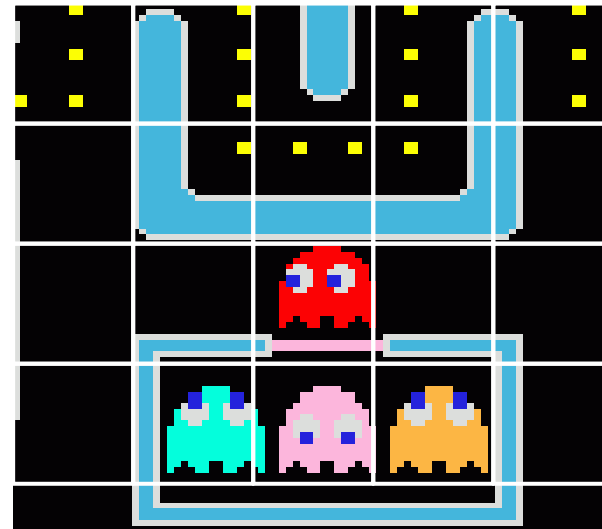
Universidad de Alicante

Para dibujar la pantalla hay que establecer qué bitmap corresponde a cada cuadro del tablero

Tablero N x M cuadros



Bitmaps n x m pixels



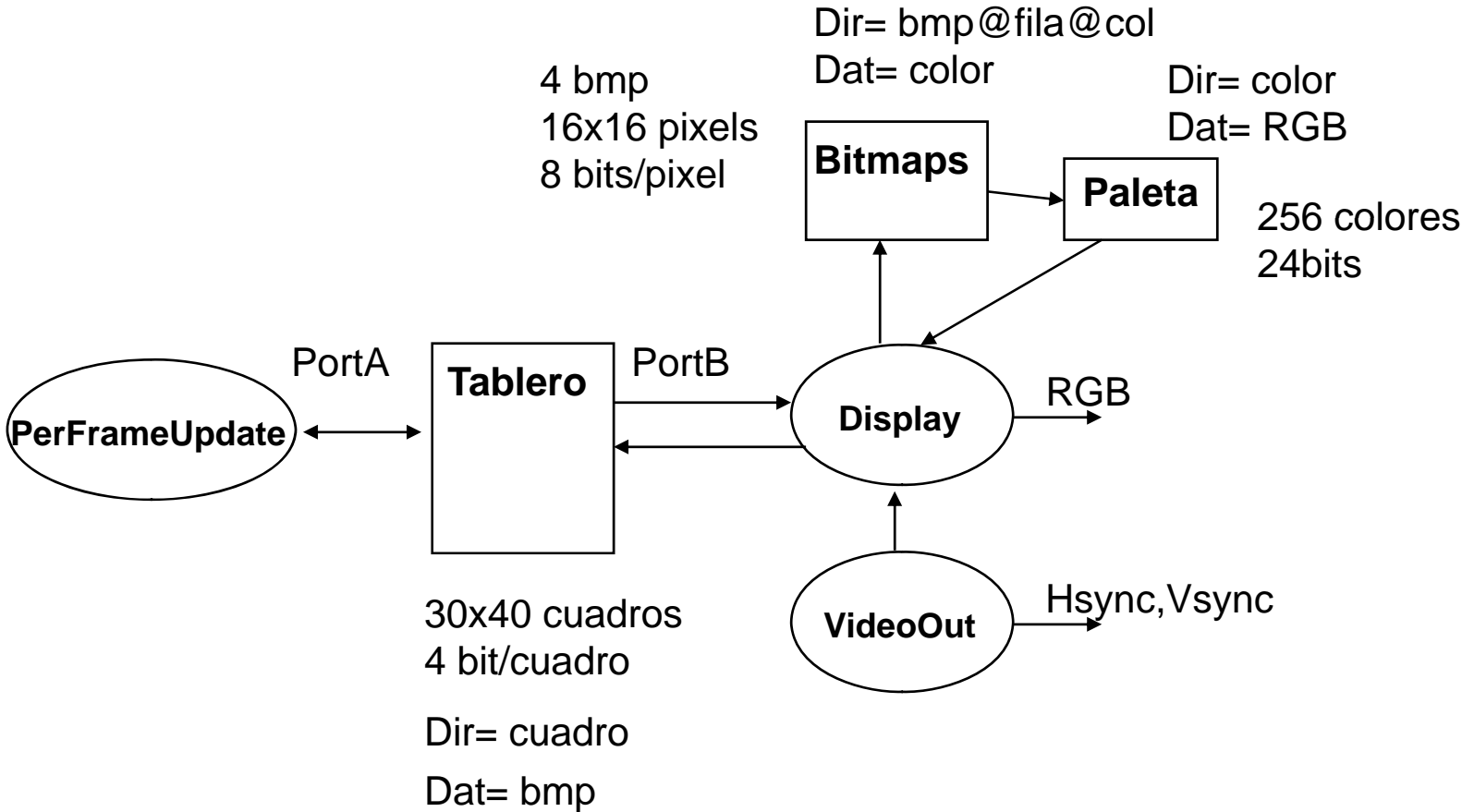
Estructura básica

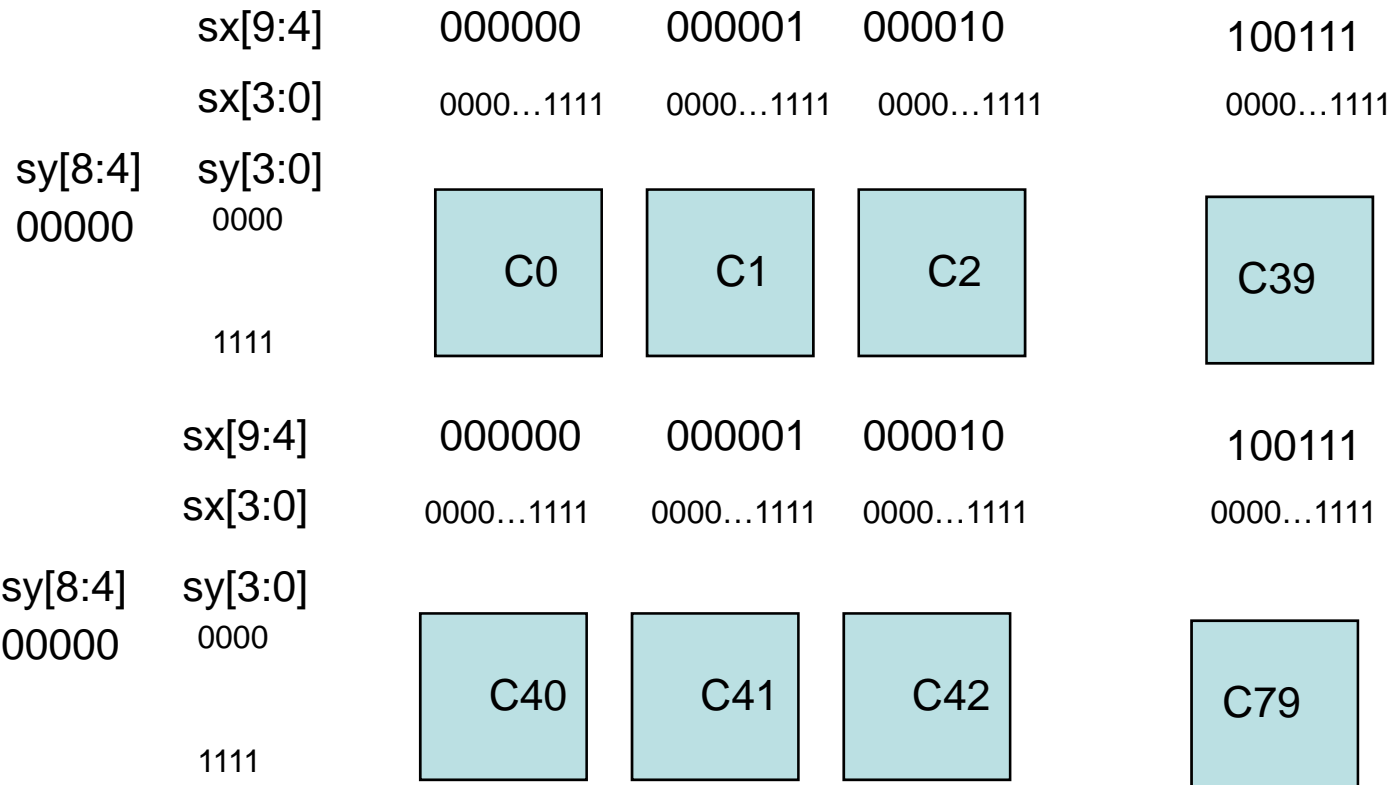
```
// VGA 640x480 TAB 30x40 BMP 16x16
mpram FB //memoria de Tablero doble puerto
{
    ram unsigned 4 A[30*40]; // Read/write port
    rom unsigned 4 B[30*40]; // Read only port
} TAB with {block = "BlockRAM"};

// fuentes o bitmaps
rom unsigned 8 BMP[4*16*16] with {block = "BlockRAM"};
// Paleta de colores
rom unsigned 24 PALETA[256] with {block = "BlockRAM"};

void main (void){
    par{
        while (1){
            Display(Video); // genera imagen de tablero
            PerFrameUpdate(); // actualiza tablero
        }
    }
}
```

Esquema general





Generación Imagen Tablero

```
macro proc Display (Video){
while (1) {
  par {
    InTabX ; InTabY; //verifico si estamos en tablero de juego
    if (sx<VisibleCols & sy<VisibleLines ){
      if (InTabX==1 && InTabY==1 ) {
        par{
          Nbmp=TAB.B[(sy[8:4])@(sx[9:4])]; // leo tablero
          Pixel= BMP [Nbmp@(sy[3:0])@(sx[3:0])]; // leo bmp
          Color = PALETA[Pixel]; // leo color de la paleta
          PalVideoOutWrite (Video, Color); // Envío color
        } //par
      } //if
    else
      PalVideoOutWrite (Video,RedColor);
  }
  else
    PalVideoOutWrite (Video,BlackColor);
} //par
} //while
```

PerFrameUpdate

```
while (1) {  
    while(sx!=VisibleCols || sy!=(VisibleLines-1)){  
        delay; // espera zona blanking  
    }  
    // Actualiza el contenido de los cuadros  
    ... ..  
    ... ..  
}
```