

DESCRIÇÃO TÉCNICA: PS2

Lucas Caetano do Nascimento

Rodrigo Fernando da Silva

Valter J. da Costa Paiva Junior

Criação do console

Fabricado pela Sony como parte da série PlayStation, o PS2 é um console da sexta geração, começou a ser desenvolvido em março de 1999 e foi lançado primeiramente no Japão em 4 de março de 2000, alcançando a marca de 980.000 unidades vendidas um dia após seu lançamento. Seus principais concorrentes eram: O Dreamcast da Sega, Xbox da Microsoft e GameCube da Nintendo.

O PS2 é o console mais vendido de todos os tempos, ultrapassou a marca de 150 milhões de unidades vendidas após 10 anos e 11 meses do seu lançamento no Japão, segundo a Sony, o PS2 tinha 10.828 títulos disponíveis para o sistema e vendeu 1,52 bilhões de títulos desde seu lançamento. Seu tempo de vida foi de 10 anos, porém segundo a Sony, seu ciclo de vida só acabaria quando os desenvolvedores pararem de criar novos jogos para o sistema e a venda de jogos e do console cessar.

Quando a Sony começou a desenvolver o PS2, ela queria que o seu sistema fosse capaz de possuir:

- Geração em tempo real de componentes;
- Simulação complexa da dinâmica do mundo real;
- Geometria complexa e animação de imagens;
- Modelagem física esquelética e cinemática inversa;
- Compatibilidade com jogos do PS1;

Para alcançar esses objetivos a Sony fez seu sistema semelhante a uma fábrica que agrupa seus funcionários em equipes de acordo com o seus talentos, desse modo separando o trabalho de uma forma mais inteligente, de maneira que cada equipe desenvolva o trabalho que sabe fazer melhor, obtendo uma melhor produtividade. Entretanto isso acabaria trazendo certa dor de cabeça para o gestor da fábrica, ou para o desenvolvedor no caso do PS2, devido ao fato deste ser responsável por distribuir de forma eficiente as tarefas para cada equipe.

Introdução ao funcionamento de um PlayStation 2

Para uma melhor compreensão do funcionamento interno do PS2, não basta apenas olhar para a sua CPU, mais se deve levar em consideração todos os seus componentes internos.

ARQUITETURA DO PS2

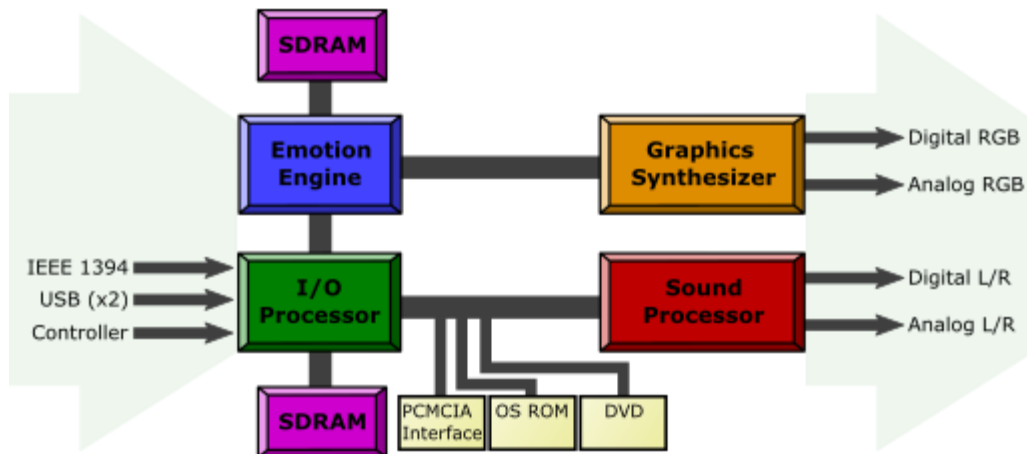


Figura 1: Arquitetura interna de um Playstation 2 (PS2).

Como mostrado na figura 1, o PS2 é constituído por quatro partes principais, são elas:

1. Emotion Engine;
2. I/O Processor;
3. Graphics Synthesizer;
4. Sound Processor;

O Emotion Engine é a CPU do console, considerado o coração do PS2, este chip foi desenvolvido pela Sony e Toshiba e é o maior responsável pela maioria dos cálculos processados, este chip têm como funções bases:

- Efetuar cálculos 3D;
- Cálculos geométricos, como transformações, translações, etc;
- Controle de programas;
- Cálculos de vetorização;

- Comportamento e Simulação do mundo, como a inteligência artificial, cálculos de colisões, cálculos de ambientes físicos em permanente transformação etc...

Compreendendo

O I/O Processor, também chamado de IOP, é responsável pelo controle das portas USB, Firewire e os comandos usados para jogar e navegar nos menus. Um exemplo de sua funcionalidade é quando estamos jogando e apertamos algum botão do controle, então o IOP processa esse input e o envia para o Emotion Engine, que fica responsável por atualizar o jogo de acordo com a entrada obtida.

Logo após o Emotion Engine processar todos os dados que lhe foram enviados é gerado um *display list* (lista de renderização), que é entregue ao Graphics Synthesizer renderizar na tela para o jogador. Este assemelha-se as GPUs dos computadores, também chamadas de placas aceleradoras gráficas. Finalmente o Sound Processor faz a sonorização do jogo, suportando som digitais 3D.

Enquanto as informações são processadas, cada componente pode estar fazendo um acesso as memórias SDRAM para buscar dados ou instruções a serem executadas.

O que é uma CPU

A Unidade Central de Processamento, também chamada de CPU, é um circuito integrado responsável por controlar todas as operações e o funcionamento do computador, fazendo execução de cálculos, decisões lógicas e instruções que resultam em todas as tarefas que um sistema computacional pode fazer. Ela é responsável por interpretar e retornar um resultado das instruções fornecidas pelo software, como programas, jogos etc. A CPU é considerado o cérebro do sistema devido a uma comparação feita com nós seres humanos, já que usamos o cérebro

para fazer cálculos matemáticos e o computador usa a CPU para realizar e processar esses cálculos.

A velocidade de uma CPU é determinada de acordo com a quantidade de instruções que ela pode executar por segundo, a essa velocidade se dá o nome de clock e utiliza-se a medida Hertz (Hz) para calculá-la, onde 1Hz equivale a uma instrução por segundo.

Apesar de um computador e um console possuírem uma CPU, as arquiteturas deles são bastante distintas, apenas sendo semelhantes no seu propósito principal que é efetuar cálculos e decisões.

Qual é a CPU do PS2?

O processador do PS2 é o MIPS R5900, seu nome menos técnico e adotado pela Sony é Emotion Engine (EE), ele foi desenvolvido pela Sony em parceria com a Toshiba, entretanto outros dispositivos também fazem uso dos processadores MIPS, como por exemplo: Nintendo 64, PSOne, PSP, WebTV da Sony, impressoras, copiadoras, scanners entre outros diversos dispositivos eletrônicos.

O que tornava o EE diferente dos demais processadores MIPS é que ele executava instruções de multimídia com operações de até 128 bits.

Arquitetura do Emotion Engine (EE)

O Emotion Engine é uma combinação de CPU com um processador DSP (Digital Signal Processing). O DSP tem como principal função processar enormes quantidades de dados de input, realizar cálculos sobre esses dados e produzir uma

grande quantidade de dados de output, por isso torna-se necessário uma boa velocidade e largura de bus. As especificações técnicas do EE são as seguintes:

- Processador de 128 bit;
- Velocidade de clock do processador: 300 MHz;
- Co-processador Floating point unit (FPU) operando a 6,2 gigaflops;
- Velocidade do barramento: 3,2 GB por segundo;
- Núcleo original da CPU do Play Station como processador I/O;

ARQUITETURA DO EMOTION ENGINE

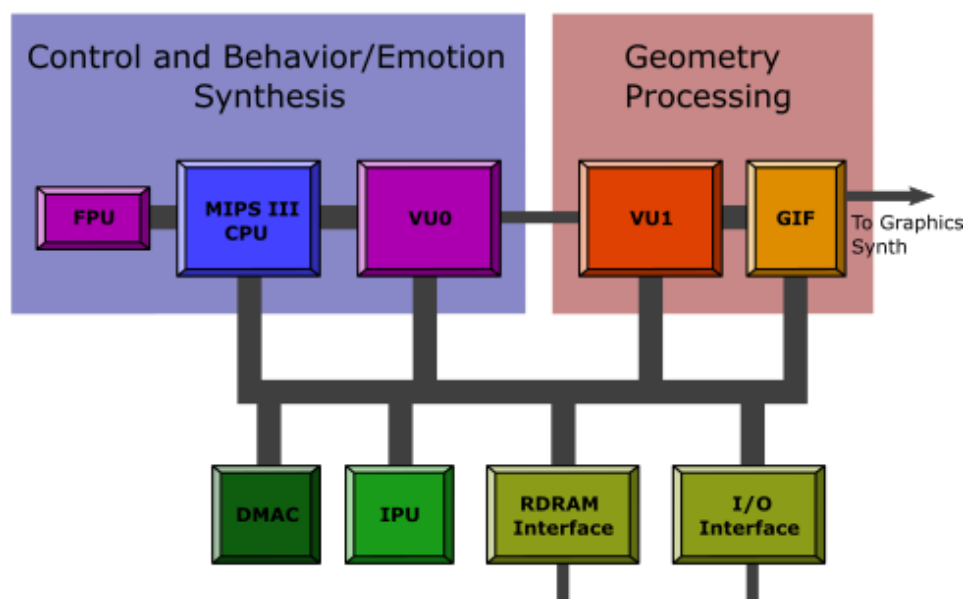


Figura 2: Arquitetura interna do Emotion Engine (EE).

Cada um dos componentes do EE tem uma tarefa específica, podendo variar de acordo com as necessidades do desenvolvedor, basicamente elas são divididas das seguintes formas:

- CPU + FPU: Controla os programas;
- CPU + FPU + VU0: Processamento de comportamento e emoções, cálculos físicos, calculos de colisões, etc;
- VU1: Calculos de geometria simples que produzem *display lists* que são enviadas diretamente para o Graphics Synthesizer (via GIF);
- IPU: Descompressão de imagem;
- DMAC e RDRAM Interface: Fazem acessos as memorias fora do chip;
- GIF: Canal DMA que estabelece comunicação entre o EE e os co-processadores GS;
- I/O Interface: Faz comunicação com os dispositivos de entrada e saída.

Apesar de cada componente ter o seu devido papel na arquitetura, eles podem estar efetuando outros tipos de cálculos, como por exemplo: todo o conjunto da FPU + CPU + VU0 podem estar realizando os cálculos de geometria simples, gerar um display list e mandar ele direto a o GIF, para que este mande essas informações para o processador gráfico, ou mandar o resultado desses cálculos para a VU1, para que ela gere o *display list* e o mande direto para o GS. Outro exemplo: o grupo da FPU + CPU + VU0 pode funcionar como um pre-processador para a VU1, calculando a IA e fisica, gerando uma informação que a VU1 aceita como input e coloca-la numa *display list*. Toda essa flexibilidade do sistema ajuda o desenvolvedor a criar ambientes 3D de acordo com as suas necessidades, porém por outro lado aumentando a complexidade da programação.

Dentro do EE existe um barramento que conecta a VU0 e a FPU diretamente a CPU, dessa forma quando surge a necessidade de troca de informação, não é necessário fazer uso do barramento do sistema, evitando um possível efeito gargalo. Esse barramento tem um tamanho de 128 bits e permite que o MIPS III acesse os registradores da VU0 fazendo com que este trabalhe como um coprocessador.

A CPU e a VU0 compartilham juntas uma SPRAM (Scratch Pad RAM) de 16K, que pode ser usada para guardar estruturas de dados a serem enviadas para outros componentes, essas transferências de dados são controladas por um software.

A VU1 também é ligada a GS através de um barramento de 128 bits, para caso o programador sinta a necessidade de enviar informações diretamente para o GS sem antes ter passado pelo GIF.

Unidades vetoriais

O PS2 é constituído por duas unidades vetoriais, a VU0 e VU1, que possuem arquiteturas idênticas apesar de suas funcionalidades serem distintas e de cada uma possuir suas próprias instruções e memória de dados. Elas são responsáveis por realizar cálculos vetoriais e matriciais. Cada VU utiliza-se de um vetor constituído por quatro componentes que contém os campos (X, Y, Z e W) como operadores, cada um desses campos guardam um número em virgula flutuante de 32 bits de precisão simples devido ao barramento do EE ser de 128 bits.

As VUs não podem acessar diretamente as memórias principais, dessa forma é necessário que os dados estejam guardados em suas próprias memórias VU, antes de serem processados. Isso acaba se tornando um grande desafio para os programadores, transferir informações entre as várias unidades do EE até as VUs de uma forma eficiente.

As VUs podem trabalhar de duas formas, que são elas: o modo macro e o modo micro. No modo micro as VUs executam as instruções dos seus próprios registradores, tanto a VU0 e a VU1 podem trabalhar nesse modo, enquanto no modo macro a VU funciona como um coprocessador para o MIPS III, nesse caso somente a VU0 pode trabalhar nesse modo.

VU0

A principal função da VU0 é a de estar atuando como um coprocessador para o MIPS III, pode também operar de forma independente ou no modo VLIW. Quando ela trabalha como um coprocessador para o MIPS III, a VU0 só leva instruções de 32 bits, no modo VLIW essas instruções podem ser estendidas a 64 bits.

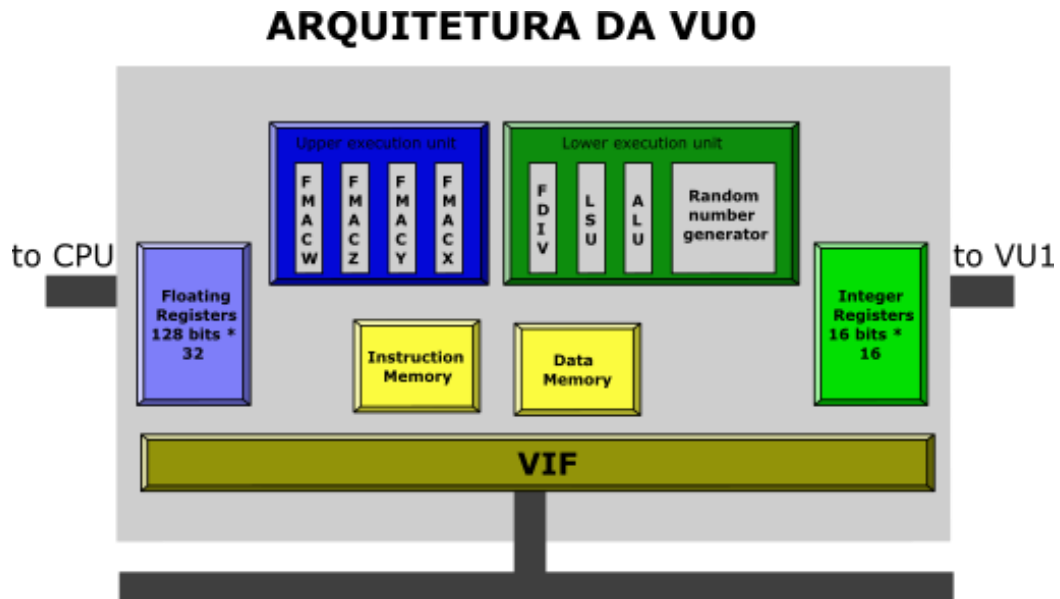


Figura 3: Arquitetura interna da VU0.

Características da VU0:

- Velocidade de 150MHz;
- 32 Registradores de ponto flutuante, com 128 bits;
- 16 Registradores de números inteiros, com 16 bits;
- 4 FMACS usados para fazer cálculo de acumulação, multiplicação e ponto flutuante;
- VIF usado para fazer acesso aos dados da VU0 quando ela está operando, devido a estes dados só poderem ser acessados quando a mesma está parada ou funcionando no modo macro;

Um exemplo de sua utilização foi no jogo Dungeon, que através de códigos em C / C++ foi feito com que a VU0 opere no modo macro, onde lhe foi dado um conjunto de instruções limitadas para acelerar parte de um *pipeline* de renderização de software do EE, além desses códigos, ela foi usada também para fazer cálculos matemáticos de matrizes e vetores, acelerando cálculos como: matriz * multiplicação de vetores e projetos.

Apesar da VU0 ser bastante poderosa, podendo até mesmo realizar cálculos de iluminação, os programadores não enfatizaram muito o seu uso, devido ser consideravelmente difícil programar um bom software que a comunique da melhor maneira possível com o MIPS III, já que as VUs só executam instruções da sua própria memória.

VU1

A VU1 é totalmente independente, contendo todas as funções da VU0 além de algumas outras. A VU1 lida com processamento de geometria, possui uma memória de instrução de 16k bytes e uma memória de dados de 16k bytes, enquanto a VU0 apenas 4k bytes cada. Essa grande diferença se deve ao fato da VU1 carregar mais instruções e dados a serem calculados. Logo após esses dados serem calculados é gerado um display list que por sua vez é mandado para o GS renderizar as informações na tela para o usuário no ponto certo.

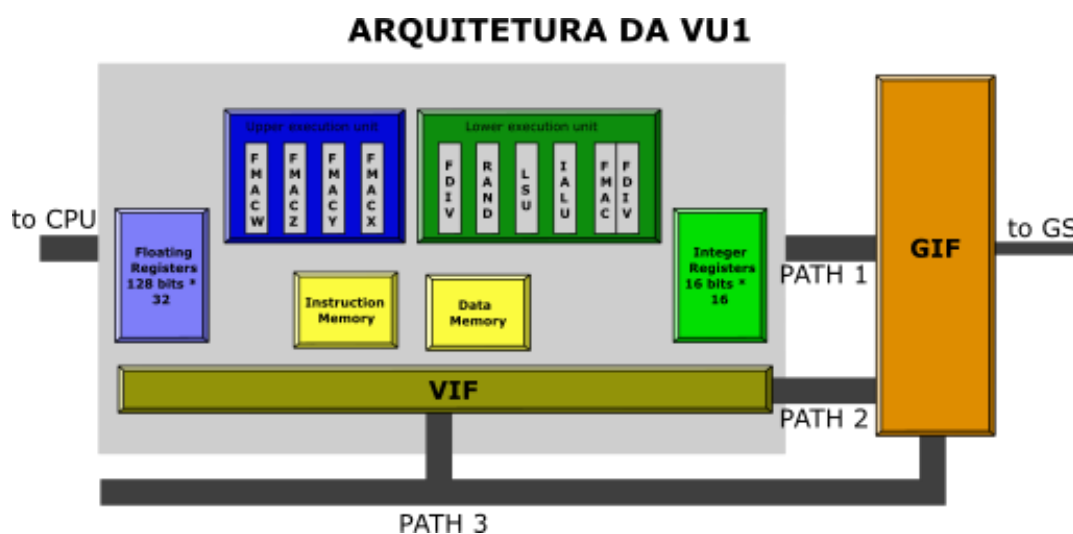


Figura 4: Arquitetura interna da VU1.

Características da VU1:

- Velocidade de 150MHz;
- 32 Registradores de ponto flutuante, com 128 bits;
- 16 Registradores de números inteiros, com 16 bits;
- 4 FMACS usados para fazer cálculo de acumulação, multiplicação e ponto flutuante;
- 3 Barramentos por onde os dados podem ser trafegados;

Esses três barramentos evitam o efeito gargalo, dificultando de ocorre-lo na VU1 devido as várias possibilidades de caminhos para tráfego de dados. Os usos desses caminhos são de fácil programação, tanto que programadores focam mais na utilização da VU1 do que na VU0. Abaixo estão listadas essas três possibilidades:

- 1º: Transmitir a lista de exibição através do barramento principal;
- 2º: Pode fazer uso do barramento que existe entre a VIF e GIF;
- 3º: O caminho sai da unidade de execução mais baixa e vai direto para o GIF.

O VIF da VU1 trabalha muito mais do que o da VU0, pois ele leva o que a Sony chama de display list 3D. Essa lista constrói dois tipos de dados: as instruções de programação VU1 (que vai para a Instruction Memory) e os dados de instruções já lidos (que vai para a Data Memory). As instruções podem ser divididas em: instrução superior e instrução inferior, que operam em unidade de execução superior e unidade de execução inferior respectivamente.

A VU1 possui a adição de um EFU que é constituído por uma FMAC em FDVI, que serve para fazer cálculos básicos necessários para cálculos geométricos mais complexos.

Pipeline do PS2

A pipeline do PlayStation 2 foi baseada na pipeline do OpenGL, que é bastante diferente do que é apresentado atualmente. Na época que o PS2 foi lançado a pipeline do OpenGL era constituído pela: aplicação, geometria, iluminação, rasterização, texturização e buffer de imagens. Não são todos os jogos que seguem à risca essa sequência de instruções, devido a possuírem características distintas, exemplo: jogos sem efeito de iluminação, sombras etc...

Essa pipeline pode variar de duas formas distintas de acordo com o funcionamento da VU0, como mostra as figuras a seguir:

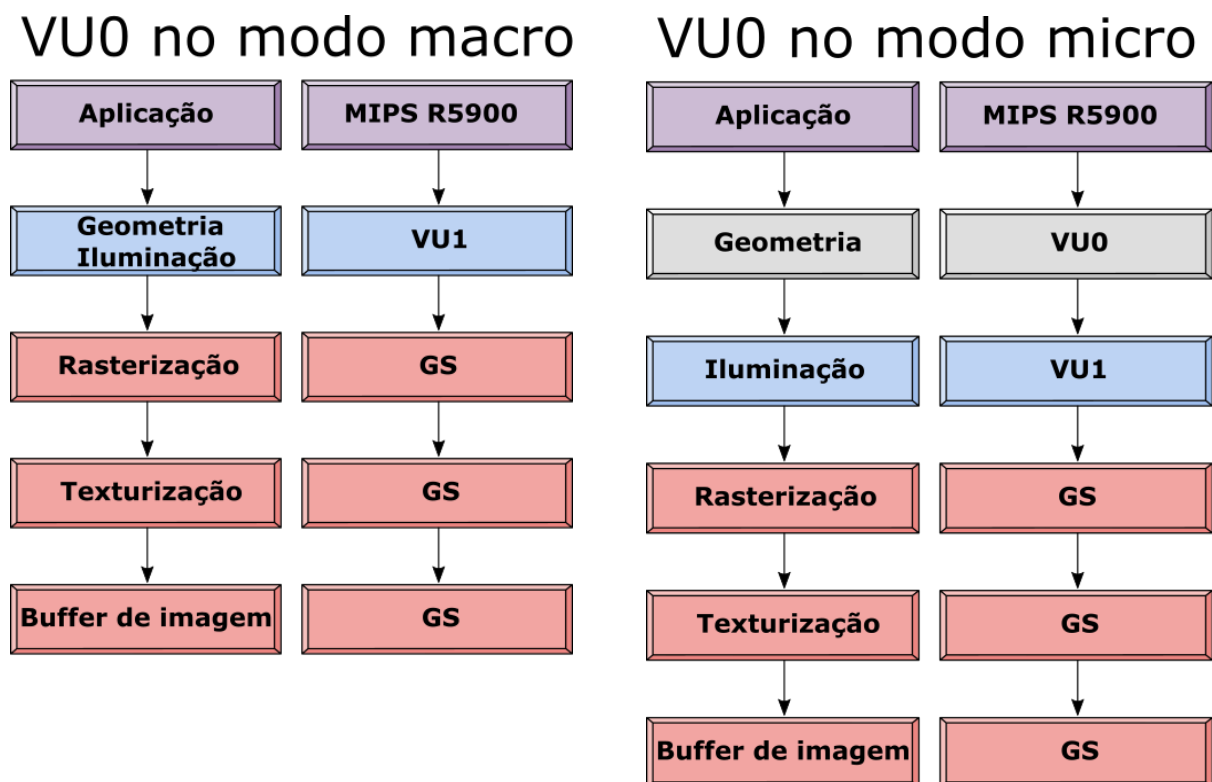


Figura 5: Pipeline nos modos macro e micro da VU0.

Obs. Rasterização é o processo de transformar dados geométricos e pixels em fragmentos.

O que é uma GPU

A GPU é um processador normalmente encontrado nas placas de vídeos, tendo como principal função a renderização em tempo real dos gráficos. Normalmente esse ou esses processadores ficam localizados na parte central das placas de vídeos, com os demais componentes ao seu redor. Devido a sua alta quantidade de calor produzido, os fabricantes incluem junto da placa um dissipador de calor e ventoinha para mantê-lo refrigerado.

Enquanto a CPU possui de quatro a oito núcleos, uma GPU é composta por centenas de núcleos menores que podem trabalhar em paralelismo, justamente para cumprir as grandes quantidades de dados que lhe são enviadas mantendo uma alta performance. Para um bom desempenho, uma GPU necessita de uma CPU para auxiliá-la nos processos.

Devido a sua alta performance atualmente o uso dessas placas não se limitam mais em apenas processar imagens, estão sendo utilizadas em várias áreas: tecnológica, financeira, medicina etc...

Núcleos de uma CPU e GPU

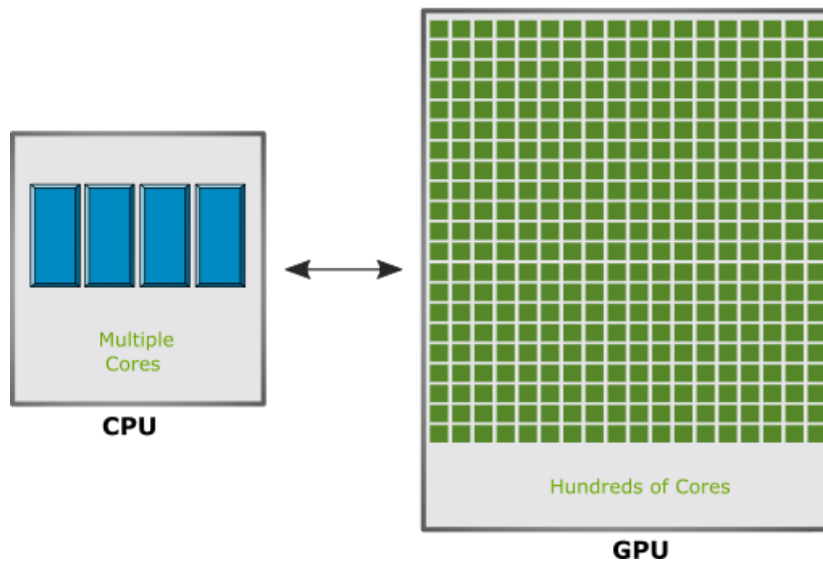


Figura 6: Comparação entre núcleos de uma CPU e GPU.

Qual é a GPU do PS2?

Graphics Synthesizer, também chamada de GS, é a GPU do PS2. Possui como principal função renderizar na tela as informações previamente processadas pelo Emotion Engine. Muitos consideravam a GS como o ponto fraco do PlayStation 2, devido a sua baixa capacidade de processamento e pouca memória interna. Entretanto com uma análise mais aprofundada, concluiu-se que a maioria dos cálculos eram realizados pelo processador principal, com a GS restringindo-se apenas a exibir as informações na tela para o jogador.

A GS suporta recursos básicos de: rasterização de polígonos, texturização, mistura de alfa, scissoring básico, zbuffer etc...

Arquitetura do Graphics Synthesizer (GS)

Especificações técnicas:

- Velocidade 150 MHz;

- Armazenamento auxiliar integrado embutido;
- 4 MB VRAM;
- Resolução: 640x480 ou 320x240 entrelaçado;
- Resolução de polígonos: 75 milhões de polígonos por segundo.

ARQUITETURA DO GRAPHICS SYNTHESIZER

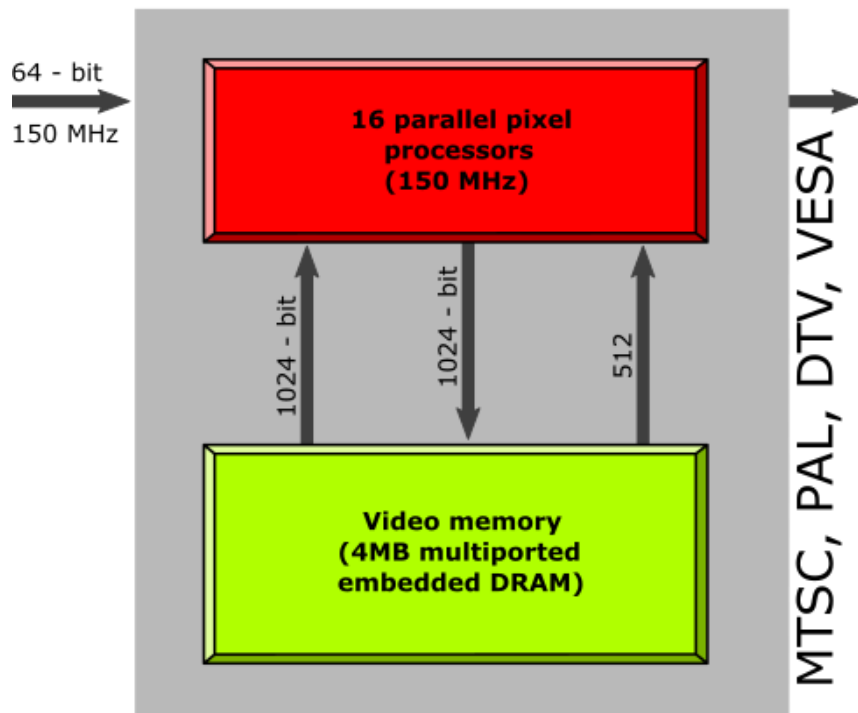


Figura 7: Arquitetura do Graphics Synthesizer (GS).

Quando alguma cena irá ser renderizada na tela para o usuário, ela é quebrada em vários subprocessos, então cada um dos dezesseis processadores fica responsável por processar partes das imagens, ao final de tudo cada parte da imagem é juntada em uma e exibida na tela para o usuário. Essa informação vem na forma de um display list, que é uma sequência de comandos que indica ao GS as ordens de espaço e formas, ou seja, indica o que deve ser desenhado e em qual lugar da tela.

Memória do Graphics Synthesizer (GS)

A memória da GS possui uma capacidade de 4 MB e é do tipo DRAM, esse tipo de memória precisa ser refrescada a cada clock devido a sua perda de elétrons. Ela é dividida em 4 partes, cada uma delas de 1 MB como mostra a figura a seguir:

Memória da GS

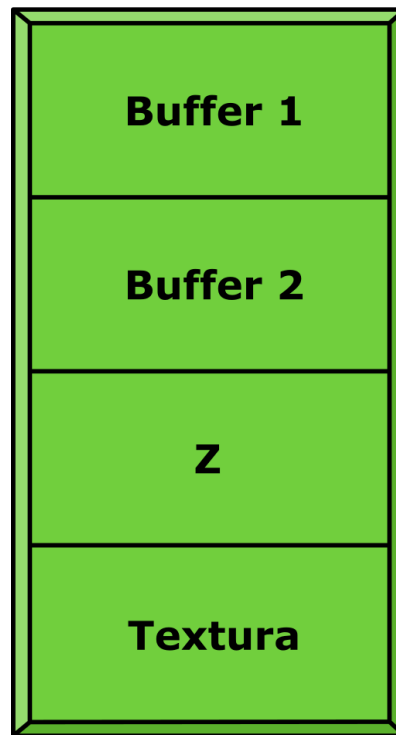


Figura 8: Memória do Graphics Synthesizer (GS).

- Buffer1: Armazena os quadros já processados;
- Buffer2: Armazena o próximo quadro a ser processado depois do Buffer1;
- Z: Armazena a profundidade do objeto;
- Textura: Armazena a textura do cenário, se necessário pode armazenar também a informação do Buffer1.

Apesar de 4 MB de VRAM não parecer muito, devemos levar em consideração que a GS possui uma velocidade de 150 MHz e seu barramento é de 128 bits de largura. Em teoria ela é capaz de transferir 76 MB de frames em trinta quadros por segundo, que é mais que o dobro da capacidade da memória da EE.

As informações contidas no Buffer 2 são preenchidas pelo sistema, logo após o Buffer 1 já ter sido devidamente renderizado na tela, o Buffer 2 se torna Buffer 1 e o sistema preenche novamente o Buffer 2 com o próximo frame. Entretanto antes de carregar a textura e renderizar a informação na tela para o usuário, a GS analisa qual é a distância do objeto em relação a câmera do jogador, para determinar o nível de qualidade da textura: alta, média ou baixa. Quanto mais longe da visão do jogador menos detalhado será o objeto.

Demonstração 3D

Em apresentação do PS2 para o público, a Sony como maneira de exibir o grande potencial de seu console, mostrou parte do rosto de um homem em idade avançada, que apresentava 76.857 polígonos por frame, como mostra a figura a seguir:

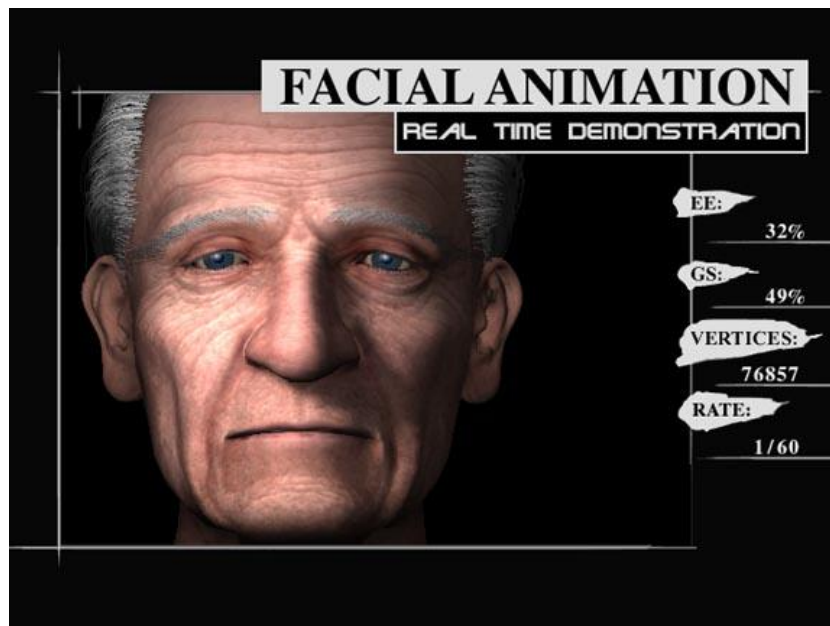


Figura 9: Animação facial, demonstração em tempo real.

O pipeline de gráficos 3D

A imagem a seguir ilustra o pipeline de gráficos 3D:

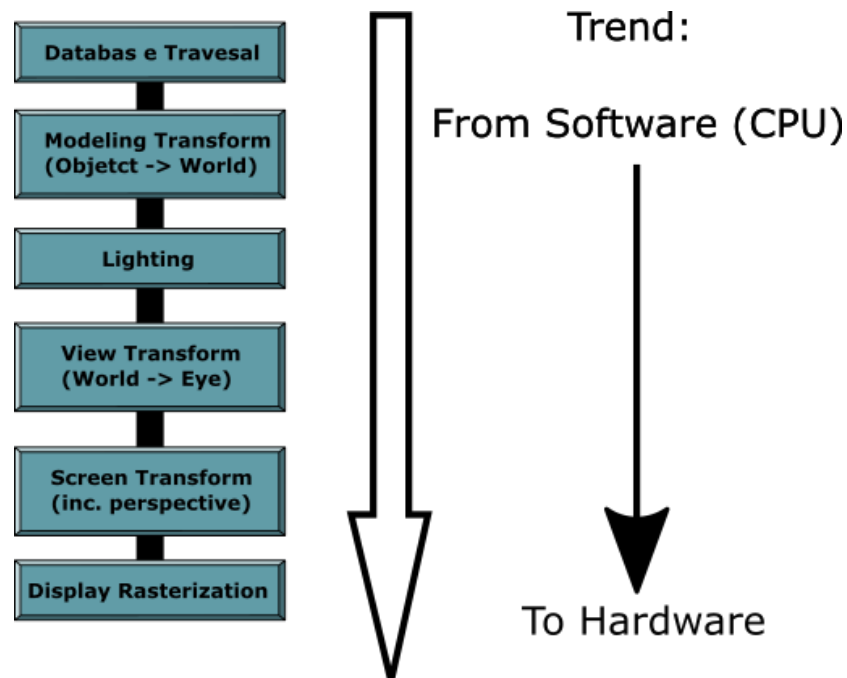


Figura 10: Pipeline de gráficos 3D.

Nos primeiros dias de testes, a tela era mapeada numa memória, a CPU escrevia os pixels diretamente no buffer de quadros e todo o pipeline era de responsabilidade da CPU. Com o passar do tempo transferiram partes desse pipeline, agora são de responsabilidade de um hardware mais especializado nesses tipos de cálculos, capazes de tomar geometricamente descrições em termos de vértices, realizar todas as transformações, iluminação, texturização e rasterização de hardware.

Compreendendo a renderização 3D

As renderizações 3D são baseadas em pequenos triângulos que juntos dão forma aos objetos.

Num processo de renderização de terreno são utilizadas funções de ruído, assim como todas as texturas. Em um exemplo, a CPU primeiramente calcula uma prévia da visualização e elabora uma amostragem baseada no tipo de terreno que está dentro do campo de visão do jogador, que futuramente pode ser adaptada, então a CPU usa uma métrica que tenta manter o tamanho dos triângulos renderizados o mesmo, tanto para aqueles que estão próximo e longe da tela. Também procura evitar que partes não visíveis do terreno sejam calculadas. A CPU transmite esses parâmetros de amostragem do terreno para os co-processadores VU0 e VU1 que executam vários micro-processos, com o objetivo de acelerar vários cálculos, finalmente executando a iluminação e renderização. A VU0 é usada para a renderização rápida de terrenos usando funções de ruídos, enquanto a VU1 fica responsável pela construção das tiras de triângulos. A importância de se construir as tiras triangulares através de algoritmos é que se torna mais fácil saber o compartilhamento dos vértices, resultando em 46% menos vértices, facilitando os cálculos de iluminação. Logo depois dos cálculos, as informações são passadas para a GS exibir na tela o resultado para o jogador.

Quando se vai renderizar partículas, ao invés de se usar triângulos são usados pontos, que são eles próprios uma simulação de partículas. Para simular uma movimentação de partícula a CPU juntamente com a VU0 realizam cálculos se baseando em uma simulação de partícula, que são emitidas ao longo de um vértice do modelo, logo após os cálculos, as informações são enviadas a VU1 para serem transformadas e processadas, gerando um display list que é enviado para a GS renderizar.

Memórias do PS2

O PS2 é composto por uma main memory do tipo RDRAM, essa memória é de acesso aleatório, que perde os dados após perda de energia, nela pode ser lido ou escrito informações.

O PS2 também acompanha um memory card, usado para guardar o progresso do jogador, esse tipo de memória não perde informação ao ser desenergizada, e nela

pode ser escrito ou lido informações salvas, essa memória é de acesso sequencial, devido a cada bloquinho da memória ser salvo a informação de um único jogo.

Também existem memórias dentro dos chips de processamento, como na VU0 e VU1, onde são usadas para guardar e ler instruções. Essas memórias perdem os dados se houver queda de energia, elas são memórias do tipo aleatório.

Dentro da GPU se encontra uma memória do tipo DRAM, essa memória também perde informação ao ser desligada. É usada para salvar informações já processadas pelo sistema, e para se ler as próximas instruções a serem executadas. Também é uma memória do tipo aleatório.

Resultados e discussão

O PS2 pode ser considerado um sistema que trabalha em paralelismo, dividindo o trabalho da CPU entre coprocessadores de uma forma mais inteligente, de maneira que cada componente desenvolva suas tarefas com mais eficiência sem sobrecarregar o seu uso. Com isso obtendo um melhor desempenho comparado a seus concorrentes na época. O uso de um bus com boa largura e alta velocidade é fundamental neste sistema devido ao grande número de dados trafegados simultaneamente.

Todas essas características foram essenciais para que os desenvolvedores de jogos explorassem ao máximo o potencial do console mantendo um alto desempenho, o consolidando como o vídeo game mais vendido da história.