

Wolfenstein 3D



memória convencional

têm prioridade na alocação de memória. Todos os seus recursos são carregados na memória convencional para evitar uma falha de cache no gerenciador de páginas.

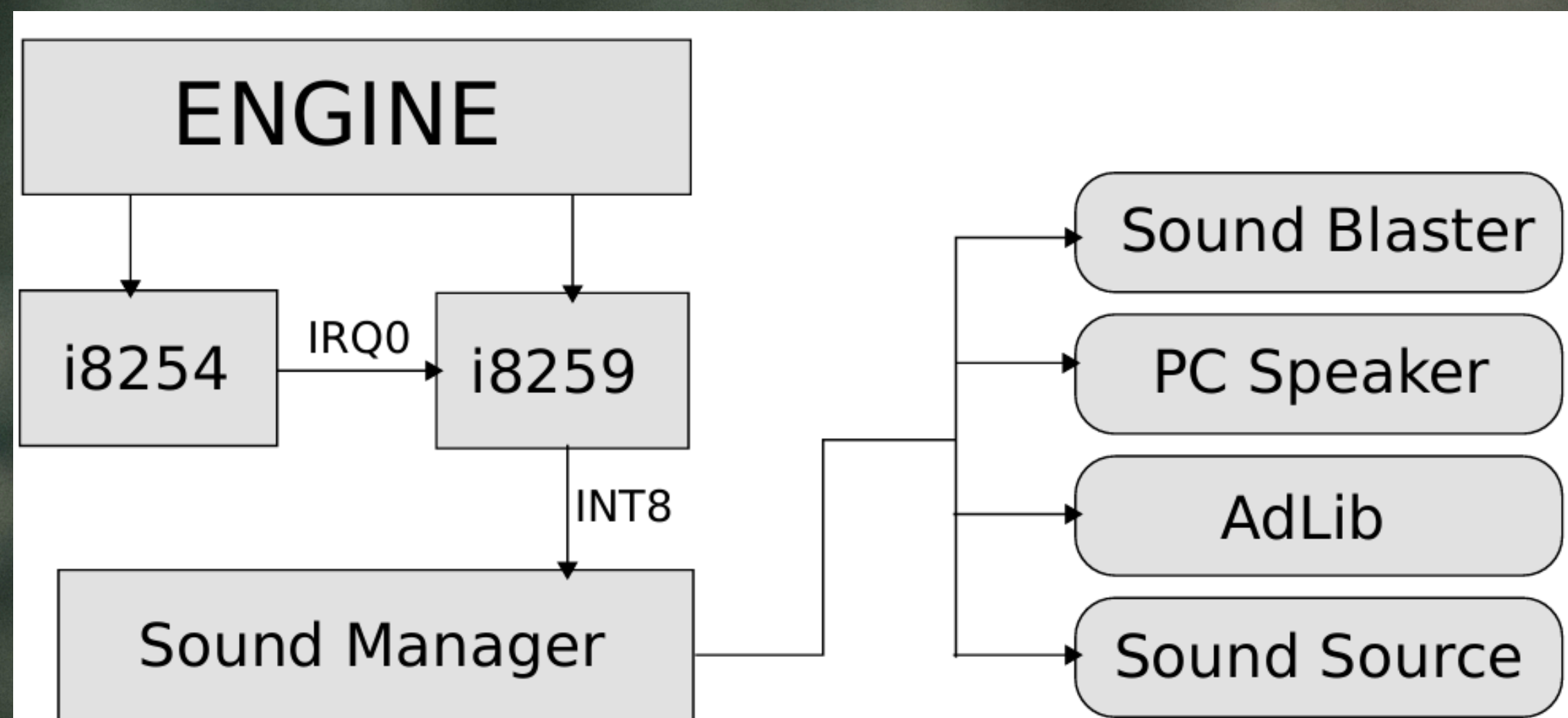


Figure 4.12: Sound system architecture.

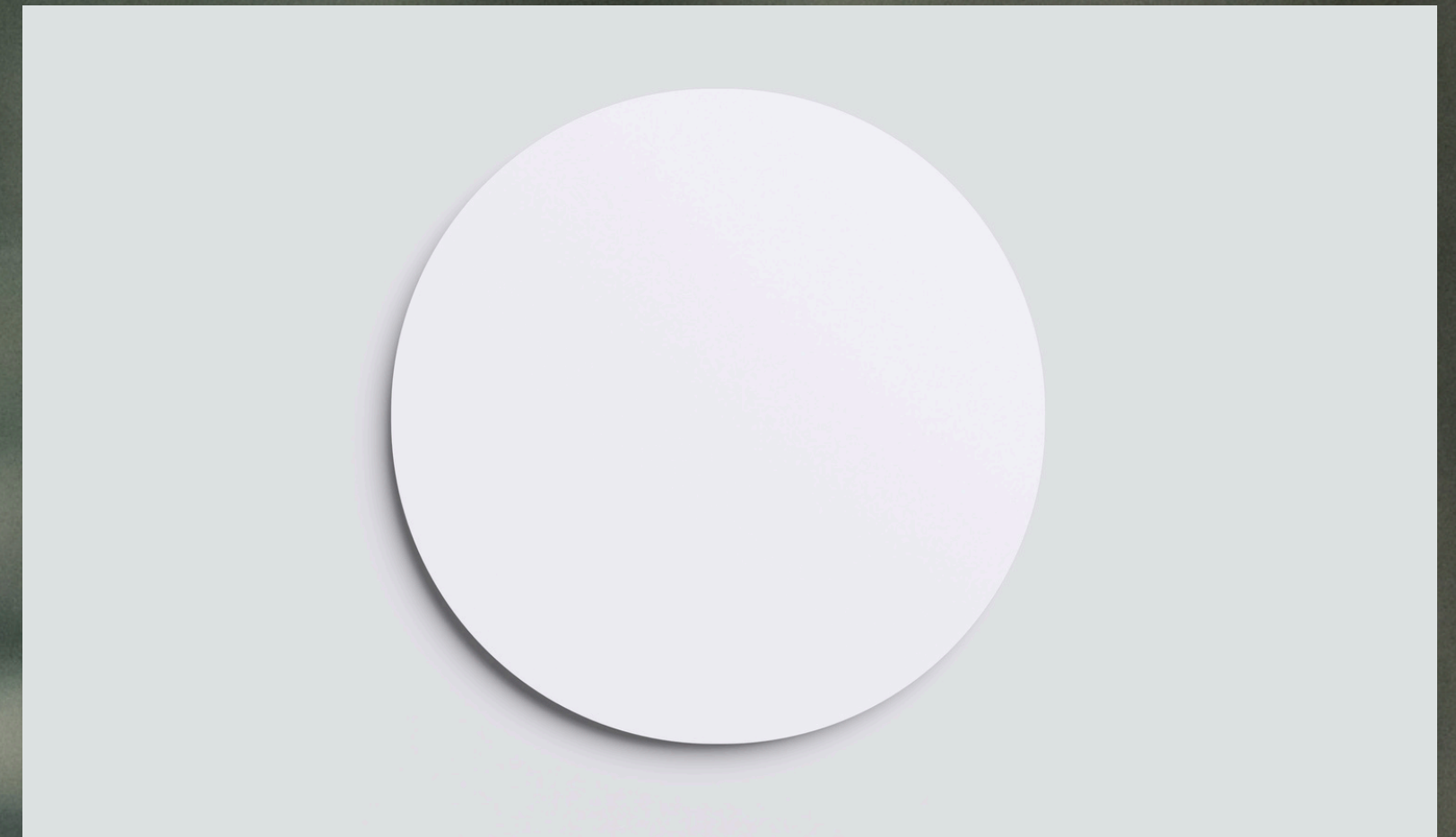
O gerenciador de som é descrito extensivamente na seção "Som e Música".

Input Manager



- O gerenciador de entrada abstrai as interações com joystick, teclado e mouse.

Startup



- À medida que o motor do jogo inicia, ele deve lidar com as dificuldades (capítulo 2)

Signon

O ecossistema heterogêneo de PCs

- diferentes drivers instalados
- diferentes placas de som
- motor precisava descobrir quanta RAM estava instalada e se seria capaz de rodar
- sem recursos



Signon

Além de mostrar dispositivos reconhecidos, como mouse, joystick e placas de som, a métrica mais importante da tela de login é intitulada "MAIN". O programa DOS possui apenas 640KiB de RAM disponível. Cada driver carregado pelo usuário consome parte desses 640KiB. Se o DOS não puder carregar o executável na RAM, o usuário verá a seguinte mensagem de erro:

```
Out of memory
```

A tela de inicialização mostra que o Wolf3D precisa de pelo menos 320KiB de RAM Convencional. "A Barreira dos 640KB".

4.5. Startup

```
// Content of GAMEPAL.OBJ:
// account for 256 * 3 = 768 bytes.
extern byte far gamepal;

// Content of SIGNON.OBJ:
// accounts for 320x200 = 64,000 bytes.
extern char far introsn;

void SignonScreen (void)
{
    unsigned  segstart, seglength;

    VL_SetVGAPlaneMode ();
    VL_TestPaletteSet ();
    VL_SetPalette (&gamepal);

    if (!virtualreality)
    {
        VW_SetScreen(0x8000,0);
        VL_MungePic (&introsn,320,200);
        VL_MemToScreen (&introsn,320,200,0,0);
        VW_SetScreen(0,0);
    }

    // reclaim the memory from the linked signon screen
    segstart = FP_SEG(&introsn);
    seglength = 64000/16;
    if (FP_OFF(&introsn)){
        segstart++;
        seglength--;
    }
    MML_UseSpace (segstart, seglength);
}
```

A variável introsn :

(usando 320x200 bytes = 64.000 bytes) é descarregada da RAM para liberar mais espaço para a execução.

Curiosidades:

As funcionalidades da tela de login são uma referência à "realidade virtual"

Resolvendo o Problema do VGA

Um problema não resolvido:

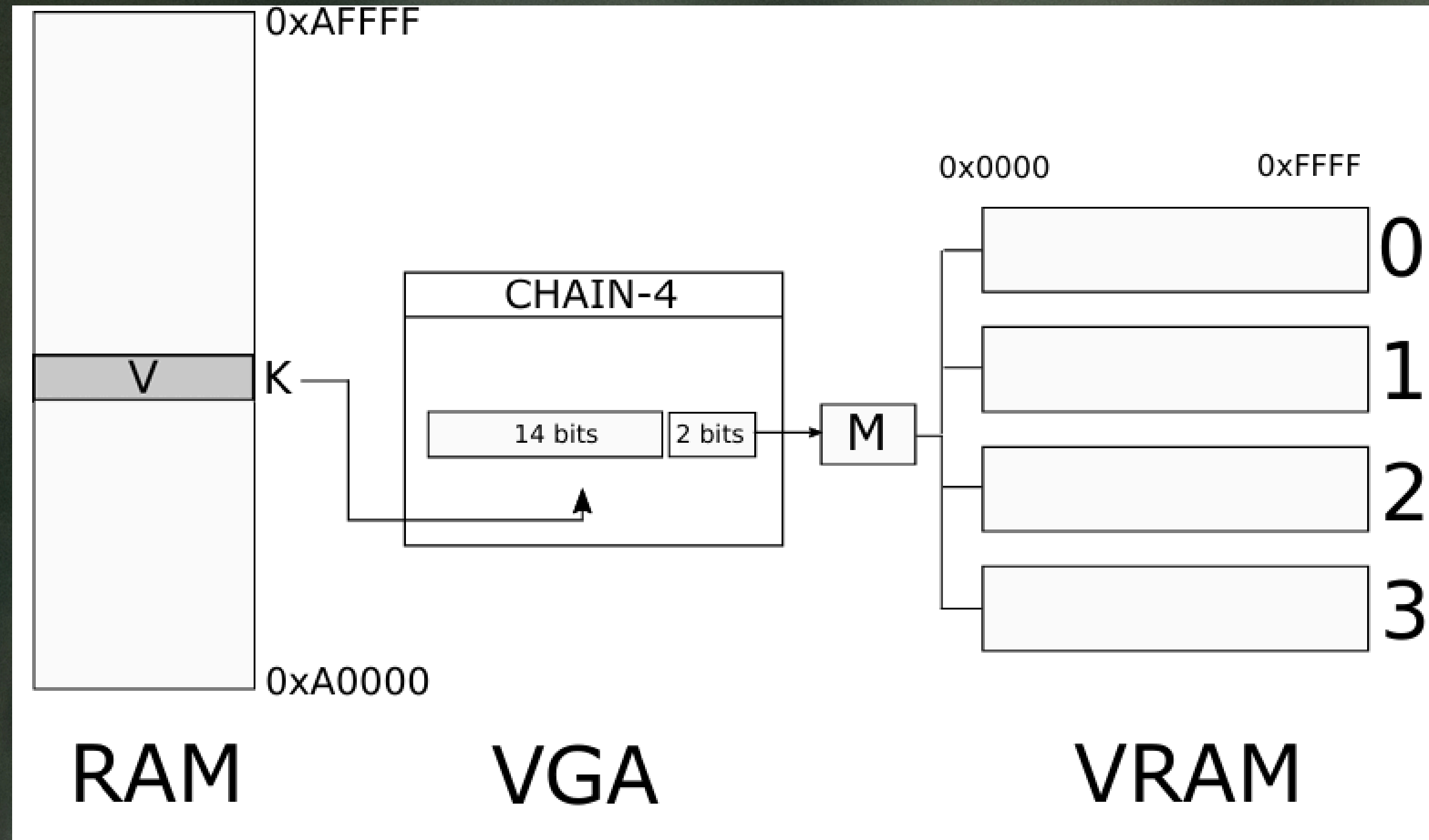
todos os modos VGA carecem de capacidade de duplo buffer. O modo mais atraente (13h) oferece um único framebuffer com resolução de 320x200 pixels não quadrados e 256 cores indexadas. O chipset Chain-4 no circuito VGA mapeia automaticamente a RAM a partir de A0000h para os quatro bancos de VRAM. Quando ativo, um desenvolvedor não precisa se preocupar com os bancos. A tela pode ser limpa com uma função simples **como a seguinte**.

```
char far *VGA = (byte far*)0xA0000000L;

void ClearScreen(void){
    asm mov ax,0x13
    asm int 0x10

    for (int i=0 ; i < 320*200 ; i++)
        VGA[i] = 0x00;
}
```

Resolvendo o Problema do VGA



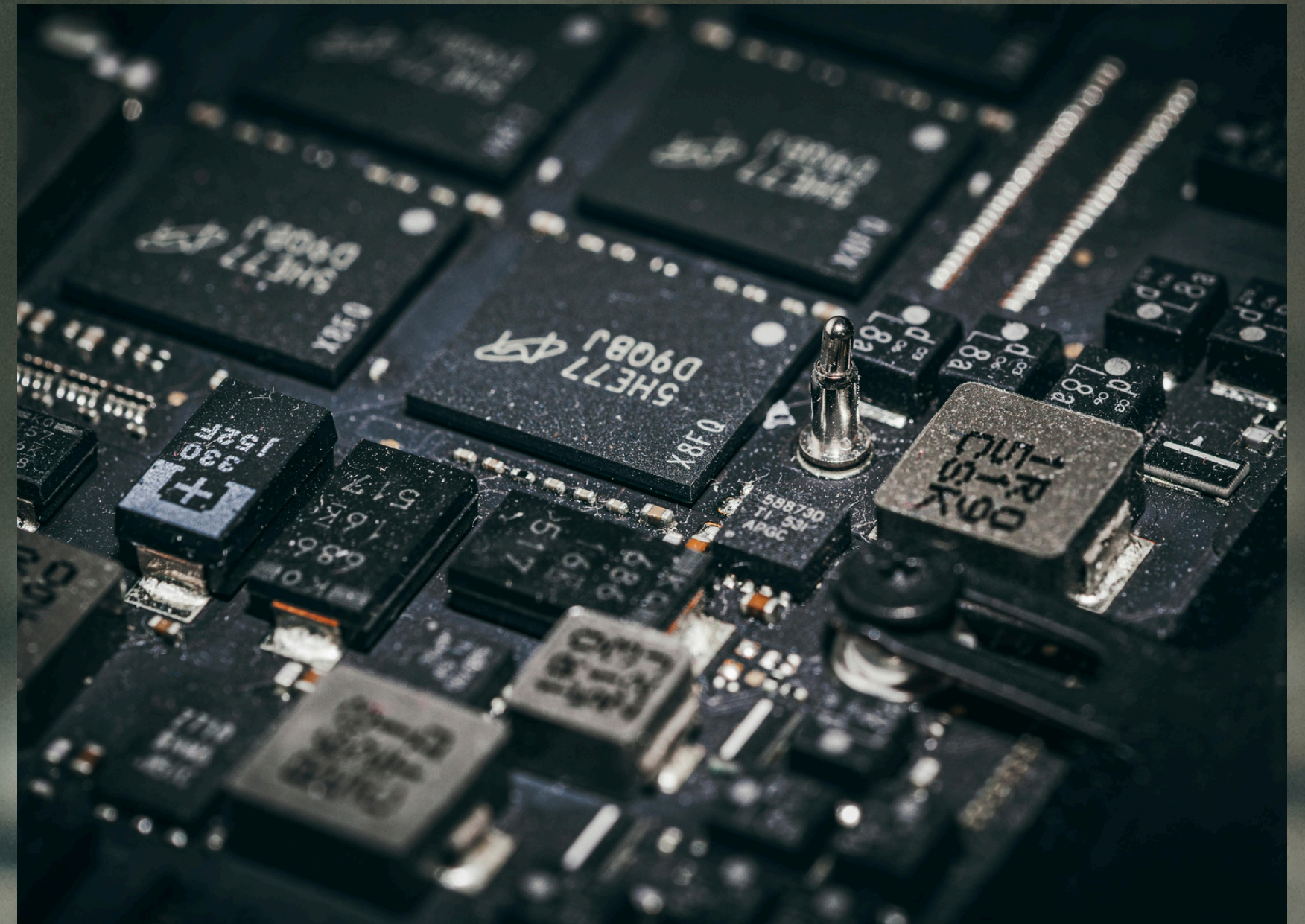
O chipset Chain-4 entre RAM e VRAM encaminha operações de E/S.

sistema de mapeamento

Este sistema de mapeamento também é chamado de "encadeamento".

apenas 14 bits são usados para o deslocamento real no banco.

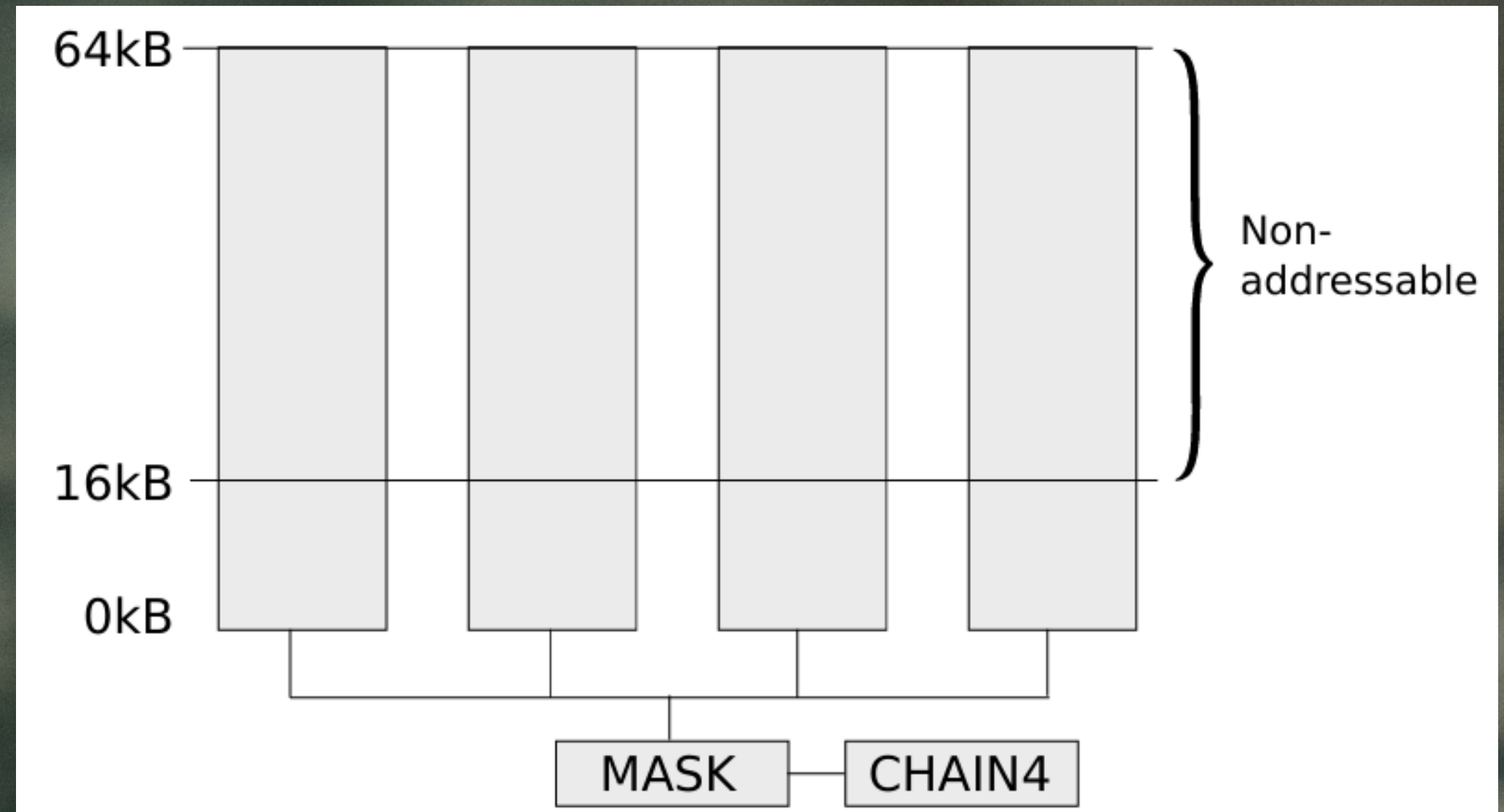
Como 14 bits só podem endereçar 16384 v, este sistema resulta em 75% da VRAM inutilizável.



4.5. Startup

problema é realmente culpa do chip Chain-4?

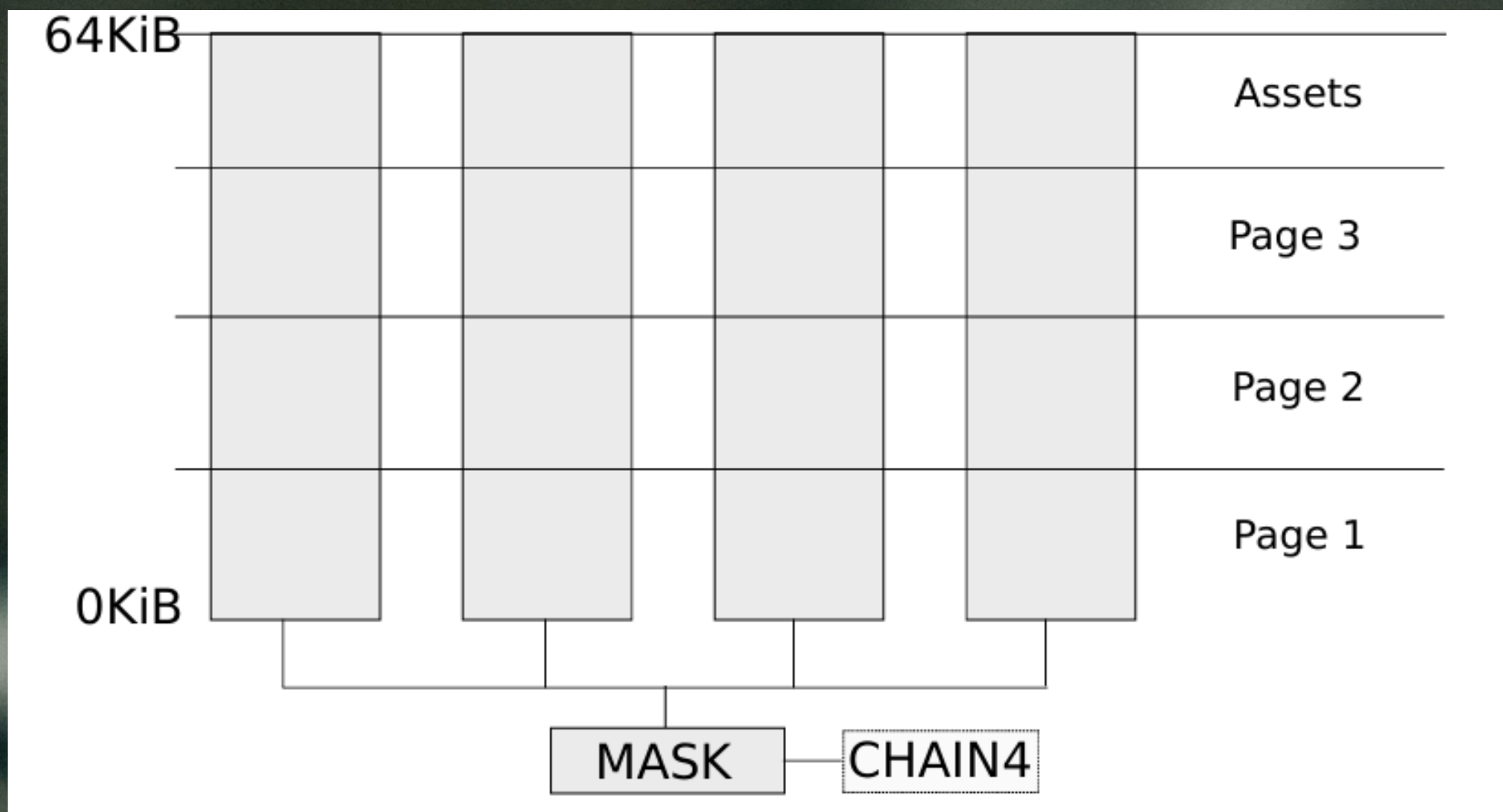
- é possível desativá-lo. chamou de Mode-X.
- A técnica foi popularizada por Michael Abrash
- sequência de operações não documentada que desativa o Chain-4 e permite uma resolução de 320x240 pixels quadrados e acesso total aos 256KiB de RAM.



Wolfenstein 3D usou uma tecnica diferente?

- desativa o Chain-4
- mantém a resolução em 320x200.
- modo foi chamado de Mode-Y um ano depois
- 17% a menos de pixels em comparação com o Mode-X com 320x240 = 76.800 pixels.
- o Mode-X simplesmente tinha pixels demais por quadro. Também teria sido inconveniente para os artistas, já que o Deluxe Paint rodava no Mode 13h, que tem pixels não quadrados.

O Resultado do "Unchain": Dividindo a VRAM



- Ao desativar o Chain-4, a id Software acessou os 256KiB totais da placa VGA.
- A memória foi dividida em 4 partes.
- Três framebuffers independentes de 64.000 bytes cada.
- O restante (70.144 bytes) virou um espaço dedicado aos ativos gráficos.

4.5. Startup

```
#define SC_MAPMASK 0x02
```

```
void selectPlane(char plane) {  
    outp(SC_INDEX, SC_MAPMASK);  
    outp(SC_DATA, 1 << plane);  
}
```

```
void CleanScreen(int color) {  
    for(int y=0 ; y < 200 ; y++) {  
        for(int x=0; x < 320 ; x++) {  
            selectPlane(x % 4);  
            writePixel(x, y, color);  
        }  
    }  
}
```

```
void CleanScreen(int color) {  
    for(int x=0; x < 320 ; x++) {  
        selectPlane(x % 4);  
        for(int y=0 ; y < 200 ; y++) {  
            writePixel(x, y, color);  
        }  
    }  
}
```

O Gargalo da CPU e o Desenho Vertical

- O Preço: Sem o Chain-4, selecionar os bancos de memória via software ficou lento demais.
- Desenhar a tela na horizontal exigia uma alteração de banco a cada pixel.
- A Solução: Eles reescreveram o motor para desenhar a tela verticalmente (coluna por coluna).
- Isso reduziu as chamadas à placa, dobrou a velocidade do jogo, e forçou as artes a serem salvas rotacionadas em 90 graus.

4.5.2 Solving the VGA Problem



0x0000	0x3E80	0x7000	
Page 1	Page 2	Page 3	

O Desafio da Atomicidade e o "Glitch" Visual

- Mudar de página de vídeo exigia atualizar um endereço de 16 bits no controlador do monitor (CRTC).
- O envio não era instantâneo: o processador mandava primeiro o byte alto e depois o byte baixo.
- Se o monitor atualizasse exatamente entre os dois envios, ocorria um problema de sincronia.
- O Glitch: O controlador lia um endereço quebrado (ex: 0x3E00 em vez de 0x3E80), exibindo uma imagem cortada que misturava duas páginas diferentes.

4.5.2 Solving the VGA Problem

```
asm mov cx, startScanOffset

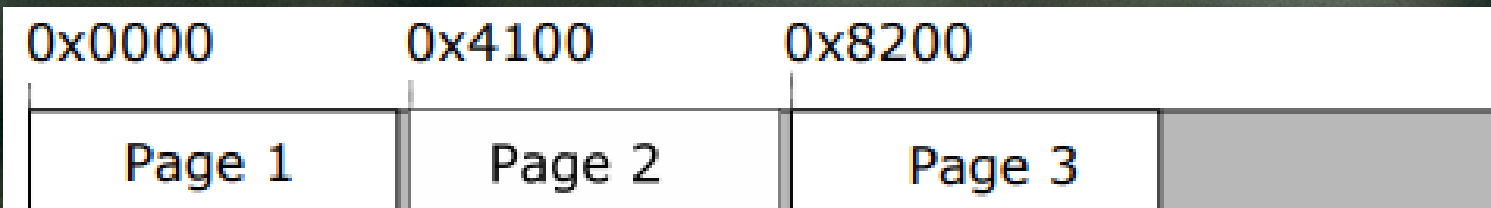
asm mov dx,0x3d4 ; 3d4h is the CRTC register

asm mov al,0xc ; Tell the CRTC we want to update
asm out dx,al ; the start address high register
asm inc dx
asm mov al,ch
asm out dx,al ; set the high byte

;***** CRTC SCAN START HERE !!!!!!!! *****
;***** AND SHOWS 2 PARTIAL FRAMEBUFFERS *****

asm mov al,0xd ; Tell the CRTC we want to update
asm out dx,al ; start address low register
asm inc dx
asm mov al,cl
asm out dx,al ; set the low byte
```

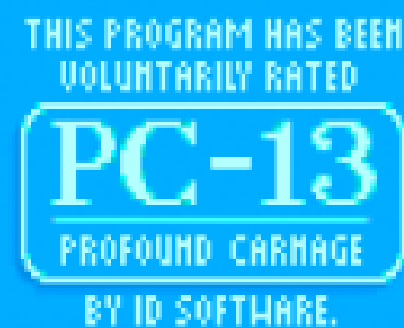
```
#define SCREENBWIDE 80
...
#define SCREENSIZE (SCREENBWIDE*208)
#define PAGE1START 0
#define PAGE2START (SCREENSIZE)
#define PAGE3START (SCREENSIZE*2u)
#define FREESTART (SCREENSIZE*3u)
```



A Solução Genial: "Padding" (Preenchimento)

- A equipe adicionou um espaço invisível extra ("padding") no final de cada página de vídeo.
- Eles configuraram o motor para tratar a tela como se tivesse 208 pixels de altura em vez de 200.
- Isso forçou os endereços das 3 páginas a terminarem sempre em zeros (0x0000, 0x4100 e 0x8200).
- O Resultado: Como o byte baixo era sempre 00, a troca de página exigia atualizar apenas o byte alto. O envio de um único valor de 8 bits tornou a operação atômica e eliminou os glitches.

A Irreverência da id Software: "Profound Carnage"



- Em 1991, não existia classificação etária oficial para videogames.
- O órgão regulador (ESRB) só surgiria em 1994, justamente devido à violência de jogos posteriores (como o próprio Doom da id Software).
- Para demonstrar irreverência, a equipe criou um aviso falso no início do jogo.
- O jogo foi "voluntariamente" classificado como PC-13 ("Profound Carnage" - Carnificina Profunda).
- A tela foi uma paródia direta do sistema de classificação de filmes dos cinemas americanos (o selo PG-13).