

Wolfenstein 3D

Capítulos 1 e 2.1

Brenno Gomes Breda

Daphne Rocha amigo

Isabel Emília Sterim Saade

Maria Carla dos Santos Bellote

Rhuan Santos Wolfgramm

08-04-2026

Resumo

Este artigo resume os Capítulos 1 a 2.1 do livro sobre a engenharia por trás de Wolfenstein 3D. O texto aborda tópicos como a criação do jogo pela id Software e a ousada decisão de utilizar o PC "de escritório" e sua poderosa CPU no lugar de consoles dedicados da época, superando diversos obstáculos técnicos para criar gráficos em 3D. Analisamos o pipeline de hardware do PC em 1991 para compreender como funcionavam seus cinco subsistemas: entradas do jogador, CPU, memória RAM, vídeo e áudio. O artigo também detalha o processador Intel 80386, explorando sua arquitetura interna, dividida em três sistemas principais (Bus Unit, Memory Management Unit e Processing Unit), e o seu pipeline de execução de três estágios (Prefetch, Decode e Execute). Por fim, são discutidos o alto custo de ciclos das instruções e o grande problema da aritmética de ponto flutuante, já que a ausência de um coprocessador matemático (FPU) no 386 tornava os cálculos de frações para o 3D lentos e inviáveis em tempo real.

Sumário

1	Introdução: A Escolha do Hardware e a id Software	3
1.1	Lançamento do Wolfenstein 3D	3
1.2	Sucesso Comercial e Código aberto	3
1.3	Escolha do PC	3
2	Pipeline do PC em 1991	4

3	Desafios Técnicos do Hardware em 1991	4
3.1	Sistema de Vídeo	4
3.2	Unidade Central de Processamento (CPU)	4
3.3	Sistema de Áudio	5
3.4	Memória	5
3.5	Análise Geral dos Subsistemas	5
4	O Processador Intel 80386	5
4.1	A Jornada do Intel 80386	6
4.2	Organização Interna do Intel 80386	7
4.2.1	Bus Unit	8
4.2.2	Memory Management Unit	8
4.2.3	Processing Unit	8
4.3	Instruções	9
5	O Problema de Ponto Flutuante	10

1 Introdução: A Escolha do Hardware e a id Software

1.1 Lançamento do Wolfenstein 3D

Lançado em maio de 1992, **Wolfenstein 3D** estabeleceu o gênero de tiro em primeira pessoa (*First Person Shooter*) e demonstrou que o PC poderia ser uma plataforma para jogos de alto impacto visual, e não apenas uma ferramenta voltada para tarefas de escritório. O sucesso do jogo consolidou a reputação da **id Software**, que tinha grandes expectativas para o projeto desde o início. Inclusive, a empresa chegou a adquirir os direitos do nome "*Wolfenstein*" de um jogo anterior da Apple II, *Castle Wolfenstein* (1981), pelo valor de US\$ 5.000.

1.2 Sucesso Comercial e Código aberto

Em seu primeiro ano, o jogo vendeu mais de 100.000 unidades, sendo amplamente aclamado pelos jogadores. O entusiasmo da comunidade foi tão grande que diversos fãs passaram a desenvolver modificações (*mods*) por meio de engenharia reversa, mesmo sem compreender completamente os segredos por trás da velocidade e das técnicas de renderização 3D utilizadas. Em 1995, em uma decisão incomum para a indústria, quando motores gráficos eram tratados como segredo corporativo, a **id Software disponibilizou o código-fonte completo do jogo**, acompanhado de instruções, permitindo que desenvolvedores explorassem e expandissem suas ideias. Essa iniciativa teve um impacto significativo na popularização e no avanço dos jogos para PC.

1.3 Escolha do PC

Na época, o PC era visto predominantemente como uma máquina de escritório, enquanto os consoles dominavam o mercado de jogos com o uso eficiente de *sprites* e animações fluidas, frequentemente operando a 60 quadros por segundo, como no Super NES e no Sega Genesis. Esses sistemas baseavam-se principalmente na manipulação de coordenadas bidimensionais (x e y).

Entretanto, a proposta da **id Software** era diferente: criar a sensação de um ambiente tridimensional. Para isso, era necessário desenhar a tela pixel a pixel em um *framebuffer*, antes de enviá-la ao monitor, o que os consoles da época não eram capazes de realizar de forma flexível. A solução encontrada foi explorar o poder de processamento da CPU do PC, que superava significativamente o dos consoles em termos de capacidade bruta, tornando possível calcular cada quadro em tempo real e viabilizar a experiência 3D inovadora proposta pelo jogo.

2 Pipeline do PC em 1991

A arquitetura do computador pessoal em 1991 pode ser compreendida como uma pipeline composta por cinco subsistemas principais: entradas do jogador (*inputs*), unidade central de processamento (CPU), memória RAM, vídeo e áudio. Esses componentes atuam de forma sequencial, semelhante a uma linha de montagem, no processamento das informações necessárias para a execução de um jogo.

O fluxo de dados inicia-se nos dispositivos de entrada, como teclado, mouse e joystick, sendo então processado pela CPU. Em seguida, os dados são armazenados e manipulados na memória RAM, até serem enviados aos subsistemas de vídeo e áudio, responsáveis pela saída visual e sonora. Embora funcional, essa arquitetura não havia sido projetada para aplicações gráficas em tempo real, o que resultava em diversas limitações quando aplicada ao desenvolvimento de jogos.

3 Desafios Técnicos do Hardware em 1991

A adaptação do PC, originalmente projetado para aplicações de escritório, ao desenvolvimento de jogos digitais impôs uma série de desafios técnicos significativos. Cada subsistema apresentava limitações específicas, exigindo soluções criativas por parte dos desenvolvedores.

3.1 Sistema de Vídeo

O padrão VGA apresentava restrições importantes para renderização gráfica. A ausência de suporte a *double buffering* dificultava a obtenção de animações fluidas, resultando em artefatos visuais conhecidos como *screen tearing*. Além disso, os pixels não eram perfeitamente quadrados, causando distorções na imagem, e o acesso ao framebuffer era limitado pela baixa velocidade de comunicação com o barramento de vídeo.

3.2 Unidade Central de Processamento (CPU)

A CPU da época, embora relativamente poderosa, operava predominantemente com números inteiros. No entanto, gráficos tridimensionais exigem cálculos com números fracionários e funções trigonométricas. A ausência de uma unidade de ponto flutuante (FPU) tornava operações com *float* extremamente lentas, uma vez que precisavam ser emuladas por software, tornando-as inviáveis para aplicações em tempo real. Como alternativa, os desenvolvedores recorreram a técnicas baseadas em aritmética inteira para manter o desempenho.

3.3 Sistema de Áudio

O subsistema de áudio também apresentava limitações significativas. O PC Speaker, dispositivo padrão da época, era capaz apenas de emitir sinais sonoros simples e de baixa qualidade. Além disso, o ecossistema de placas de som era fragmentado, exigindo implementações específicas para cada tipo de hardware, o que dificultava a padronização e o desenvolvimento de soluções sonoras mais avançadas.

3.4 Memória

A memória representava outro desafio relevante. O modelo de memória segmentada do MS-DOS tornava o acesso aos dados mais complexo, exigindo cálculos adicionais para manipulação de endereços. Somado a isso, havia a limitação de aproximadamente 1 MB de memória diretamente endereçável, o que restringia significativamente a quantidade de dados que poderiam ser utilizados simultaneamente.

3.5 Análise Geral dos Subsistemas

De forma geral, os subsistemas apresentavam diferentes níveis de dificuldade. As entradas e a memória eram relativamente gerenciáveis, embora com restrições, enquanto a CPU impunha limitações arquiteturais importantes. Por outro lado, os sistemas de vídeo e áudio configuravam os maiores obstáculos, devido às suas severas limitações técnicas.

Esse conjunto de restrições evidencia que o desenvolvimento de jogos para o PC em 1991 exigia não apenas conhecimento técnico, mas também um alto grau de criatividade. O sucesso de títulos como *Wolfenstein 3D* demonstra a capacidade dos desenvolvedores de explorar ao máximo o hardware disponível, frequentemente transformando limitações em soluções inovadoras.

4 O Processador Intel 80386

A relevância da CPU era tamanha que os computadores eram conhecidos não pelo modelo, mas pelo processador interno, de forma que qualquer máquina com o chip Intel 80386 era referida popularmente apenas como um "386". A Intel comercializava duas versões com capacidade idêntica de cálculo, porém com estruturas físicas diferentes: o **386-DX** e o **386-SX**. O DX possuía um barramento completo de 32 bits, enquanto o modelo SX (mais popular e acessível) foi arquitetado com um barramento de apenas 16 bits para aproveitar antigas placas do processador 286 e cortar o preço.

4.1 A Jornada do Intel 80386

O desenvolvimento do processador **Intel 80386** não começou como o projeto principal da empresa. Inicialmente, ele era um projeto secundário conduzido por uma pequena equipe em San Jose, enquanto outro grupo da Intel trabalhava no processador **Intel i960**, que utilizaria um novo conjunto de instruções e suporte a linguagens de alto nível diretamente em hardware. Quando o projeto do i960 enfrentou problemas de desempenho, o 386 passou de projeto paralelo a foco principal da empresa.

Duas decisões de projeto foram fundamentais para o sucesso do 80386. A primeira foi a escolha de **manter compatibilidade com processadores anteriores (SX)**, especialmente com o **Intel 80286**, abandonando a ideia de um novo conjunto de instruções. A segunda foi a introdução de um **modo operacional de 32 bits (DX)**, que solucionava diversas limitações de endereçamento de memória presentes no 286.

Fabricado com um processo de **1,5 μm** , o chip teve sua área ampliada de **49 mm^2 para 104 mm^2** , permitindo integrar aproximadamente **275.000 transistores**, mais que o dobro do seu antecessor. Essas melhorias consolidaram o 80386 como um dos processadores mais importantes da arquitetura x86 no início da década de 1990.

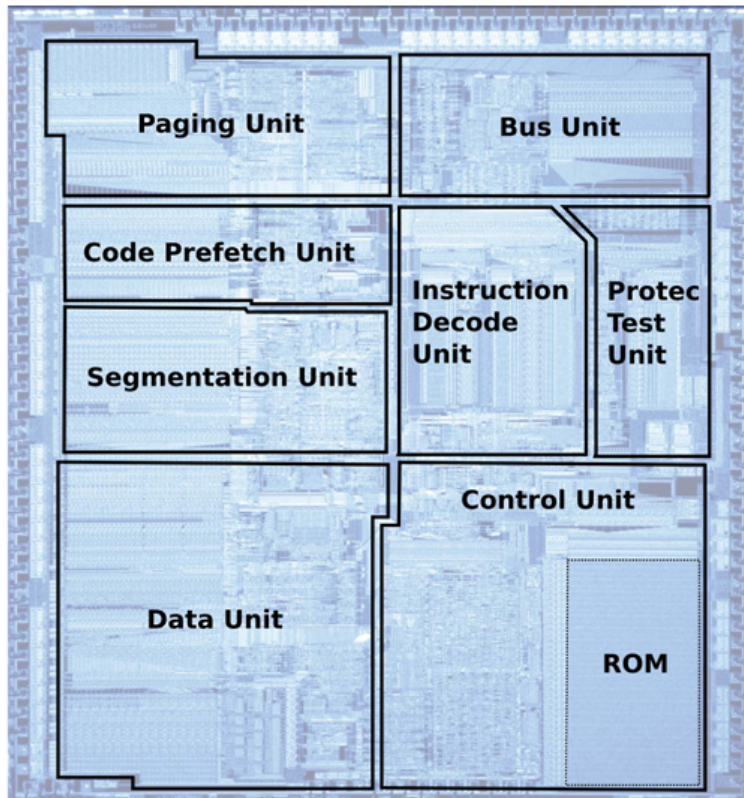


Figura 1: Layout do processador Intel 80386, mostrando a organização interna do chip.

4.2 Organização Interna do Intel 80386

A arquitetura do Intel 80386 pode ser resumida em três sistemas e um pipeline de instruções de três estágios.

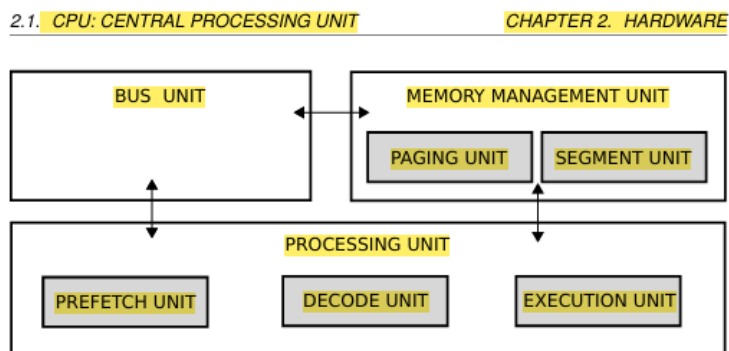


Figura 2: Die (o chip físico) do Intel 80386

Na imagem acima (Figura 2) é possível identificar três sistemas principais, sendo eles: **Bus Unit**, **Memory Management Unit** e **Processing Unit**. Cada um possui funções bem definidas sendo elas:

4.2.1 Bus Unit

responsável pela comunicação entre o processador e os componentes externos, como memória RAM e dispositivos da placa-mãe. Ela gerencia o envio e recebimento de dados através do barramento do sistema. Além de ser a principal diferença entre as versões **386-SX** e **386-DX**. O modelo **SX** possui um **barramento de 16 bits**, o que permitiu reutilizar placas-mãe do **Intel 286** e **reduzir significativamente os custos**. Já o **386-DX** possui uma **Bus Unit totalmente de 32 bits**, oferecendo maior capacidade de transferência de dados.

4.2.2 Memory Management Unit

encarregada do gerenciamento de memória, realizando a tradução de endereços e organizando o acesso à memória. É composta pelas unidades de *segmentação* e *paginação*.

- **A Unidade de Segmentação** divide a memória em segmentos lógicos, como código do programa, dados e pilha. Cada segmento possui um endereço base e um limite, permitindo que o processador organize melhor as diferentes partes de um programa.
- **A Unidade de Paginação** divide a memória em blocos menores de tamanho fixo chamados páginas. Esses blocos permitem que o sistema operacional gerencie a memória de forma mais eficiente, podendo mover páginas entre a memória RAM e o disco quando necessário.

4.2.3 Processing Unit

representa o núcleo de processamento da CPU, onde as instruções são efetivamente executadas. Ela contém os registradores, a ALU (Unidade Lógica Aritmética) e implementa o pipeline de execução das instruções.

Na processing Unit se encontram as três unidades de grupo de execução que formam o pipeline de três estágios: **Prefetch Unit**, **Decode Unit** e **Execution Unit**.

- **Prefetch Unit (Busca)**: responsável por buscar antecipadamente as próximas instruções do programa na memória e armazená-las em uma fila (*FIFO*) de até 16 bytes, deixando-as prontas para execução. Esse mecanismo opera de forma linear e não consegue prever desvios de execução; por isso, quando ocorre uma instrução de salto como `JMP`¹, todo o pipeline precisa ser descartado e reiniciado.

¹A instrução `JMP` (Jump) altera diretamente o fluxo de execução do programa, fazendo com que o processador passe a executar instruções a partir de outro endereço de memória.

- **Decode Unit (Decodificação):** recebe as instruções previamente buscadas e realiza sua interpretação, traduzindo-as para operações compreensíveis pelo processador. O resultado da decodificação é armazenado temporariamente em uma pequena fila interna para ser utilizado pela etapa seguinte.
- **Execution Unit (Execução):** etapa responsável por executar efetivamente as instruções utilizando a ALU² e os registradores do processador. Devido à ausência de memória cache rápida (SRAM) no chip e à relativa lentidão do decodificador, até mesmo instruções simples como ADD ou INC demandavam pelo menos dois ciclos de clock para serem concluídas.

Observe a figura abaixo 3:

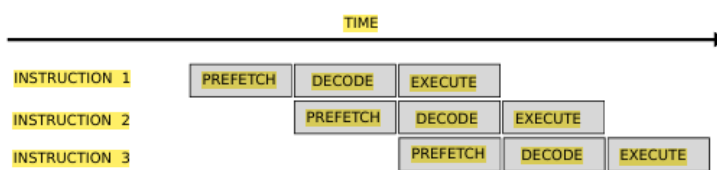


Figura 3: Pipeline de três estágios

O processador não espera uma instrução terminar para começar a próxima. Ele sobrepõe as etapas de várias instruções ao mesmo tempo.

4.3 Instruções

O processador 386 pode ser pensado em 4 partes, a partir da perspectiva da programação, a CPU pode ser separada em:

1. **Unidade Lógica Aritmética (ALU)**
2. **16 Registradores:**
 - Registradores de uso geral (EAX, EBX, ECX, EDX): usados para operações aritméticas, armazenamento temporário e passagem de dados.
 - Registradores de índice (ESI, EDI, EBP, ESP): auxiliam em acesso à memória e controle da pilha (ESP = topo da pilha, EBP = base da pilha).
 - Registradores de segmento (CS, DS, ES, FS, GS, SS): definem segmentos de memória (código, dados, pilha, etc.).
 - Registrador de status (EFLAGS): guarda flags de controle e resultado de operações (zero, carry, overflow, etc.).

²A ALU (*Arithmetic Logic Unit*, ou Unidade Lógica Aritmética) é o componente do processador responsável por executar operações matemáticas e lógicas, como soma, subtração, comparações e operações booleanas.

- Contador de programa (EIP): aponta para a próxima instrução a ser executada.
3. **Um barramento de endereços** para uma memória RAM endereçável plana³, de 32-bits à 4GB.
 4. **Unidade de Paginação de Memória**

Embora seu design de Pipeline tenha permitido a sobreposição de etapas de execução, o processador 386 não conseguia realizar uma operação em menos de 2 ciclos do clock, mesmo operações simples como o operador `ADD reg16, reg16`. Isso ocorre pela falta do cache interno, com uma unidade de decodificação lenta, não suficiente para realizar uma operação por ciclo.

5 O Problema de Ponto Flutuante

A criação de gráficos em 3D exige cálculos intensivos de frações e trigonometria, o que normalmente seria processado através de aritmética de ponto flutuante (*floating-point*), onde uma janela matemática contendo um expoente protege números grandiosos e minúsculos de sofrer transbordamento.

O imenso problema para construir *Wolfenstein 3D* foi que **a CPU Intel 386 vinha equipada apenas com uma ALU para números inteiros, sem ter uma Unidade de Ponto Flutuante (FPU) em seu hardware.**

Sempre que códigos demandavam cálculos fracionários em ponto flutuante, a CPU do PC se via obrigada a emular todas as etapas de aproximação usando software, isto é, executando uma sequência de instruções inteiros para aproximar o resultado. Isso gerava um **processamento incrivelmente lento e impróprio para qualquer aplicação renderizada em tempo real.**

Os usuários que realmente precisassem acelerar esses cálculos tinham que comprar, separadamente, um hardware adicional de coprocessador matemático da marca — como o **Intel i387** — cujo valor ultrapassava a casa dos 200 dólares, atraindo principalmente o nicho acadêmico científico.

Como o custo operacional de usar divisão ou multiplicação convencional de inteiros também consumia enormes porções dos ciclos de desempenho do computador (`DIV` custando 27 ciclos e `IMUL` podendo levar até 259), os programadores acabaram imprensados num dilema entre inteiros que eram imprecisos e frações que paralisavam as máquinas.

³Uma RAM endereçável plana é uma arquitetura de gerenciamento de memória onde o sistema operacional e os programas veem a memória física como um único bloco contínuo e linear de endereços.

Referências