MICROPROCESSADORES II (EMA864315)

GRÁFICOS E ANIMAÇÃO

1º SEMESTRE / 2019

Alexandro Baldassin

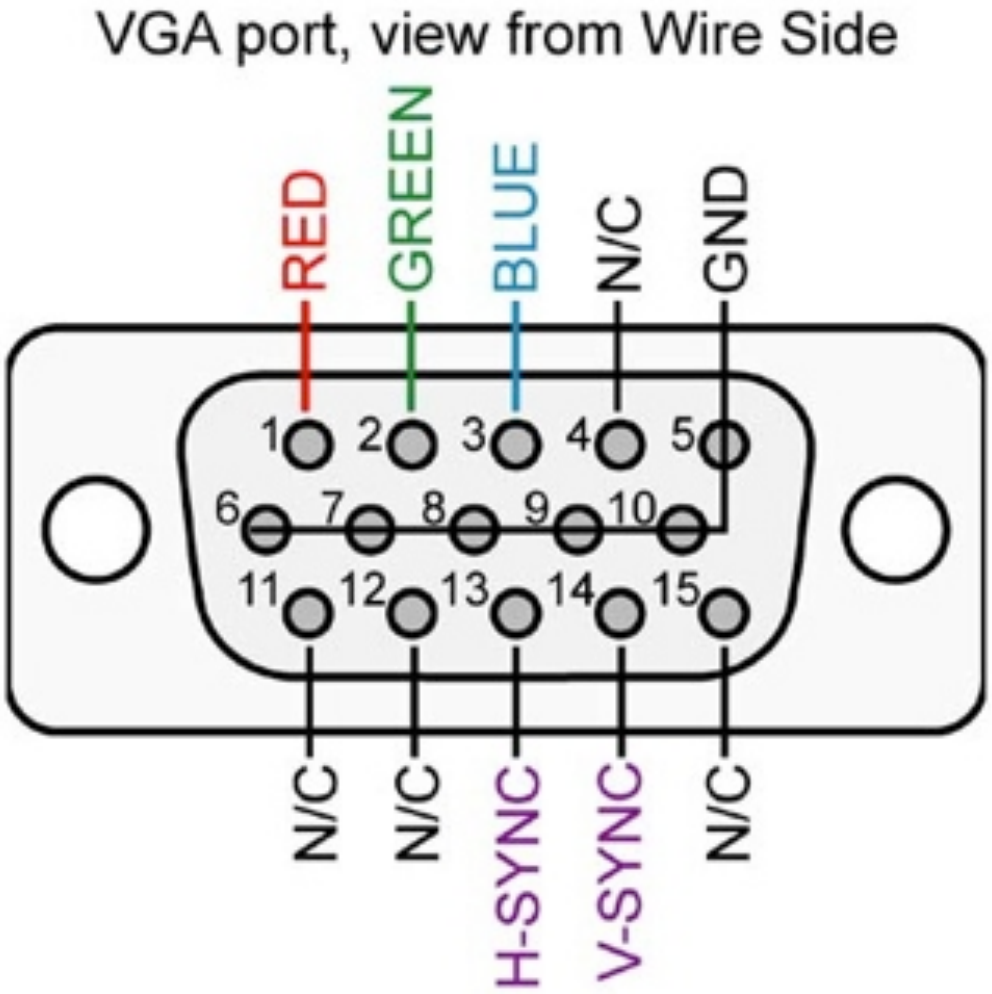
VGA

- **O *Video Graphics Array* (VGA) foi introduzido em 1987 pela IBM na sua nova linha de computadores IBM PS/2**

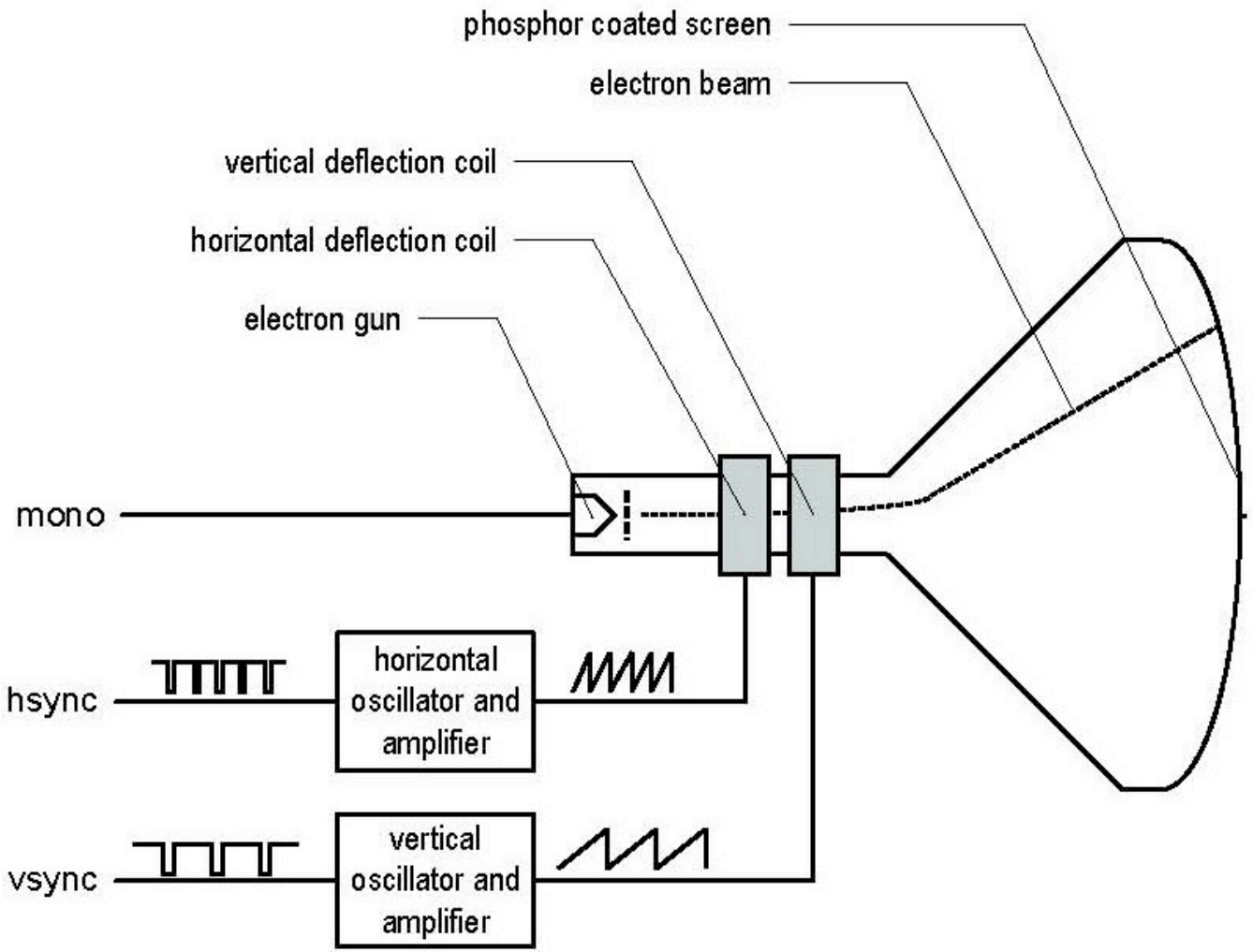
- **O VGA se tornou o padrão para gráficos de computadores em displays analógicos, sendo hoje o denominador comum**
  - Praticamente todo hardware de monitor possui suporte para VGA

- **Usado inicialmente em monitores CRT e mais tarde também adotado por monitores LCD**

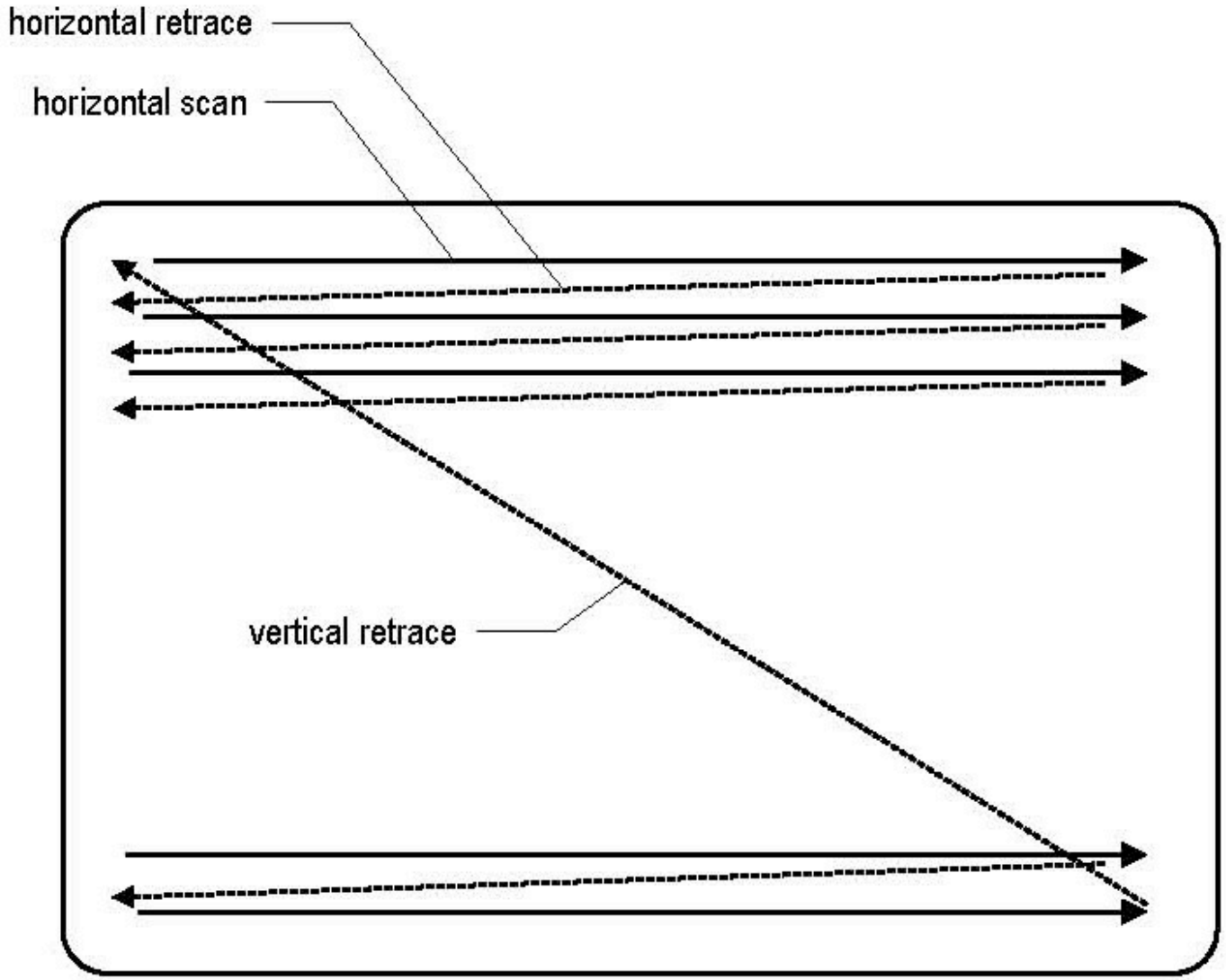
CONECTOR VGA (15 PINOS)



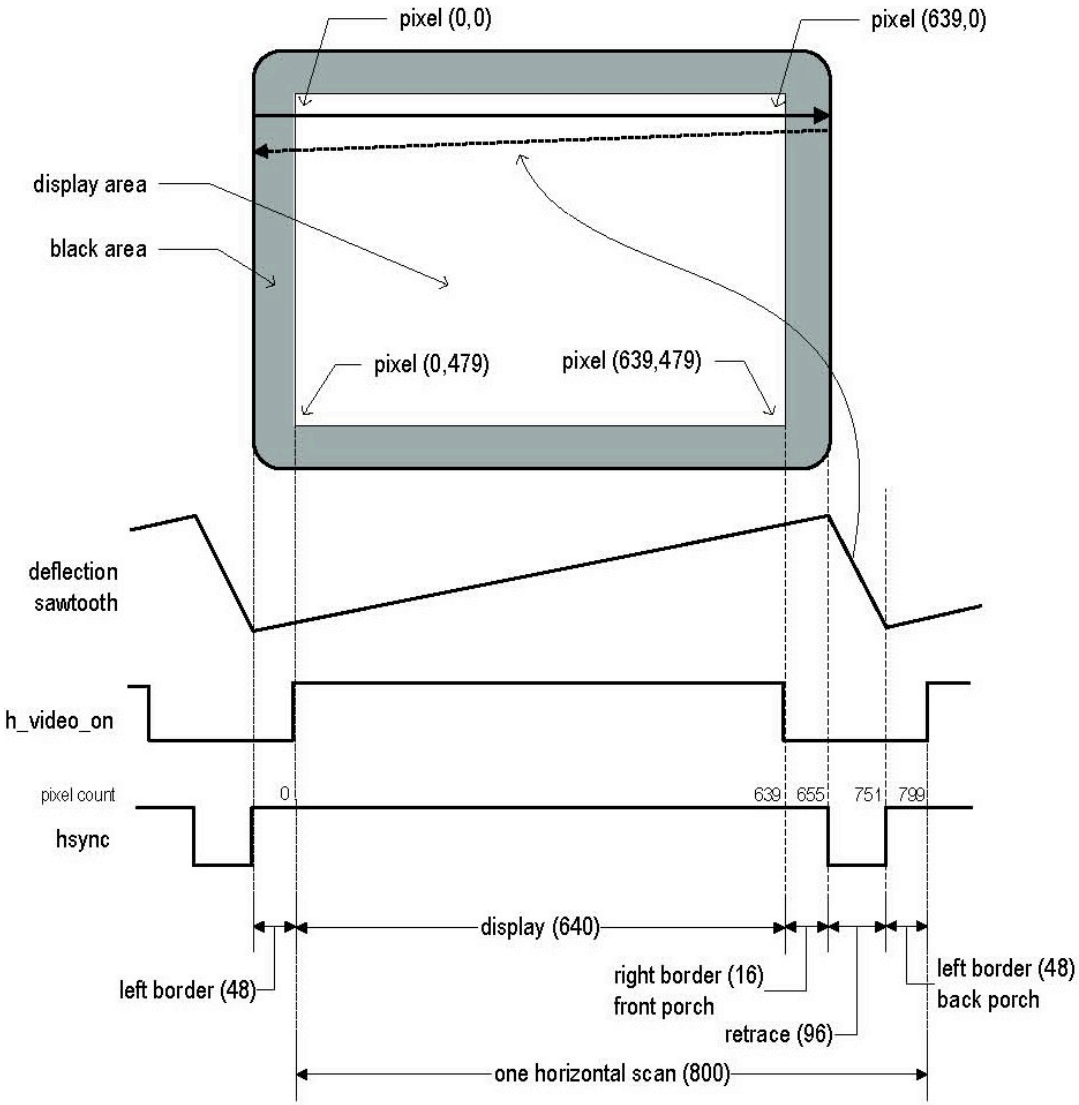
CRT – TUBO DE RAIOS CATÓDICOS



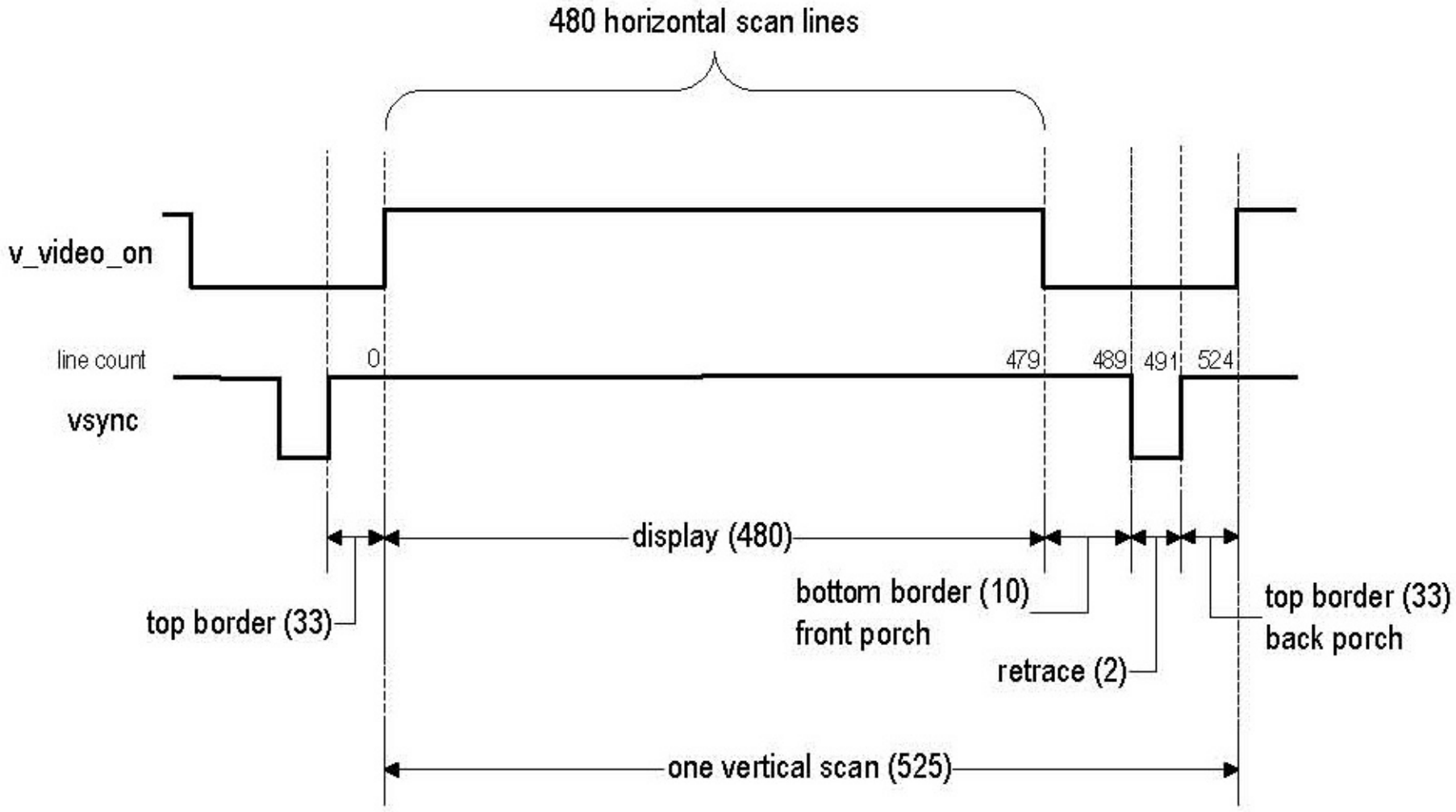
VARREDURAS HORIZONTAIS E VERTICAIS



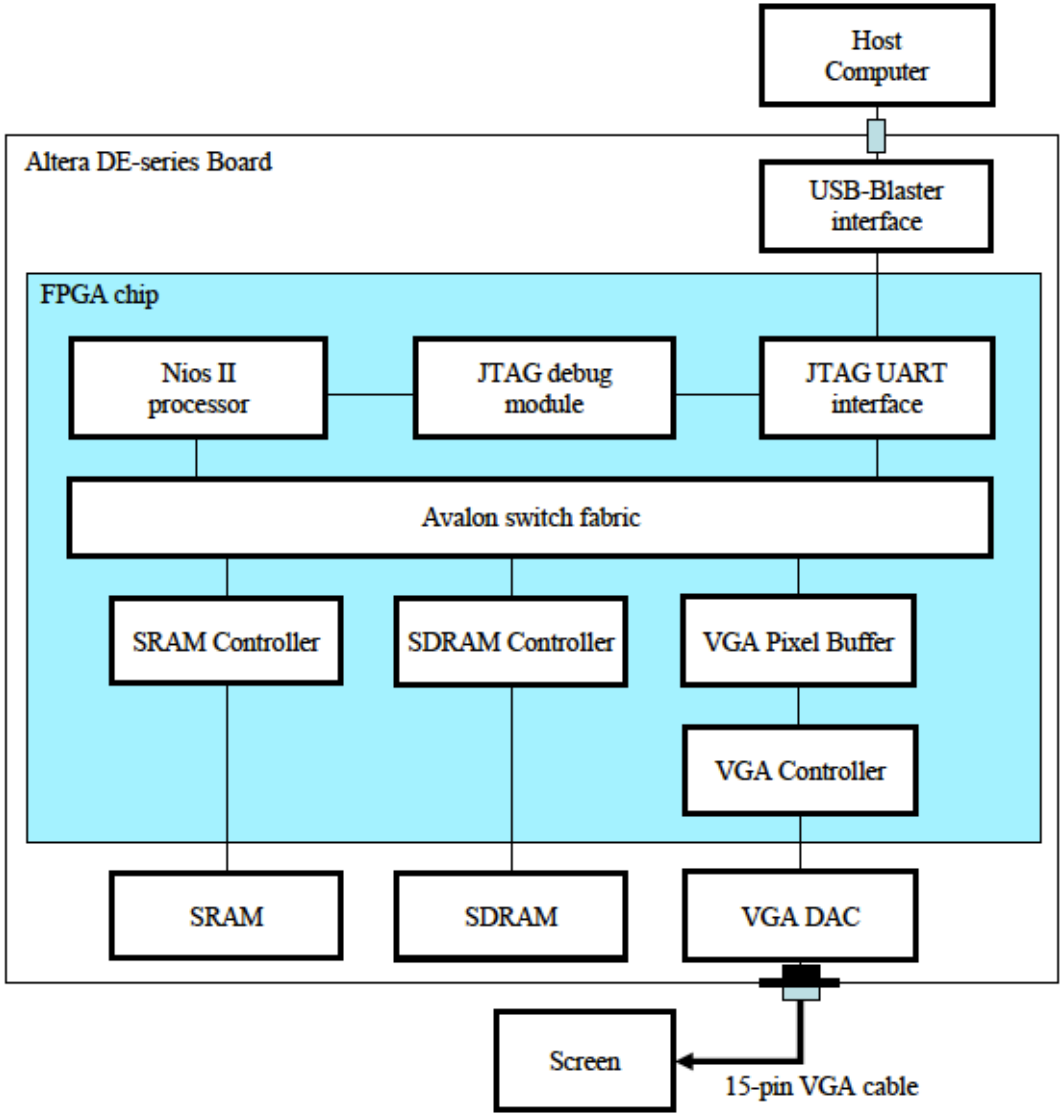
SINCRONIZAÇÃO HORIZONTAL



SINCRONIZAÇÃO VERTICAL



SUPORTE PARA VGA NO DE2



FUNCIONAMENTO GERAL

- **Memória de vídeo é usada para armazenar os dados (pixels) a serem apresentados na tela**
  - Usaremos uma resolução de 320x240 pixels, com 16 bits de cores
  - Memória de vídeo localizada na SRAM, endereço 0x08000000
  - Qual o tamanho da memória de vídeo?

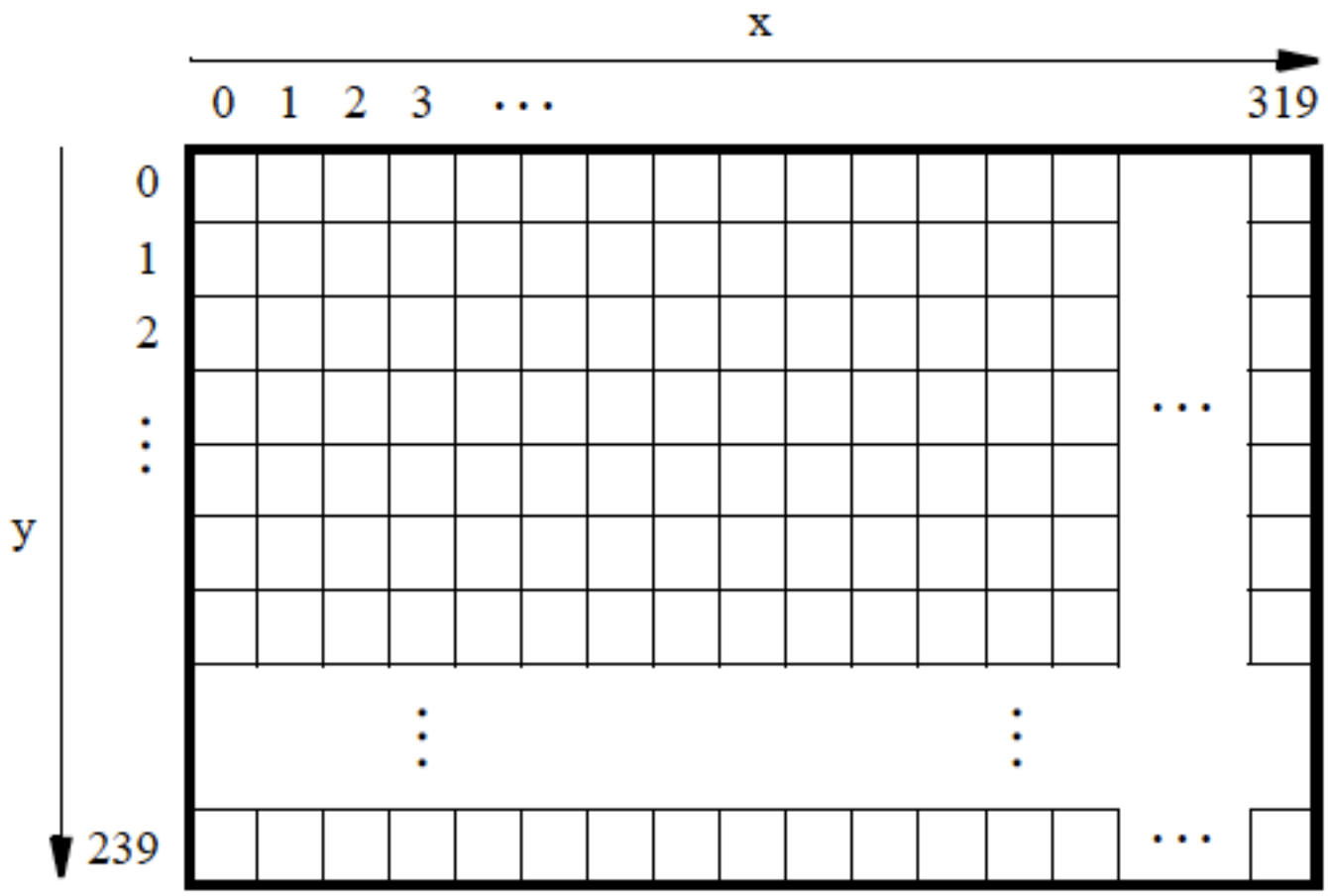
FUNCIONAMENTO GERAL

- **Memória de vídeo é usada para armazenar os dados (pixels) a serem apresentados na tela**
  - Usaremos uma resolução de 320x240 pixels, com 16 bits de cores
  - Memória de vídeo localizada na SRAM, endereço 0x08000000
  - Qual o tamanho da memória de vídeo?
- **Cada pixel é representado por um endereço de memória que armazena sua cor**

FUNCIONAMENTO GERAL

- **Memória de vídeo é usada para armazenar os dados (pixels) a serem apresentados na tela**
  - Usaremos uma resolução de 320x240 pixels, com 16 bits de cores
  - Memória de vídeo localizada na SRAM, endereço 0x08000000
  - Qual o tamanho da memória de vídeo?
- **Cada pixel é representado por um endereço de memória que armazena sua cor**
- **O controlador VGA continuamente lê os pixels da memória e os envia para a tela**
  - Para mudar o conteúdo da tela, a CPU deve alterar o conteúdo da memória de vídeo

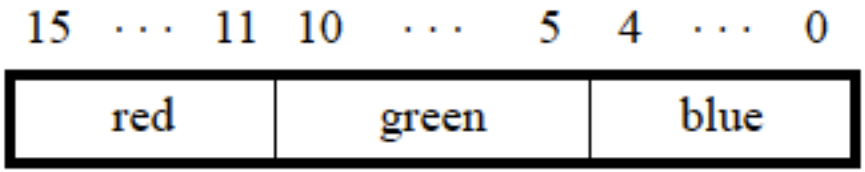
DISPOSIÇÃO DOS PIXELS – 320X240



Cada pixel é representado na memória pela sua cor, de 16 bits

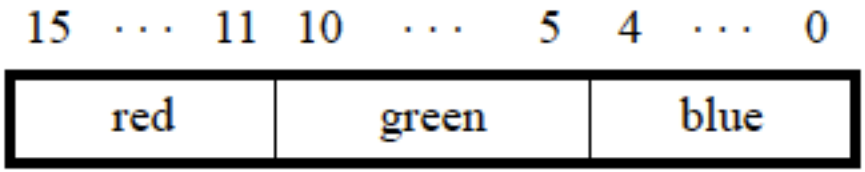
PROPRIEDADES DE CADA PIXEL

- **Cor**



PROPRIEDADES DE CADA PIXEL

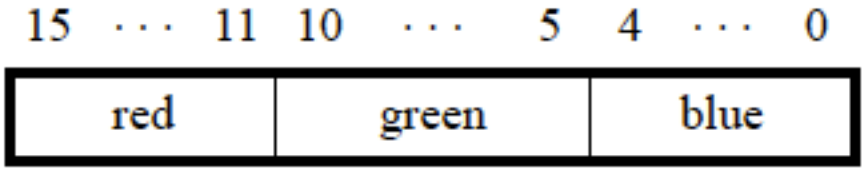
- **Cor**



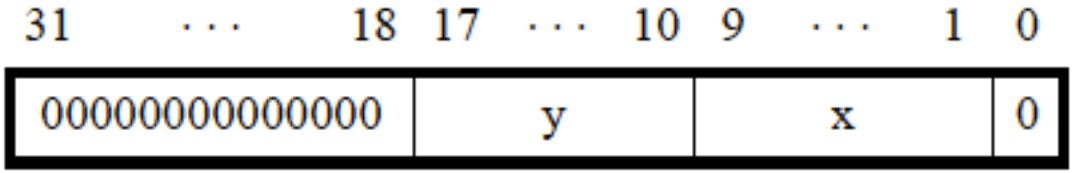
- **Posição**
- Dado um pixel na posição (x,y) como achar o endereço na memória que o representa?

PROPRIEDADES DE CADA PIXEL

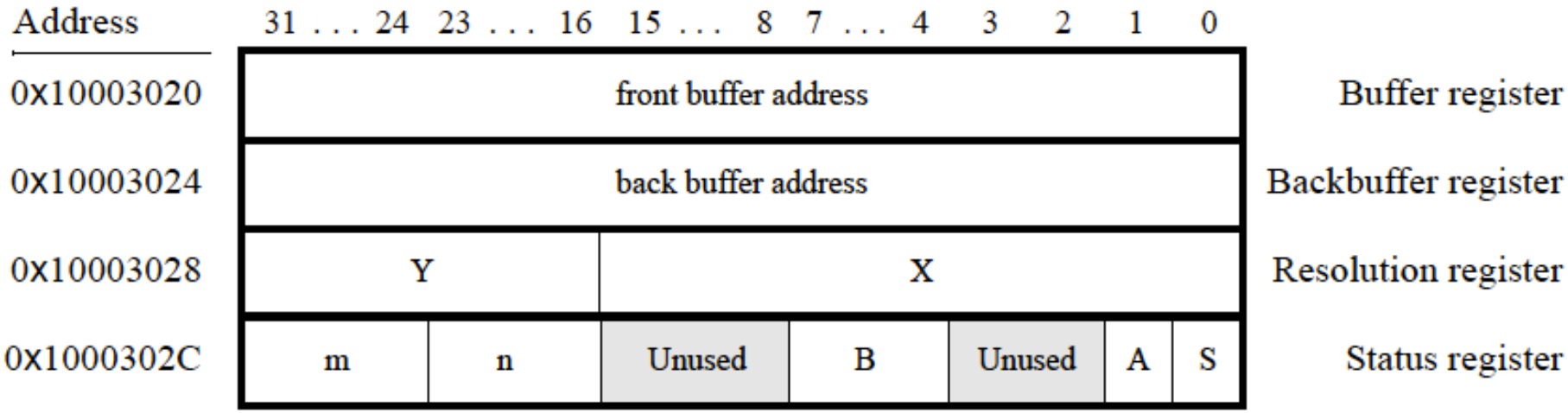
- **Cor**



- **Posição**
- Dado um pixel na posição (x,y) como achar o endereço na memória que o representa?
  - Base: 0x08000000
  - Deslocamento em bytes dado por **(x << 1) + (y << 10)**



VGA PIXEL BUFFER



DO ASSEMBLY À LINGUAGEM C

- **Equivalência**

```
.equ PUSHBUTTON, 0x10000050

…

movia r2, PUSHBUTTON

POLLING:
   ldwio   r3, 12(r2)
   andi  r4, r3, 2
   beq   r4, r0, POLLNG

…
```

DO ASSEMBLY À LINGUAGEM C

- **Equivalência**

```
.equ PUSHBUTTON, 0x10000050

…

movia r2, PUSHBUTTON

POLLING:
   ldwio   r3, 12(r2)
   andi  r4, r3, 2
   beq   r4, r0, POLLNG

…
```

```
int *pushbutton = (int *)0x10000050;

while ((*(pushbutton+3) & 0x2) == 0);
```

DO ASSEMBLY À LINGUAGEM C

- **Equivalência**

```
.equ PUSHBUTTON, 0x10000050

…

movia r2, PUSHBUTTON

POLLING:
   ldwio   r3, 12(r2)
   andi  r4, r3, 2
   beq   r4, r0, POLLNG

…
```

```
int *pushbutton = (int *)0x10000050;

while ((*(pushbutton+3) & 0x2) == 0);
```

```
volatile int *pushbutton = (int *)0x10000050;

while ((*(pushbutton+3) & 0x2) == 0);
```