

Roteiro de Vídeo: Explicando o Emulador SAP-1

Este roteiro detalha a estrutura e o conteúdo para um vídeo de demonstração e explicação do funcionamento do emulador SAP-1. O objetivo é guiar o espectador através das funcionalidades do programa de forma clara e envolvente.

CENA 1: INTRODUÇÃO E GANCHO

- **Tempo:** 0:00 - 0:30
- **Visual:** Vinheta/introdução do canal/projeto. Em seguida, tela do emulador SAP-1 já aberta, mas ainda não em execução.
- **Áudio (Locutor):**

"Olá a todos! Já parou para pensar como um computador realmente funciona por dentro? Como ele processa as instruções que damos a ele? O que acontece no nível mais básico?"

"Neste vídeo, vou apresentar um projeto que desenvolvi para desmistificar isso: um emulador do microcomputador didático SAP-1, o 'Simple-As-Possible 1', baseado nos estudos de Albert Paul Malvino."

"Vamos explorar suas funcionalidades, entender sua arquitetura e ver uma demonstração prática em tempo real!"

CENA 2: O QUE É O SAP-1?

- **Tempo:** 0:30 - 1:30
- **Visual:**
 - Slide rápido/overlay explicando "SAP-1: Computador Didático de 8 bits".
 - Foco na área de visualização da CPU do emulador. Usar setas e destaques no vídeo para apontar cada componente enquanto o locutor fala.
 - [Imagem de SAP1 Architecture Diagram from Article, Fig. 101](#)
- **Áudio (Locutor):**

"O SAP-1 é um modelo de computador muito simples, criado para nos ajudar a compreender os fundamentos da arquitetura de computadores. Ele é um processador de 8 bits, com uma memória RAM de apenas 16 bytes e opera através de um único barramento de dados, o famoso Barramento W."

"No emulador, vocês verão os principais blocos funcionais do SAP-1. Temos o Contador de Programa, o Registrador de Endereço de Memória – o MAR –, a própria Memória RAM. Depois, o Registrador de Instruções, o Acumulador e o Registrador B para operações. A Unidade Lógica Aritmética, ou ULA, e o Registrador de Saída, com seus LEDs visuais."

"O conjunto de instruções é bem básico: LDA para carregar dados, ADD e SUB para operações, OUT para exibir resultados e HLT para parar o programa."

CENA 3: FUNCIONALIDADES E ENTRADA DE EXPRESSÃO (NOVO)

- **Tempo:** 1:30 - 3:30
- **Visual:**
 - Visão geral da janela do emulador, mostrando as três grandes seções: Entrada de Expressão/Editor, Controles, Visualização da CPU.
 - **Corte para a área de 'Entrada de Valores da Expressão':** Mostrar os botões numéricos e de operação.
 - **Ação:** Digitar uma expressão simples (ex: "5+3") usando os botões e clicar em "Entrar". Mostrar o código Assembly sendo gerado no editor principal.
 - **Corte para o Editor Assembly:** Destaque no código gerado.
 - **Corte para a área de Controles:** Destaque nos botões.
 - **Corte para a Área de Visualização da CPU:** Destaque nos displays dos registradores e nas células da memória.
 - Corte rápido para a Legenda de Cores.
- **Áudio (Locutor):**

"Agora, vamos ver como o emulador nos ajuda a interagir com essa arquitetura, especialmente com as recentes melhorias."

"Na primeira versão deste emulador, a única forma de interagir era escrevendo todo o código Assembly diretamente no Editor Assembly, aqui embaixo. Isso exige um conhecimento detalhado de cada instrução e endereço."

"Porém, inspirados por um exemplo que o professor deu em aula, adicionamos uma nova e mais amigável forma de entrada: a área de 'Entrada de Valores da Expressão'. Com ela, podemos digitar expressões matemáticas simples, como '5+3' ou '10-2', usando esses botões."

"Ao clicar em 'Entrar', o emulador automaticamente gera o código Assembly correspondente e o insere no editor principal, simplificando muito a programação para cálculos básicos!"

"Para vocês terem uma ideia da fidelidade da emulação, observem como no código do emulador, a inicialização da CPU no método `initialize_cpu()` reflete diretamente o que está na seção 10.1 do artigo de Malvino, que descreve a arquitetura do SAP-1, incluindo o PC, ACC, MAR e a memória de 16 bytes. Essa é a base de todo o meu projeto."

"Além disso, temos o Painel de Controle com botões para 'Carregar Exemplo', 'Montar', 'Executar', 'Passo a Passo', 'Reset' e um slider de 'Velocidade do Clock'. Tudo isso interage com a Visualização da CPU, que mostra os valores dos registradores, da memória e, o mais interessante, as animações do fluxo de dados, indicadas pela legenda de cores."

CENA 4: DEMONSTRAÇÃO PRÁTICA (SOMA 5 + 3, VIA EXPRESSÃO)

- **Tempo:** 3:30 - 6:00
- **Visual:**
 - **Início:** Emulador com a área de entrada de expressão visível.
 - **Ação:** Digitar "5+3" na área de expressão usando os botões.

- **Ação:** Clicar em "Entrar". Mostrar o código Assembly gerado aparecendo no editor. (Ex: LDA OE, ADD OF, OUT, HLT e a área de dados ORG OE, DB 5, DB 3). O montador é chamado automaticamente.
- **Ação:** Clicar em "Passo a Passo" várias vezes, com zoom na área da CPU.
 - **LDA OE:** Animar PC->MAR, PC++, Mem->IR. Depois IR->MAR, Mem[OE]->ACC. Mostrar ACC atualizando para 0x05.
 - **ADD OF:** Animar PC->MAR, PC++, Mem->IR. Depois IR->MAR, Mem[OF]->Reg B. Animar ACC->ULA, Reg B->ULA. **Mostrar o valor da soma (0x08) aparecendo temporariamente na ULA.** Animar ULA->ACC. Mostrar ACC atualizando para 0x08.
 - **OUT:** Animar PC->MAR, PC++, Mem->IR. Depois ACC->Saída. Mostrar LEDs acendendo (bit 3 aceso).
 - **HLT:** Animar PC->MAR, PC++, Mem->IR. Destaque no IR, simulação parando.
- **Ação:** Mostrar o resultado final nos LEDs.
- **Ação (Opcional):** Clicar em "Reset", depois digitar "10-2" na expressão e "Executar" para uma demonstração rápida da execução contínua.
- **Áudio (Locutor):**

"Para demonstrar o funcionamento, vamos usar a nova entrada de expressão. Vou digitar '5+3' usando os botões e clicar em 'Entrar'."

"Automaticamente, o emulador gera o código Assembly no editor e já chama o montador! Vocês podem ver o LDA e o ADD com os endereços de memória corretos para nossos dados, e as diretivas ORG e DB para armazenar o 5 e o 3."

"Agora, vamos rodar 'Passo a Passo'. Observem o que acontece no diagrama da CPU a cada clique:"

"Primeira instrução: LDA OE. O PC aponta para a instrução, o endereço vai para o MAR, o PC incrementa. A instrução é buscada na memória e vai para o IR. O operando 'OE' do IR é enviado para o MAR, que busca o valor '5' na memória. O '5' é então carregado no Acumulador. Vejam o ACC agora com 0x05!"

"Segunda instrução: ADD OF. Novamente, busca da instrução. O operando 'OF' vai para o MAR, buscando o valor '3' na memória. Esse '3' é carregado no Registrador B. Agora, a ULA entra em ação: ela pega o 5 do ACC e o 3 do Registrador B. O resultado intermediário, 0x08, é brevemente exibido na própria ULA, e então é enviado de volta para o Acumulador! O ACC agora mostra 0x08."

"Terceira instrução: OUT. Mais uma busca. Em seguida, o conteúdo do Acumulador, que é 0x08, é transferido para o Registrador de Saída. E vejam só! Nossos LEDs se acendem, mostrando a representação binária do número 8!"

"Quarta instrução: HLT. A busca ocorre e, ao decodificar o HLT, o programa para de executar. O PC para de incrementar, e o emulador indica que a execução foi interrompida."

"Com o botão 'Executar', poderíamos ver tudo isso acontecer rapidamente, ideal para programas mais longos, como um '10-2' que geraria LDA, SUB, OUT, HLT."

CENA 5: CONCLUSÃO

- **Tempo:** 6:00 - 6:30
- **Visual:** Tela do emulador com o resultado da soma nos LEDs. Pode voltar para a vinheta final.
- **Áudio (Locutor):**

"Como vimos, este emulador do SAP-1 serve como uma poderosa ferramenta didática. As melhorias na entrada, inspiradas em situações de aula, tornam a experiência ainda mais acessível."

"Ele nos permite ir além da teoria, visualizando na prática os ciclos de busca e execução, o papel de cada registrador e o fluxo de dados em um computador básico, com a fidelidade à arquitetura do Malvino."

"Entender a 'mágica' que acontece dentro de um processador é fundamental para qualquer um que se aprofunda na computação."

"Espero que este vídeo tenha sido útil para vocês. O código fonte está disponível no meu GitHub para quem quiser explorar ou contribuir."

"Se gostaram, deixem o like e se inscrevam no canal. Até a próxima!"