



# BATALHA DE BITS



CONQUISTE O MELHOR  
COMPUTADOR!

REGRAS DO

JOGO

START



# MANUAL GERAL

## 1. 💡 Visão Geral e Objetivo

O **Batalha de Bits** é um jogo estratégico onde cada jogador assume o papel de um engenheiro de hardware com um projeto específico em mente. O objetivo principal é ser o primeiro a atingir a distribuição ideal de Capacitores (recursos) de acordo com sua Carta de Objetivo.

## 2. 🧩 Componentes Chave

- **Tabuleiro (Placa-Mãe):** Mapa dividido em áreas funcionais (CPU, Cache, RAM, Armazenamento, E/S, Barramento).
- **Tokens de Capacitor:** Recursos de otimização de hardware.
- **Cartas de Pergunta:** Usadas para validar o conhecimento e progredir na alocação de Capacitores.
- **Cartas de Objetivo:** Define o tipo de computador que o jogador deve construir, e qual a distribuição necessária de recursos.
- **Dados:** Usados para resolver definir a quantidade de capacitores a serem distribuídos por rodada.

## 3. 🎯 Condição de Vitória

O jogo termina e o jogador é declarado vencedor quando ele cumpre a distribuição necessária de acordo com sua Carta de Objetivo.

## 4. 🎮 Regras de Jogo e Fluxo da Rodada

### 4.1. Configuração Inicial

1. Cada jogador recebe uma Carta de Objetivo Secreta.
2. Cada jogador recebe 5 Tokens de Capacitor a serem distribuídos inicialmente pela Placa-Mãe, baseando-se no seu objetivo.

## 4.2. Fluxo da Rodada

Na sua vez, o jogador segue os passos:

**Declaração de Alocação:** O jogador joga o dado, e por ele é definido quantos capacitores serão alocados naquela rodada

1. Então ele anuncia em qual área do Tabuleiro deseja alocar os Capacitores.

### 2. Verificação de Espaço:

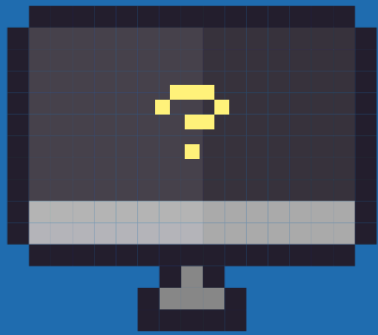
- **Caso haja espaço disponível no componente:** O jogador retira uma Carta de Pergunta.
  - Se responder corretamente dentro do tempo 30 segundos, ele aloca os capacitores na área determinada..
  - Se responder incorretamente ou o tempo se esgotar, ele perde o Capacitor e a vez.
- **Caso o espaço esteja lotado:**
  - Se o espaço está cheio, mas ocupado por ele mesmo, ele não pode alocar mais.
  - Se o espaço está cheio e ocupado por um rival, é declarada uma Batalha.

## 4.3. ⚔️ Mecânica de Batalha

A batalha ocorre quando um jogador (Atacante) tenta otimizar um componente que já atingiu seu limite máximo de Capacitores e está sob o domínio de outro jogador (Defensor).

1. Batalha de Respostas: Atacante e Defensor retiram a mesma Carta de Pergunta e o que responder primeiro corretamente ganha.
2. Resolução por Dados:
  - Se ambos errarem: A batalha é resolvida unicamente pela rolagem de dados.

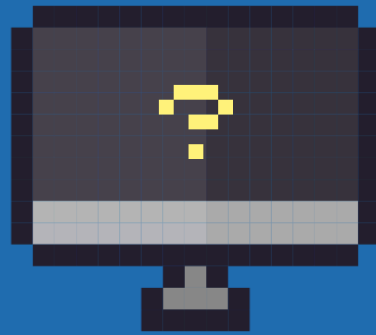
O vencedor da batalha (maior resultado no dado ajustado) conquista o espaço. O Capacitor do perdedor é retirado daquela área, simulando a destruição/superação do componente do rival.



Sou a lei que previu que o número de transistores em um chip dobraria aproximadamente a cada 18 meses.

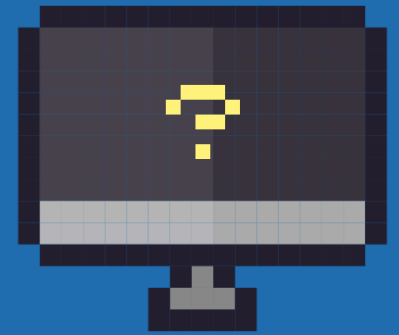
Quem sou eu?

R: Lei de Moore.



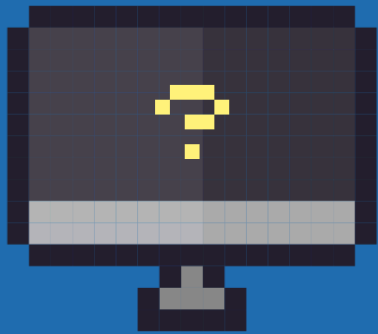
No modelo de Von Neumann, quais são os quatro componentes fundamentais de um computador?

R: CPU, Memória Principal, Entradas/Saídas (E/S) e Barramento.



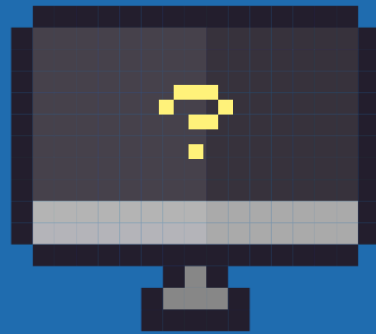
Sou o "cérebro" do computador, responsável por buscar, decodificar e executar instruções. Quem sou eu?

R: CPU (Unidade Central de Processamento) ou Processador.



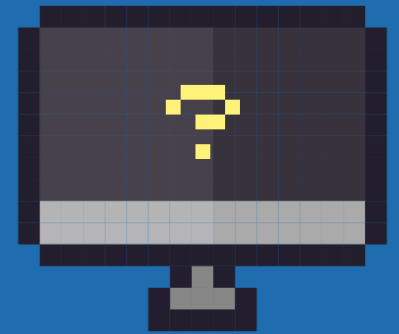
Sou um tipo de memória volátil. Se a energia acabar, eu esqueço tudo. Sou usada para armazenar programas em execução. Quem sou eu?

R: Memória RAM.



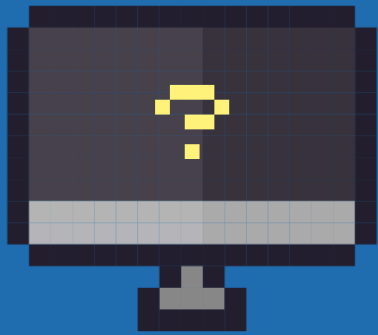
O que significa a sigla MIPS no contexto desta arquitetura?

R: Microprocessor without Interlocked Pipeline Stages (Microprocessador sem estágios intertravados de pipeline).



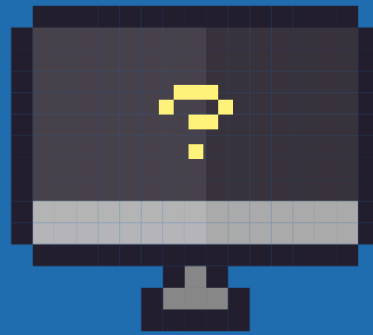
Sou um registrador especial que aponta para a *próxima* instrução que será executada. Quem sou eu?

R: PC (Program Counter).



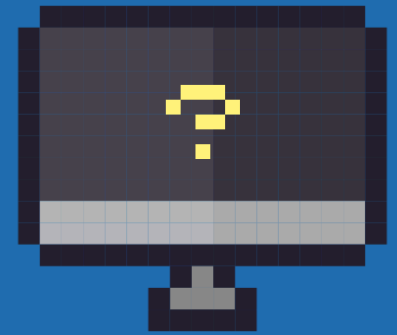
Qual a diferença fundamental entre arquiteturas RISC e CISC?

R: RISC tem instruções reduzidas e simples (execução rápida); CISC tem instruções complexas.



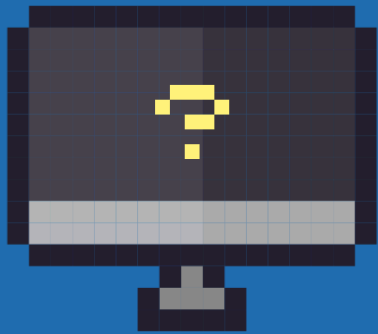
Faço as contas do computador. Somas, subtrações e operações lógicas (AND, OR) são comigo mesma. Quem sou eu?

R: ULA (Unidade Lógica e Aritmética).



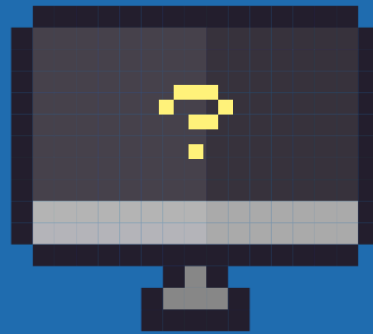
Sou um registrador do MIPS. Não importa o que você tente gravar em mim, meu valor será sempre 0. Quem sou eu?

R: Registrador \$zero (ou \$0).



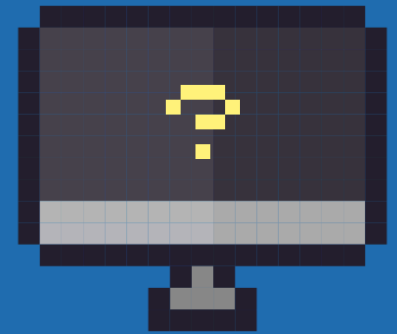
No MIPS, existem 3 formatos principais de instrução. O formato "R" é usado para operações aritméticas. Para que servem os formatos "I" e "J"?

R: "I" para Imediato (constantes) e transferências de dados; "J" para Desvios incondicionais (Jumps).



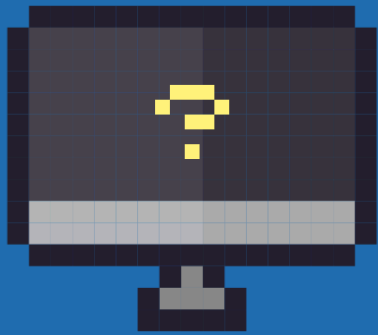
Sou um modo de endereçamento onde o operando (o valor) já está escrito dentro da própria instrução. Não preciso buscar na memória. Quem sou eu?

R: Endereçamento Imediato.



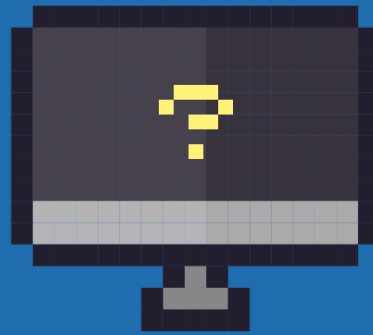
Para chamar um procedimento no MIPS usamos jal. Qual registrador é usado para salvar o endereço de retorno para que o programa saiba voltar depois?

R: Registrador \$ra (Return Address / \$31).



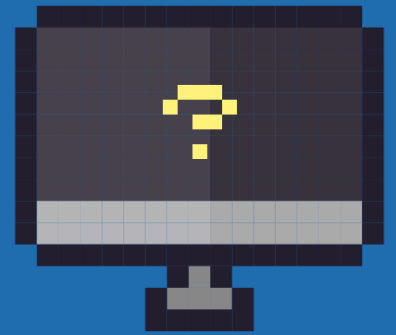
Sou a convenção que diz que o byte menos significativo fica no menor endereço de memória. O contrário do meu irmão "Big". Quem sou eu?

R: Little-Endian.



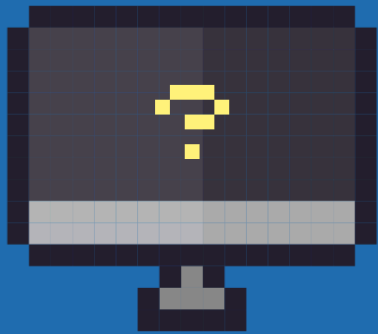
O que faz a instrução lw \$s1, 100(\$s2)?

R: Carrega uma palavra (Load Word) da memória para o registrador \$s1, usando o endereço base em \$s2 somado ao deslocamento 100.



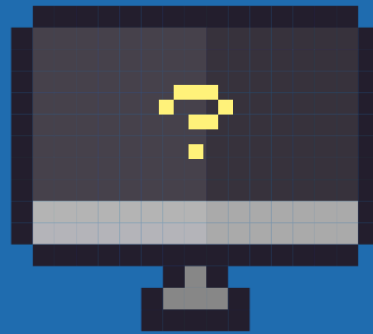
Sou usado para armazenar variáveis locais e endereços de retorno quando as chamadas de função são aninhadas. Cresço de cima para baixo na memória. Quem sou eu?

R: A Pilha (Stack) ou Stack Pointer (\$sp).



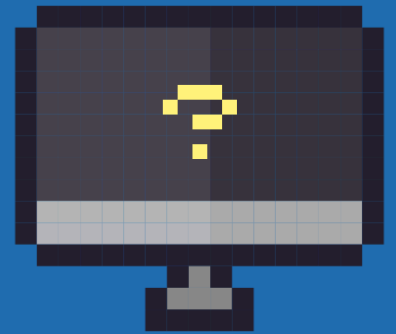
Quantos registradores de propósito geral de 32 bits a arquitetura MIPS possui?

R: 32 registradores.



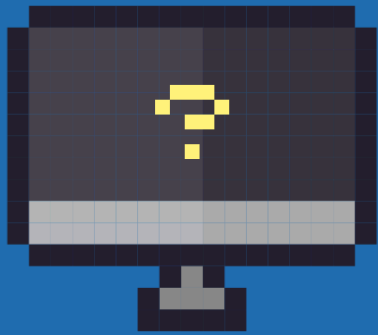
Sou uma técnica que permite executar múltiplas instruções simultaneamente, dividindo-as em estágios como numa linha de montagem de carros. Quem sou eu?

R: Pipeline.



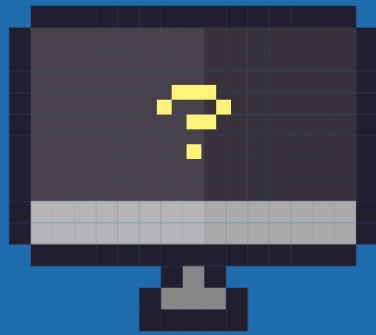
No Pipeline, o que é um "Conflito de Dados" (Data Hazard)?

R: Quando uma instrução precisa de um dado que ainda não foi escrito pela instrução anterior.



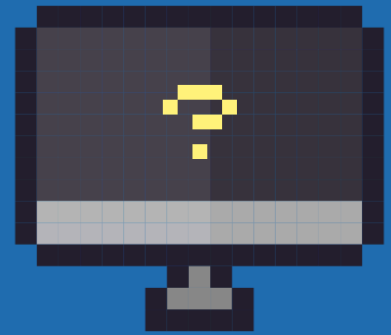
Sou uma solução para conflitos de dados. Eu pego o dado da ULA e passo direto para a próxima instrução sem esperar escrever no registrador. Quem sou eu?

R: Forwarding (ou Adiantamento).



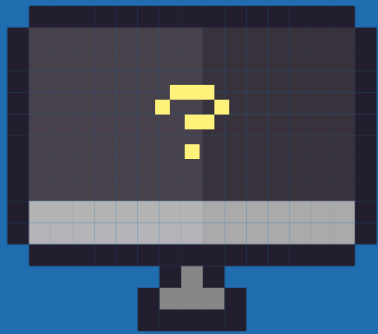
O que acontece num "Conflito de Controle" (Control Hazard) em um pipeline?

R: Ocorre em desvios (branches). O processador não sabe qual será a próxima instrução a buscar até que o desvio seja resolvido, podendo causar a inserção de "bolhas" (stalls).



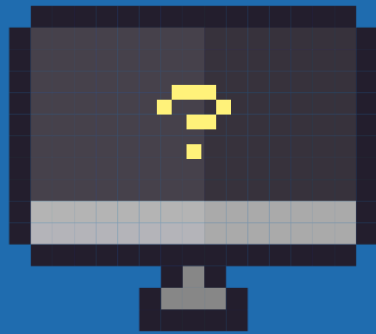
Sou o princípio que diz: "Se você usou um dado agora, provavelmente vai usá-lo de novo em breve". As caches me amam. Quem sou eu?

R: Localidade Temporal.



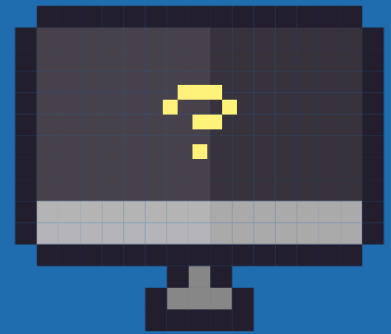
O que é um "Cache Miss"?

R: É quando o processador procura um dado na memória Cache e não o encontra, tendo que buscá-lo na memória principal (que é mais lenta).



Sou um tipo de mapeamento de cache onde cada bloco da memória principal tem um lugar fixo e único para ficar na cache. Sou simples, mas posso gerar conflitos. Quem sou eu?

R: Mapeamento Direto.



Na hierarquia de memória, quem é mais rápido: a Memória Cache L1 ou os Registradores?

R: Os Registradores (são o topo da pirâmide, a memória mais rápida e interna da CPU).

 <h1>MISSÃO</h1>	 <h1>MISSÃO</h1>	 <h1>MISSÃO</h1>
<h3>O PC Gamer Supremo</h3> <p>PCs voltados para jogos exigem alto processamento gráfico e uma boa CPU para não gerar gargalos.</p> <p><b>Condição de Vitória:</b> Conquistar totalmente a GPU (ocupar os 8 espaços) e possuir a maioria absoluta da CPU (pelo menos 4 dos 6 espaços).</p>	<h3>O Servidor de Banco de Dados</h3> <p>Servidores priorizam imensa capacidade de memória volátil e armazenamento rápido para lidar com muitas requisições.</p> <p><b>Condição de Vitória:</b> Conquistar totalmente a RAM (8 espaços) e o SSD (4 espaços).</p>	<h3>A Máquina de Von Neumann Pura</h3> <p>Recriar o modelo clássico que define os computadores modernos: CPU, Memória e Entrada/Saída.</p> <p><b>Condição de Vitória:</b> Conquistar totalmente a Entrada/Saída (4 espaços), ter o controle da CPU (maioria dos espaços) e pelo menos metade da RAM (4 espaços).</p>
 <h1>MISSÃO</h1>	 <h1>MISSÃO</h1>	 <h1>MISSÃO</h1>
<h3>Mestre da Hierarquia de Memória</h3> <p>O jogador deve dominar o fluxo de dados, garantindo que a informação transite eficientemente do armazenamento lento ao rápido.</p> <p><b>Condição de Vitória:</b> Possuir pelo menos 2 capacitores no SSD, 3 na RAM e conquistar totalmente a Memória Cache (5 espaços).</p>	<h3>O Controlador do Sistema (Chipset)</h3> <p>O Chipset e o Barramento são o "coração" da comunicação. Quem controla o fluxo, controla o computador.</p> <p><b>Condição de Vitória:</b> Conquistar totalmente o Chipset (3 espaços) e ter presença (pelo menos 1 capacitor) em TODOS os outros componentes do tabuleiro.</p>	<h3>Arquitetura Balanceada (Anti-Gargalo)</h3> <p>Um sistema eficiente não tem um componente muito forte e outro muito fraco; ele é equilibrado para evitar o "estrangulamento" do desempenho.</p> <p><b>Condição de Vitória:</b> Ocupar 50% da capacidade de 4 componentes diferentes à sua escolha.</p>



# BATALHA DE BITS

